

Çevik Bir Dünyada TMMi®

TMMi® Foundation
tarafından
hazırlanmıştır.

Editör: Erik van Veenendaal

Telif Hakkı Bildirimi
Sınırsız dağıtım Telif Hakkına tabidir
Telif Hakkı © TMMi® Foundation, İrlanda.

Bu TMMi® Foundation içeriği, "olduğu gibi" esasına göre hazırlanmıştır.

TMMi® Foundation, amaca uygunluk veya ticari elverişlilik ya da münhasırlık taahhüdü veya içeriğin kullanımından elde edilen sonuçlara dair taahhütler de dahil olmak üzere ancak bunlarla sınırlı olmaksızın herhangi bir konuda, açıkça veya ima yoluyla hiçbir taahhütte bulunmaz. TMMi® Foundation, patent, marka veya telif hakkı ihlalinin muaf olma konusunda hiçbir taahhütte bulunmaz.

Herhangi bir markanın bu belgede kullanılması hiçbir şekilde ilgili marka sahibinin haklarını ihlal etme amacı taşımaz.

Bu belgenin çoğaltılmasına ve bu belgeden, kurum içi kullanım için başka çalışmalar türetilmesine, tüm çoğaltılan belgelerde ve türev çalışmalarda telif hakkı ile "Taahhüt Yok" ifadelerine yer verilmesi kaydıyla izin verilir.

Bu belgenin çoğaltılmasına veya bu belgeden, harici ve ticari kullanım için başka çalışmalar türetilmesine yönelik talepler TMMi® Foundation'a iletilmelidir.

TMMi®, TMMi® Foundation'ın tescilli markasıdır.

Katkıda Bulunanlar

Asim Ali	(Birleşik Arap Emirlikleri)
Katalin Balla	(Macaristan)
Clive Bates	(İngiltere)
Jan Jaap Cannegieter	(Hollanda)
Vahid Garousi	(Hollanda)
Alon Linetzki	(İsrail)
Fran O'Hara	(İrlanda)
Jurian van de Laar	(Hollanda)
Leanne Howard	(Avustralya)
Poonam Jain	(Hindistan)
Tim Moore	(İngiltere)
Alfonsina Morgavi	(Arjantin)
Meile Posthuma	(Hollanda)
Matthias Rasking	(Almanya)
Chaobo Shang	(Çin)
Erik van Veenendaal	(Bonaire - Hollanda Antilleri)
Blaine Webb	(İngiltere)
Karolina Zmitrowicz	(Polonya)

Versiyon geçmişi

Bu bölüm yalnızca bilgi amaçlıdır.

Versiyon	Tarih	Açıklama
1.0	30/06/2017	Giriş Bölümü ile tüm TMMi® 2. Seviye süreç alanları ele alınmıştır
1.1	16/05/2018	Tüm TMMi® 3. seviye süreç alanlarının Çevik bağlamdaki yorumu eklenmiştir.
1.2	10/12/2018	Tüm TMMi® 4. seviye süreç alanlarının Çevik bağlamdaki yorumu eklenmiştir.
1.3	03/07/2019	Tüm TMMi® 5. seviye süreç alanlarının Çevik bağlamdaki yorumu eklenmiştir.

Önsöz

Yazılımdaki bir satır kod değişikliğinin artık saniyeler içerisinde canlıya alınması gereken günümüz rekabetçi dünyasında “continuous integration”, “continuous delivery”, ve “continuous deployment” gibi kavramları sıkça duymaya başladık. Bu kavramların yanında olmazsa olmazlardan biri de vardır ki o da “continuous testing” dir.

“Sürekli test etme” kavramı, yazılım geliştirmenin her aşamasına test sürecinin entegre edilerek testlerin manuel veya otomatize olarak yapılması olarak karşımıza çıkmaktadır. Bu da test sürecinin kurumun yazılım geliştirme kültürüne ve kullandığı yazılım geliştirme çerçevesine, bu Scrum, Kanban, XP gibi çevik yazılım geliştirme çerçeveleri olabileceği gibi, V-model ve şelale gibi yöntemler de olabilir, uygun bir şekilde kurgulanmasını gerekli kılmaktadır. İşte bu noktada TMMi® bu kurgunun yapılabilmesi için referans noktası olarak alınabilecek test süreçlerini, süreç alanlarını ve hedeflerini Entegre Test Olgunluk Modeli adı altında tanımlamıştır. Model hangi yazılım geliştirme yaşam döngüsü çerçevesini kullanırsanız kullanın temel alınıp kurumunuza özgü revizeler yapıp üzerine kendi test süreçlerinizi inşa edebileceğiniz bir modeldir.

Elinizdeki bu doküman ise özellikle günümüzde çok ihtiyaç duyulan çevik yazılım geliştirme çerçevelerinde test sürecinin nasıl ele alınacağına referans noktası sağlayan değerli bir dokümandır. “Çevik Bir Dünyada TMMi®” dokümanı Yazılım Test ve Kalite Derneği gönüllülerinin uzun süren özveri ile çalışmaları sonucunda titizlikle hazırlanmıştır. Bu vesile ile tüm dernek gönüllülerine tekrar teşekkür eder, dernek ve TMMi arasındaki bu iş birliğinin Türkiye’de yazılımların daha iyi, daha kaliteli ve daha hızlı geliştirilmesine katkıda bulunmasını dileriz.

Çevirinin Türkiye bilişim sektörüne faydalı olması dileğiyle.

Yazılım Test ve Kalite Derneği

Ocak, 2022

Teşekkür

“Çevik Bir Dünyada TMMi®” dokümanının Türkçeleştirme çalışmasına katkıda bulunan Yazılım Test ve Kalite Derneği çeviri çalışma grubu üyelerine burada tekrar teşekkür etmek isteriz. Çalışma grubu üyeleri (alfabetik sıraya göre):

- Aynur Akkol Özkan
- Azmi Yüksel
- Başak Zülfikar Şen
- Betül Bulut
- Burcu Balaban
- Cem Polat
- Çiçek Tuna
- Çiğdem Aldan
- Çiğdem Koyuncuoğlu
- Didem Çolak Arslan
- Emre Altuntaş
- Ergül İnanc
- Fırat Gültekin
- Gizem Taşçı
- Gonca Çiftçi
- Gülşah Bayram Çetin
- Hakan Başer
- Hatice Işık Özata
- Koray Yitmen
- Meltem Aydaş
- Merve Gündüz
- Merve Nergiz Afatsun
- Miraç Emektar
- Naime Zümra Yücesoy
- Onur Uysal
- Onur Yılmaz
- Sadullah Kişi
- Serkan Cura
- Serra Seren Dizen
- Sevdâ Aynacı
- Sinan Verdi
- Volkan Şanoğlu
- Yunis Emre Meral

Yazılım Test ve Kalite Derneği Hakkında

(Turkish Testing Board – www.turkishtestingboard.org)

Yazılım Test ve Kalite Derneği, 2006 yılından bu yana Türkiye bilişim sektöründe yazılım testi farkındalığının artması ve gelişmesi için kâr amacı gütmeyen gönüllü bir şekilde aşağıdaki faaliyetleri gerçekleştirmektedir:

Uluslararası Sertifikasyon Sınavları

Dernek uluslararası ISTQB® sertifika sınavlarını gerçekleştirerek sınavlarda başarılı olan katılımcılara uluslararası geçerliliği olan sertifikalar vermektedir. 2006 yılından bu yana 5.000'den fazla test uzmanı adayı derneğe başvurarak sertifika sınavlarına girmiştir. Uluslararası geçerliliği olan sınavlar dernek tarafından Türkçe ve İngilizce olarak düzenlenmekte ve sınavlara remote(online) ve fiziksel olarak girilebilmektedir:

- o ISTQB® Uluslararası Sertifikalı Temel Seviye Yazılım Test Uzmanı Sınavı
- o ISTQB® Uluslararası Sertifikalı Çevik Test Uzmanı Sınavı
- o ISTQB® Uluslararası Sertifikalı Temel Seviye – Performans Test Uzmanı Sınavı
- o ISTQB® Uluslararası Sertifikalı İleri Seviye – Test Analisti Sınavı
- o ISTQB® Uluslararası Sertifikalı İleri Seviye – Teknik Test Analisti Sınavı
- o ISTQB® Uluslararası Sertifikalı İleri Seviye – Test Yöneticisi Sınavı
- o ISTQB® Uluslararası Sertifikalı Uzman Seviye – Test Otomasyon Mühendisi Sınavı
- o ISTQB® Uluslararası Sertifikalı Uzman Seviye – Yapay Zeka Testi Sınavı

Uluslararası Testİstanbul Konferansları – www.testistanbul.org

Dernek, 2010 yılından beri Uluslararası Testİstanbul Konferanslarını düzenlemektedir. Geçtiğimiz 12 konferansta 50 ülkeden, 50 keynote ve 7.000'den fazla katılımcı ağırlanmıştır.

Paneller

Dernek, yazılım test sektörünün gelişimi için sektör veya konu bazlı paneller organize etmektedir. Bu panellere şu ana kadar 1.000'den fazla profesyonel katılım göstermiştir. Şimdiye kadar düzenlenen paneller:

- o TestFintech,
- o TestDefence,
- o TestGames,
- o TestInsurance,
- o TestAnkara,
- o Testİzmir,
- o Test Finance,
- o TestRemote

Turkey Software Quality Report (TSQR)

Dernek tarafından 2011 yılından itibaren yüzlerce bilişim profesyoneli ve akademisyeninin katılımıyla her yıl düzenlenen anket sonuçlarının değerlendirilmesiyle hazırlanan, Türkiye bilişim sektörüne yön verir nitelikte çıkarımların olduğu rapordur. İngilizce yayınlanan rapor tüm ISTQB® üye dernekleri aracılığıyla 100'den fazla ülkedeki bilişim profesyoneline ulaşmaktadır.

ISTQB® Worldwide Software Testing Practices Report (WSTPR)

ISTQB® tarafından 100'den fazla ülkeden binlerce bilişim profesyoneli ve akademisyeninin katılımıyla düzenlenen anket sonuçlarının değerlendirilmesiyle hazırlanan, dünya bilişim sektörüne yön verir nitelikte çıkarımların olduğu rapordur.

Türkçeleştirme Çalışmaları

Uluslararası yazılım test terminolojisinin ülkemize kazandırılması için dernek bünyesinde yer alan gönüllü çeviri grubu ISTQB® dokümanlarının çevirisi üzerinde çalışmaktadır. Şu ana kadar çevrilen dokümanlar:

- o Çevik Bir Dünyada TMMi®
- o ISTQB® Uluslararası Sertifikalı Temel Seviye Yazılım Test Uzmanı Ders Programı 2018
- o ISTQB® Uluslararası Sertifikalı Temel Seviye Yazılım Test Uzmanı Ders Programı 2011
- o ISTQB® Yazılım Testi Terimler Sözlüğü
- o ISTQB® Uluslararası Sertifikalı İleri Seviye – Test Analisti Ders Programı 2012

Burslar

Dernek her yıl karından belli bir miktarı T.C. Üniversitelerinin Bilgisayar/Yazılım Mühendisliği ve Bilgisayar Programcılığı bölümlerinde okumakta olan başarılı ve ihtiyaç sahibi öğrencilere burs olarak aktarmaktadır.

TMMi® Turkey Chapter

Test Maturity Model integration (TMMi®), yazılım testlerinin daha etkili ve daha hızlı bir şekilde yapılması için çalışmalarda bulunan kar amacı gütmeyen uluslararası bir kuruluştur. TMMi® tarafından geliştirilen Entegre Test Olgunluk Modeli (TMMi®) ücretsiz olarak kullanılabilen bir yazılım kalitesi ve test olgunluğu modelidir. Modeli hayata geçirebilmek ve test süreci iyileştirme alanında faaliyet gösteren profesyoneller için TMMi® Professional ve TMMi® Test Process Improver sertifika programları bulunmaktadır.

Yazılım Test ve Kalite Derneği 2018 yılından beri TMMi® Türkiye temsilcisi olarak çalışmalarını sürdürmektedir.

İçindekiler

1	Giriş.....	7
1.1	Hedefler.....	7
1.2	TMMi® ve Çeviklik.....	7
1.3	Entegre Test Olgunluk Modeli (TMMi®).....	8
1.4	Çeviklik.....	9
1.5	Çevik Bağlamda Test Süreci İyileştirmesi	10
2	TMMi® 2. Seviye Yönetilen	12
2.1	Süreç Alanı 2.1 Test Politikası ve Stratejisi	12
2.2	Süreç Alanı 2.2 Test Planlama	14
2.3	Süreç Alanı 2.3 Test Gözetimi ve Kontrolü	17
2.4	Süreç Alanı 2.4 Test Tasarımı ve Yürütme	20
2.5	Süreç Alanı 2.5 Test Ortamı.....	24
3	TMMi® 3. Seviye Tanımlanan	26
3.1	Süreç Alanı 3.1 Test Organizasyonu	26
3.2	Süreç Alanı 3.2 Test Eğitim Programı	28
3.3	Süreç Alanı 3.3 Test Yaşam Döngüsü ve Entegrasyonu	29
3.4	Süreç Alanı 3.4 Fonksiyonel Olmayan Test	32
3.5	Süreç Alanı 3.5 Eş-Gözden Geçirmeler	34
4	TMMi® 4. Seviye Ölçülen.....	37
4.1	Süreç Alanı 4.1 Test Ölçümü	37
4.2	Süreç Alanı 4.2 Ürün Kalitesi Değerlendirme.....	38
4.3	Süreç Alanı 4.3 İleri Gözden Geçirmeler	39
5	TMMi® 5. Seviye Optimize Edilen	40
5.1	Süreç Alanı 5.1 Hata Önleme	40
5.2	Süreç Alanı 5.2 Kalite Kontrol.....	41
5.3	Süreç Alanı 5.3 Test Süreci Optimizasyonu	42
6	TMMi® Özel Hedef ve Uygulamalarının Uygulanabilirliğine dair Genel Değerlendirme	44
6.1	TMMi® Değerlendirmeleri.....	44
6.2	TMMi® 2. Seviye Yönetilen	44
6.3	TMMi® 3. Seviye Tanımlanan.....	44
	Referanslar	45

1) Giriş

1.1 Hedefler

Bu dokümanda, yazılım geliştirmeye yönelik Çevik bir yaklaşım ile TMMi® test süreci iyileştirme modelini birleştirmenin, iş hedeflerinize ulaşmak için nasıl bir yol izlenmesi gerektiği açıklanır. TMMi®'ın Çevik bağlamda nasıl faydalı bir şekilde kullanılıp uygulanabileceği ele alınır. Çevikliği korurken ve hatta belki de artırırken TMMi® uygulamalarını hayata geçiren geleneksel test yaklaşımlarına dair kanıtlanmış alternatifler sunulur. Bu doküman, kuruluşlar büyüyüp proje baskıları artıkça sıklıkla gözden kaçabilen kritik test uygulamalarına ilişkin hatırlatmalar sunarak TMMi®'ın, kuruluşun çeviklik seviyelerini artırmadaki ilerleyişine bakmaksızın Çevik bir kuruluşa nasıl yardımcı olabileceğini göstermektedir. Her bir TMMi® süreç alanı ve bu alanların özel hedefleri birer birer ele alınır. Bunların Çeviklik ile nasıl ilişkili olduğu ve uygulamaların genel itibarıyla nasıl görüneceği ifade edilir. Bir uygulamanın katma değeri olmaması sebebiyle Çevik bir bağlamda gerçekleştirilmesi beklenmiyorsa bu da açıkça belirtilir.

Birçok (küçük) Çevik organizasyon başarılı olup büyümektedir; ancak bu organizasyonların belgelenmiş süreçleri sayıca az olabilir ve ilgili kişiler için resmi eğitim programları yetersiz kalabilir. Çevik organizasyonlar büyüdükçe başarıyı koruyabilmek için biraz daha fazla süreç disiplinine ihtiyaç duyacaklardır. Sonrasında ise genellikle mevcut başarılarını sağlayan Çeviklik kültürünü kaybetme korkusunu yaşamaktadırlar. Çevik bir ortamda bir TMMi® test süreci iyileştirme girişimi başlatılırken ortaya çıkabilecek zorluklardan biri de budur.

Bu dokümanın hitap ettiği birçok hedef kitle olup bunlar arasındaki iki ana grup şöyledir:

- Süreçlerinin olgunluğunu korurken Çevik olmak isteyen olgun geleneksel organizasyonlar.
- Başarılı ve büyümekte olan çevik organizasyonlar. Çünkü bu organizasyonlar çevik olmanın faydalarını korumak isterlerken bir tür süreç olgunluğuna da ihtiyaç duyarlar.

Bu dokümanın orijinal TMMi® modelinden bağımsız olarak kullanılması amaçlanmamıştır. Aksine, bu doküman, Çevik ortamda test süreci iyileştirmesi gerçekleştirenlere çeşitli TMMi® hedef ve uygulamalarının yorumlanması ve anlamları konusunda rehberlik etme amacındadır. Bu doküman orijinal TMMi® modelinin bir eki niteliğinde olup tamamlayıcı bir belge olarak kullanılmalıdır.

1.2 TMMi® ve Çeviklik

TMMi® ve Çevik yaklaşımların birbirine zıt olduğu yönünde yanlış bir inancı vardır. Çevik yaklaşımlar ile TMMi® sadece bir arada olduğunda değil aynı zamanda başarılı bir şekilde entegre edildiklerinde de önemli faydalar sağlarlar. Ayrıca doğal olarak çevikliğe bir test iyileştirme çerçevesi gözünden bakmanın zorlukları da bulunmaktadır. TMMi®'daki "i" harfinin, test faaliyetinin yazılım geliştirmenin bütünleşik bir parçası olması gerektiğini ve tamamen ayrı bir süreç olarak değerlendirilmemesi gerektiğini ifade ettiği unutulmamalıdır. Çevik projelerdeki test faaliyetleri daha çok birim testi, test otomasyonu ve keşif testi üzerinde yoğunlaşma eğilimindedir, ancak bundan fazlası mevcuttur! TMMi® modelini Çevik bağlamda kullanmak, genelde unutilan kritik test uygulamalarının hatırlanmasını sağlar. Bu doküman, TMMi® ve Çevik yaklaşım yöntemlerinin birlikte etkili bir şekilde çalışabileceğini örneklerle göstermektedir. Buradaki zorluk, Çevik uygulamaları güçlendirmek ve TMMi® uygulamalarını kolaylaştırmak üzere yalın ilkeleri uygulamaktır.

TMMi® modelinin amacı organizasyona bir dizi uygulamayı "empoze etmek" değildir. Uygun şekilde kullanıldığında TMMi®, test alanları kullanılarak değişimin başarısına ve iş hedeflerine ulaşmaya yardımcı olabilir. Bu husus, uygulanan yaşam döngüsü modelinden bağımsız olarak geçerlidir. TMMi® uygulamaları yazılım geliştirme yaşam döngüsünün bir parçası olarak hayata geçirilmesi beklenen bir parçadır ama yaşam döngüsüne paralel bir alternatif olarak da algılanabilir. Daima TMMi®'ı hayata geçirme amacının ne olduğunu, gerekçesini ve işe nasıl değer kattığını düşünün. Çoğunlukla, son kullanıcı ve iş birimi ihtiyaçlarından hareket edildiği sürece "her" çözüm uyumludur! TMMi®'yı uygularken çok kuralcı olmayın, çünkü TMMi®'ın amacı aslen bu değildir. TMMi® hedeflerini ve uygulamalarını her zaman durumunuzun bağlamına göre yorumlayın. Genel anlamda konuşacak olursak, kendi özel iş bağlamanın kapsamında süreç gereksinimlerini belirleyerek süreç iyileştirme önceliklerinin nasıl yoğunlaştırılacağı ve yönlendirileceği konusunda kararlar alabilirsiniz.

TMMi® ile Çeviklik arasında meydana gelen uyumsuzlukların çoğu ya "iyi bir uygulamanın" hayata geçirildiğinde nasıl görünmesi gerektiğine dair eski TMMi® bakış açısından ya da Çevik değerleri nasıl desteklemeleri gerektiğine dayanan çevik uygulamaların mantığının yanlış anlaşılmasına dayanır. (Öncü) değerlendirmenler de dahil olmak üzere TMMi® uzmanlarının, "iyi TMMi® uyumlu" bir uygulamanın hayata geçirildiğinde nasıl görünmesi gerektiği ile ilgili olarak yanlışlıkla paylaşılabilecek mesajları yeniden değerlendirmeleri ve muhtemelen yeniden kurgulamaları gerekecektir. Çevik yaklaşımlar, TMMi® süreçleriyle beraber doğru bir şekilde uygulandığında test uygulamalarının etkinliği azalmayacak, aksine etkin bir şekilde uygulanacaktır. TMMi® bünyesinde özel (teste ilişkin) uygulamaların yanı sıra genel uygulamaların da olduğunu unutmayın. Genel uygulamaların amacı bir süreç alanının kurumsallaşmasını desteklemektir. Bu sayede organizasyona yeni kişiler katıldığında veya organizasyon bünyesinde başka değişiklikler olduğunda, organizasyonun süreç alanını destekleyecek bir altyapısının olması sağlanır.

Geleneksel tabanlı yazılım geliştirmeden Çevik'e geçmek bugün tanımlandığı gibi süreçlerin azaltılmasına ve yalınlaştırılmasına yönelik bir girişimi de ortaya çıkarabilir. TMMi® tabanlı organizasyonlar, bu şekilde Çevik düşünme biçiminden faydalanacaktır. İnsanların TMMi® modelinde gerçekte olmayan şeyleri varmışçasına algılamaya eğilimi olmuştur, bu da katma değeri olmayan gereksiz süreçler ve iş ürünleri oluşmasına neden

olmuştur. Meselenin özüne inilmesi, iyileştirme hedeflerine geri dönülmesi ve TMMi® modelinin amacına uygun şekilde kullanılmasıyla gerçek katma değerli süreçlerin gerçek süreç gereksinimleri ve hedefleriyle uyumunu desteklemek mümkün olacaktır. Süreçlerin Çevik anlayış ile azaltılıp yalınlaştırılması, insanların gerçekte yaptıklarını yansıtan süreçler oluşmasını ve yalnızca kullanılan verilerin toplanmasını sağlayacaktır. Çevik anlayış, süreçleri olabildiğince basit tutmaya odaklanılmasını sağlar. Bu genellikle kolay değildir ancak TMMi® uygulaması yürütenler için faydalı olacaktır. Çeviklikteki iyileştirmeler genellikle, hızlı hareket edebilen yetkilendirilmiş küçük ekipler vasıtasıyla gerçekleştirilecektir ki bu da TMMi®'ın Çevik olmaktan faydalanabileceği bir başka yöntemdir. Ayrıca, TMMi® 3. seviye ile TMMi® 4. ve 5. seviyeleri arasında doğal bir boşluk olduğunu da unutmayın. Özellikle Çevik organizasyonların, TMMi® 4. ve 5. seviyelerdeki fark yaratan ve katma değeri bulunan uygulamaları büyük bir özenle seçmeleri ve yürütmeleri önerilir. TMMi® yeterince kapsamlı olsa da organizasyonların, başarılı olabilmek için odaklanmalarını gerektiren temel test uygulamalarını ve iyileştirmeleri belirlemeleri gerekir.

1.3 Entegre Test Olgunluk Modeli (TMMi®)

TMMi® sistemi, TMMi® Foundation tarafından test süreci iyileştirmesine yönelik bir kılavuz ve kaynak çerçevesi sunacak şekilde geliştirilmiş olup test yöneticileri, test mühendisleri, yazılımcılar ve yazılım kalitesi uzmanları için önemli olan konuları ele alır. Test çalışması, TMMi®'da tanımlandığı şekilde en geniş kapsamlı haliyle yazılım ürünü kalitesine ilişkin tüm faaliyetleri kapsamalıdır.

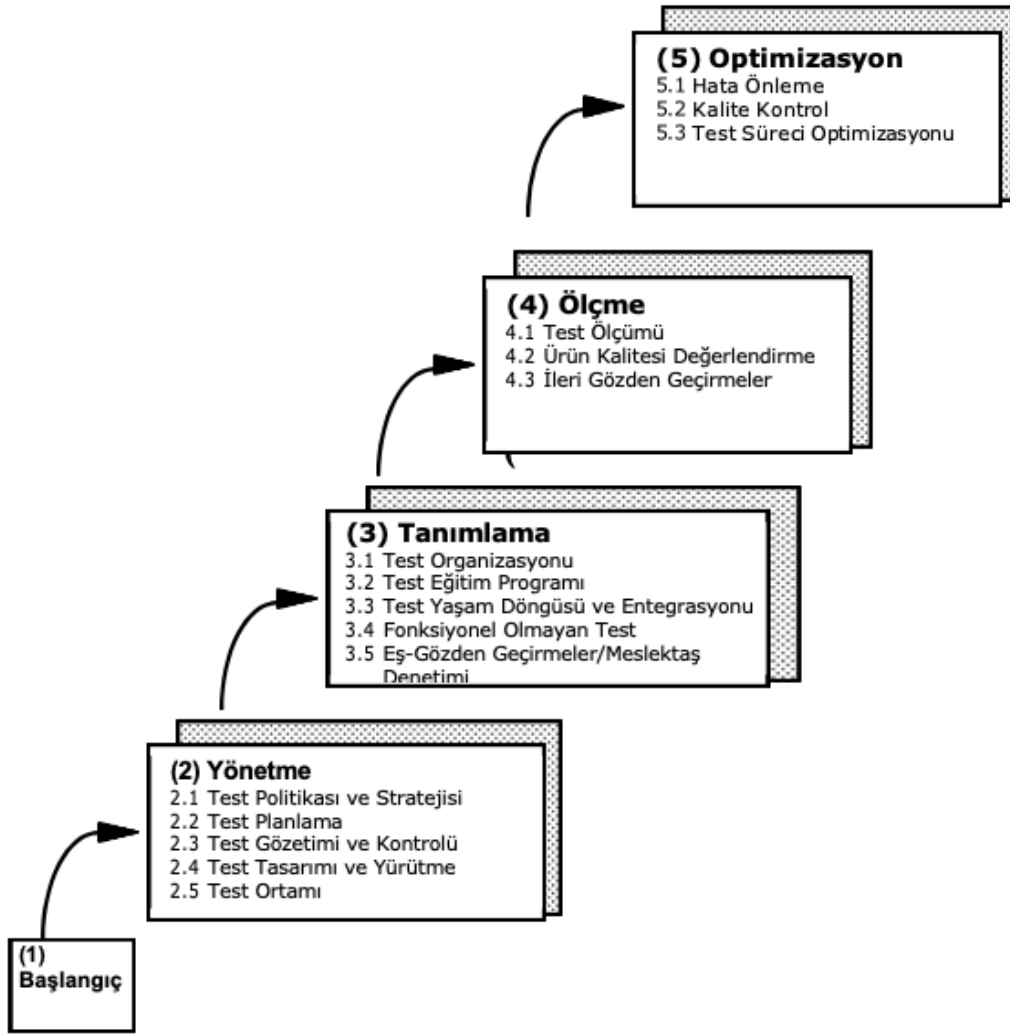
TMMi®, süreç değerlendirme ve iyileştirme için olgunluk seviyeleri kavramını kullanır. Ayrıca süreç alanları, hedefleri ve uygulamaları da belirlenir. TMMi® olgunluk kriterlerinin uygulanması test sürecini iyileştirecek olup ürün kalitesi, test mühendisliği verimliliği ve döngü süresi çabası üzerinde olumlu bir etkisi olduğunu kanıtlamıştır. TMMi®, organizasyonları test süreçlerini değerlendirmeleri ve iyileştirmeleri konusunda desteklemek için geliştirilmiştir.

TMMi®'ın, süreç iyileştirmeye yönelik aşamalı bir mimarisi vardır. Bir organizasyonun test süreci, kurgusuz ve yönetilmeyen bir durumdan, yönetilen, tanımlanmış, ölçülen ve en uygun seviyeye getirilmiş bir duruma evrilirken organizasyonun geçtiği aşamaları veya kademeleri içerir. Her aşamanın başarıyla tamamlanması, o aşamadaki tüm hedeflere ulaşılmasını sağlar ve gerçekleşen iyileştirmeler bir sonraki aşamanın temelini oluşturur.

TMMi®'ın iç yapısı, artan adımlarla iyileştirilen kaliteli bir test sürecini desteklemek üzere sistematik bir şekilde öğrenilebilen ve yürütülebilen test uygulamaları bakımından zengindir. TMMi®'da olgunluk hiyerarşisini ve test süreci iyileştirmesine giden evrimsel yolu belirleyen beş seviye vardır. Her seviye, organizasyonların o seviyedeki olgunluğa erişmek için uygulaması gereken bir dizi süreç alanını içerir. TMMi®'ın her bir olgunluk seviyesindeki süreç alanları Şekil 1'de gösterilmektedir.

TMMi®'ın altında yatan temel ilkelerden biri de çeşitli yaşam döngüsü modelleri ve ortamlarına uygulanabilir genel bir model olmasıdır. Çoğu hedef ve uygulama, TMMi® tarafından tanımlandıkları halleriyle, Çevik çerçeveler de dahil olmak üzere hem sıralı hem de yinelenen yaşam döngüsü modelleriyle uygulanabilir olduklarını kanıtlamışlardır. Bununla birlikte, modelin en alt seviyesinde, verilen alt uygulamaların ve örneklerin çoğu, uygulanan yaşam döngüsü modeline bağlı olarak (çok) farklıdır. TMMi®'da sadece hedeflerin zorunlu olduğunu, uygulamaların ise zorunlu olmadığını unutmamak gerekir.

TMMi®'a, TMMi® Foundation'ın web sitesinden ücretsiz olarak erişilebilir. Model, İspanyolca, Fransızca ve Çince dillerine çevrilmiştir. TMMi® ayrıca, basılı kitap formatında da mevcuttur.



Şekil 1: TMMi® olgunluk seviyeleri ve süreç alanları

1.4 Çeviklik

En yaygın kullanılan sade yazılım geliştirme metodolojilerini temsil eden bir grup kişi, 2001 yılında Çevik Yazılım Geliştirme Manifestosu veya Çevik Manifesto olarak bilinen ortak bir değer ve ilkeler grubu üzerinde anlaştılar. Çevik Manifestosu, dört değer ifade eder.

- Süreçler ve araçlardan ziyade bireyler ve etkileşimlere
- Kapsamlı dokümantasyondan ziyade çalışan yazılıma
- Sözleşme pazarlıklarından ziyade müşteri ile iş birliğine
- Bir plana bağlı kalmaktan ziyade değişime karşılık vermeye

Çevik Manifesto, soldaki kavramlar değerli olsa da sağdakilerin daha büyük değere sahip olduğunu savunmaktadır. Çeviklik, kendi başına bir metodoloji değildir, ancak bu değerlerin hayata geçirilmesini sağlayan bir çerçevedir. Çeviklik'i destekleyen metodolojilerin üç önemli temsilcisi Extreme Programming (XP), Scrum ve Kanban'dır.

1.5 Çevik Bağlamda Test Süreci İyileştirmesi

Çeviklik, kendi sürekli gelişimlerini kendi içlerinde yapan ve ayrıca bunu sık sık geriye dönük gözlemler yoluyla gerçekleştiren yetkin ekiplere odaklanır. Bu iyileştirmelerden bazıları test odaklı olabilir. Söz konusu iyileştirmeler, daha üst düzeyde bir TMMi® tabanlı iyileştirme projesi ile yönlendiriliyor olabileceği gibi proje ekibinin test süreciyle ilgili sorunlarına odaklanıyor da olabilir. Test sürecinin iyileştirilmesi bağlamındaki faaliyetlerin, bu sahipliği Çevik ekiplerden alma olarak yorumlanmaması ve/veya bu amaçla kullanılmaması çok önemlidir.

Çevikliğin iyileştirme bağlamı üzerindeki etkisine bakarken göz önünde bulundurulması gereken başlıca hususlardan bazıları şunlardır:

- İyileştirme döngüsü sıklığı
- Organizasyonel unsurlar
- İyileştirmelerin kapsamı
- İyileştirmelerin kaynağı
- (Test) dokümantasyon seviyesi
- İyileştirme yöntemleri.

Çevik çerçeveyi kullanan projelerde iyileştirmeler genellikle test süreci iyileştirmelerinin sıklıkla gözden geçirilmesini sağlayan (örneğin Scrum kullanırken bir sprintin sonunda) kısa aralıklı geri bildirim döngülerinde gerçekleşir. Kapsam çoğu zaman önceki döngü ile sınırlı olduğu için temel olarak belirli proje sorunlarını çözmeye odaklanan küçük ama sık iyileştirmeler yapılır. Bu iyileştirmelerin odağı genellikle projeler arası öğrenme ile iyileştirmelerin kurumsallaştırılması üzerine değildir.

Test süreci iyileştirmesinin nasıl düzenlendiğine ve yönetildiğine bakıldığında operasyonel düzeydeki bir test süreci grubuna daha az odaklanıldığı ve proje içindeki ekiplerin kendi kendilerini yönetmesine daha fazla önem verildiği görülecektir. Bu ekipler genellikle, projenin test sürecini son derece uyumlu süreçler oluşturarak ihtiyaçlarına göre değiştirme yetkisine sahiptir. Ancak bazı organizasyonlar, işleri projeler arası ve daha üst bir seviyeye getirmek için haftalık ayakta yapılan test toplantılarını da kullanmaktadır.

Test sürecinin iyileştirilmesine daha çok proje özelinde odaklanıldığından, organizasyon genelinde test çalışmasını etkileyen daha geniş kapsamlı konulara muhtemelen daha az önem verilecektir. Örneğin, temel test sorunları, bu proje merkezli anlayışın dışında olduklarından tam olarak ele alınmayabilir. Buradaki tipik örneklerden biri de performans ve güvenilirlik gibi belirli kalite özelliklerini test edilmesi konusunda gösterilen yaklaşımdır. Bu konular, bir döngüden diğerine devamlı ötelenebilir çünkü genellikle proje ekibinin sahip olduğundan daha fazla beceri ve kaynak gerektirirler. Bu alanlarda, büyük yatırımlar olmadan ileriye doğru önemli bir adım atılması zordur. Sorunları yalnızca proje seviyesinde çözmek de hızlıca optimizasyonun gerilemesine ve büyük resmi görmekten uzaklaşılmasına yol açabilir.

Çevik bağlamda değerlendirilecek alternatif iyileştirme fikirlerinin çeşitliliği ve sayısı, Çevik olmayan yaşam döngüsü modellerine kıyasla çok daha fazla olabilir. Tüm üyeler proje içinde bazı testler gerçekleştirdiğinden bu fikirler herhangi bir ekip üyesinden gelebilir. Bu da iyileştirme önerilerinin değerlendirilmesi ve önceliklendirilmesinin önemini ortaya koyar. Bu süreç, bir test süreci iyileştiricisine atanan görevden ziyade bir ekip çalışması şeklinde olabilir. Söz konusu süreçte, bir test süreci iyileştiricisinin özel test bilgilerine ihtiyaç duyulabileceğinden bu kişiler talep üzerine ekibe danışmanlık veya koçluk da sağlayabilirler.

Çevik metodoloji kullanan projelerde, sıralı yaşam döngüsü kullanan projelerden bekleyeceğiniz test belgeleri düzeyini bulmayı beklemeyin. Bir test politikası, test stratejisi ve hatta üst düzey test planının temel öğelerini kapsayan birleştirilmiş tek bir "test belgesi" olabilir. Test süreci iyileştiricileri, daha titiz ve kapsamlı test dokümantasyonu gerektiren "iyileştirme" önerileri yapmaktan kaçınmalıdır. Temel Çeviklik ilkelerinden biri; elbette ki dokümantasyonun ancak bunun için açık ve net bir ihtiyaç olduğu durumlarda oluşturulmasıdır. Test dokümantasyonu gibi süreç dokümantasyonu da daha az detaylı olacaktır. Genellikle "kayıt dışılığın resmileştirilmesi" olarak adlandırılan strateji Çevik ortamlarda başarıyla uygulanır. Bir şey iyi işliyorsa TMMi® uğruna bunu değiştirmeye gerek yoktur. Bununla birlikte, belgelendirilmiş şeyler başkalarına öğretilbilir ve başkalarıyla paylaşılabilir, bu da genellikle faydalıdır. Burada "kayıt dışılığın resmileştirilmesi" stratejisi ile kastedilen, iyi işleyen ancak belli açılardan kayıt dışı olan bir süreç var ise bu sürecin tıpkı gerçekleştirildiği şekilde öğretilip belgelenebileceğidir. Bir sürecin sade olması demek, bu sürecin tanımında olası her kullanım senaryosunun yer almaması demektir. Dolayısıyla bu süreçler, özellikle ilk devreye alma döneminde rehberlik ve iş başında yardımlarla desteklenmelidir.

Sonuç olarak Çevik bir organizasyona Çevik kültürünü koruyarak aynı zamanda test süreci olgunluğunu kazandırırken daha az değil, daha çok eğitim ihtiyacı bulunmaktadır. Aynı şekilde, bir değerlendirme sırasında kanıt toplama üzerinde yoğunlaşan dikkat, ürünlerin incelenmesi yerine daha fazla görüşme yapılmasına kayacaktır.

Çeviklik kullanılırken test süreci iyileştirmeleri önermek için kullanılan yöntemler daha çok neden-sonuç diyagramları gibi sorunların temel nedenlerini değerlendirmeye yarayan analitik yöntemlere odaklanacaktır. Bunlar, bir döngünün sonunda önemli olan problem çözme anlayışı için özellikle faydalı yöntemlerdir.

2) TMMi® 2. Seviye Yönetilen

Süreç Alanı 2.1 Test Politikası ve Stratejisi

Test Politikası ve Stratejisi süreç alanının amacı, test aktivitelerinin (örneğin test çeşitleri -fonksiyonel ve fonksiyonel olmayan- ve test çeyrekleri - iş birimi bakış açılı, teknoloji bakış açılı -) açıkça belirlendiği bir test politikası ile organizasyon veya program genelinde bir test stratejisi geliştirmek ve oluşturmaktır. Test performansı ile test faaliyetlerinin değerini ölçmek ve iyileştirme alanlarını ortaya çıkarmak için test performansı göstergeleri oluşturulmuştur.

2.1.1 Özel Hedef 1 (ÖH1) Test Politikası Oluşturma

Test iyileştirme projesi başlatan her organizasyon, işe öncelikle bir test politikası belirleyerek başlamalıdır. Test politikası, organizasyonun test ve test uzmanlarıyla ilgili genel test hedeflerini, amaçlarını ve stratejik görüşlerini tanımlar. Test politikasının, organizasyonun kalite politikasıyla tümünden uyumlu olması önemlidir. Test iyileştirmeleri, net iş hedefleri doğrultusunda oluşturulmalı, bu hedefler de test (iyileştirme) politikasında belgelenmelidir. Test politikası, organizasyon bünyesindeki tüm paydaşlar arasında test ve testin hedefleri konusunda ortak bir bakış açısının oluşması için gereklidir. Bu ortak bakış açısı, organizasyon genelinde test (süreç iyileştirme) faaliyetlerini düzenlemek için gereklidir. Test hedeflerinin asla kendi başına birer hedef olmaması gerektiği; çalışan yazılımlar ve ürün kalitesi oluşturmaya yönelik daha büyük bir amaçtan ortaya çıktıkları unutulmamalıdır.

Bu durumun aynısı, Çevik yazılım geliştirme uygulayan bir organizasyonda da geçerlidir. Gerçekten de birçok organizasyon bünyesinde, Çevik yazılım geliştirmede testin değişen rolü, testin bağımsızlığı, test otomasyonu ve profesyonel test uzmanları konusunda bolca tartışma vardır. Bu unsurlar ve diğerleri, genellikle yönetim ve diğer paydaşlarla yapılacak bir görüşmede ele alınması ve bir test politikasında belgelenmesi gereken konulardır. Çevik yaklaşımı uygulayanlar da dahil olmak üzere test iyileştirme projesi başlatmak isteyen her organizasyon, böyle bir girişim için işe yön veren amaçlarını ve ihtiyaçlarını tespit edip ortaya koymalıdır. İyileştirme projesi başlatmanın amacı da zaten bu değil midir? Gerçek ihtiyaçları belirleme konusunda zaman harcayarak (test) iyileştirme önceliklerinin hangi alanlara (örneğin hangi süreç alanına) yoğunlaşacağı konusunda bir karar verebilmek için kaynak oluşturmak mümkündür. Test politikasının genellikle organizasyonel düzeyde, bir sayfalık ya da bir belge, web sayfası veya duvar grafiği olduğunu ve proje düzeyinde bir belge olmadığını unutmayın.

TMMi®'ın özel bir hedefi olan Test Politikası Oluşturma ile ona ait uygulamalar, Çevik yazılım geliştirmeyi uygulayan organizasyonlar için tam olarak uygulanabilir. Elbette ki test politikasının unsurları bir geliştirme politikasına da dahil edilebilir. Bunun ayrı bir belge olması konusunda özel bir TMMi® gereksinimi yoktur. Çevik yazılım geliştirmeyi uygulayan bir organizasyondaki geliştirme politikası, örneğin, Scrum'ı yönetim çerçevesi, XP'yi kullanan ana yöntem ve Çevik değerlere uyumu da ana ilke olarak belirleyebilir. Değinilebilecek bir diğer önemli ilke de ekipteki herkesin her şeyden sorumlu olmasıdır; ayrıca ürün kalitesi de ekibin sorumluluklarından biridir.

Ancak, Çeviklik açısından test politikası ve özellikle de belirlenmiş test (iyileştirme) hedefleriyle ilgili olarak üzerinde durulması gereken önemli bir husus vardır. Organizasyonun test süreci iyileştirmeye ilgili genel hedefleri olabilir, ancak bunların, bireysel projelerle ve Çevik ekiplerin kendi süreçlerini iyileştirmekten sorumlu olmaları ile dengelenmesi gerekir. Çevik süreç iyileştirmenin zorluğu, iyileştirmeyi organizasyon düzeyinde yönlendirip çerçevelerken Çevik ekibin, kendi süreçlerinin sahibi olma anlayışını köreltmemektir.

2.1.2 ÖH2 Test Stratejisi Oluşturma

Test stratejisi, test politikasının hemen arkasından projelerdeki test faaliyetleri için atılması gereken ilk adımdır. Test stratejisi, genellikle organizasyon veya program genelinde belirlenir. Tipik bir test stratejisi, üst düzeyde bir ürün risk değerlendirmesini esas alır ve uygulanacak test çeşitlerinin, test çeyreklerinin ve test seviyelerinin (örneğin birim, kabul) tanımlarını içerir. Örneğin, bir döngüde birim testi ile kabul testinin yapılacağını belirtmek tek başına yeterli değildir. Birim ve kabul testiyle ne kastedildiğini, bunların genelde nasıl uygulandığını ve temel amaçlarının ne olduğunu da açıklamamız gerekir. Deneyimler, bir test stratejisi belirlendiğinde ve takip edildiğinde, çeşitli test faaliyetleri arasında daha az çakışma yaşanmasının muhtemel olduğunu ve böylelikle daha verimli bir test süreci elde edildiğini göstermektedir.

Ayrıca test hedefleri ile farklı test çeşitleri ve seviyeleri uyumlu olduğunda her iki grup arasında daha az sürtüşmenin kalması muhtemeldir; bu da daha etkin bir test süreci ve dolayısıyla daha yüksek bir ürün kalitesi elde edilmesini sağlar. Bu uyumun sağlanabilmesi için aynı projede gerçekleştirilen Çevik ve Çevik olmayan tüm test çalışmalarının test stratejisi çatısı altında toplanması önemle tavsiye edilir. Ancak, Çevik ve Çevik olmayan projeler tamamen ayrı yönetiliyorsa genellikle iki ayrı test stratejisi bulunmalıdır.

Test stratejisi, Çevik ortamda hayati bir belge niteliğindedir. Çevik ekiplerde yapılacak test çalışmasında hangi test çeşitlerinin, test çeyreklerinin ve test seviyelerinin uygulandığını ve bunların yaklaşımlarını ayrıntılı olarak açıklar. Test stratejisi dokümanında, testin nasıl düzenlendiği (örneğin hangi testlerin Çevik ekiplerde, hangilerinin dışarıda yapıldığı) açıklanır. Çevik ekiplerle onların kapsamı dışında gerçekleştirilen test seviyelerine kadarki (örneğin donanım/yazılım entegrasyon testi, sistem entegrasyon testi veya beta testi) ilişki tanımlanır. Bir test stratejisi, testte görev alanların teste ilişkin büyük resmi anlamalarını sağlar.

Yalın test stratejisi belgesi, Çevik organizasyon için çoğu zaman iyi bir çözümdür ve ayrıntılı (proje) test planlarından kaçmanın bir yoludur. Sürüm planlaması süresince organizasyonun veya programın mevcut test stratejisi tartışılıp onaylanır veya proje için özel olarak türetilmiş bir test stratejisi oluşturulur. Onaylanan veya oluşturulan test stratejisi, tüm döngüleri kapsayan bir test çerçevesi sağlar.

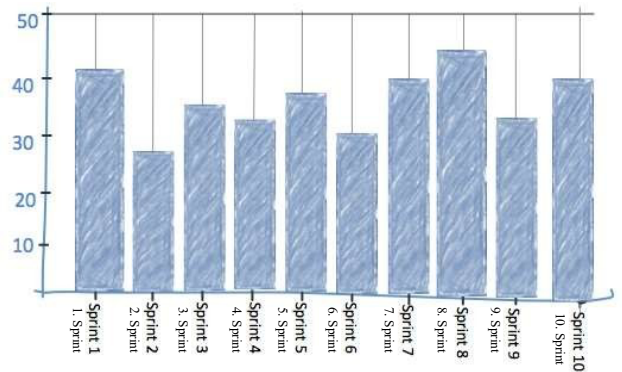
Kendine ait uygulamalarıyla birlikte TMMi®'in özel bir hedefi olan Test Stratejisi Oluşturma, Çevik yazılım geliştiren organizasyonlar için tam olarak uygulanabilir. Test stratejisi belgesinin etkin dağıtımı gibi faaliyetlerin Çeviklikle daha az ilintili olduğu unutulmamalıdır. Bütünsel ekip yaklaşımı dahilinde her paydaş önceki görüşmelerde çoktan yer almış olmalıdır. Bununla birlikte bir test stratejisi, ekip düzeyinde olmayıp daha üst düzey bir çalışmadır. Ekibin tamamı ekip düzeyinde çalıştığından organizasyonel veya programa ilişkin bir belgenin ekiplere olduğu kadar diğer paydaşlara da (örneğin hibrit bir model varsa ekip dışında test faaliyetleri gerçekleştirenler) dağıtılması gerekir.

2.1.3 ÖH3 Test Performans Göstergeleri Oluşturma

Test politikasında tanımlandığı gibi test iyileştirmeye yönelik hedeflerin, bir dizi temel test performans göstergesine uyarlanması gerekir. Test politikası ile beraberindeki performans göstergeleri, açık bir yönerge oluşturmakla birlikte beklenen ve ulaşılan test performansı seviyelerinin iletilmesine de olanak sağlar. Performans göstergeleri, test çalışması ve test süreç iyileştirmesinin değerini paydaşlara göstermelidir. Süreç iyileştirmesine yapılan yatırımlar uzun vadeli yönetim desteğine ihtiyaç duyduğundan paydaşların motivasyonlarını canlı tutmak için bir iyileştirme programının faydalarını nicel olarak ölçmek kritik önem taşır. Bu TMMi® hedefinin sınırlı sayıda (örneğin 2 veya 3) test performans göstergesi belirlemekten ibaret olduğunu unutmamak gerekir. Bu hedef, tam kapsamlı bir ölçüm programının oluşturulması ve uygulanması ile ilgili olmayıp daha ziyade test çalışmasının değerinin zaman içinde ve farklı teslimat ortamlarında nasıl değiştiğini gösteren bir dizi temel gösterge tanımlamaktadır.

Çeviklik'te odak daha çok ekip bazlı ve sistem düşüncesi üzerinde olacaktır. Bu durum, göstergelerin, yalnızca test çalışmasının kendi özel koşullarıyla sınırlı kalmasından ziyade ekibe ve sistemin geneline yayılmasına sebep olabilir. Ayrıca, TMMi® 2. seviyedeki göstergeler, büyük ölçüde döngülerin nihai sonuçlarıyla ilgilidir. Buna ilişkin örnekler arasında atlanmış hatalar, hız, müşteri memnuniyeti derecelendirmeleri, efor/zaman kaybı, test otomasyonu yüzdesi vb. sayılabilir. Buradaki zorluk, TMMi® tabanlı bir test iyileştirme programının performans başarılarına dair iyi bir gösterge verirken aynı zamanda ekip temelli yaklaşım ile sistem odaklı düşünmeye ilişkin göstergelerin uygun bir karışımını belirlemek olacaktır.

Seçilen ve uygulanan performans göstergelerinin yalnızca test çalışmalarıyla ilgili olmanın ötesinde daha geniş bir kapsamı olsa bile kendine ait yöntemleriyle TMMi®'in özel bir hedefi olan Test Performans Göstergeleri Oluşturma, tamamen uygulanabilir durumdadır. Elbette ki bu durum, performans göstergelerinin analizini ve yorumlanmasını daha zor hale getirecektir. Hatta, kullanılan performans göstergeleri test performansı göstergeleri olmaktan ziyade ekip performans göstergeleri veya sistem performansı göstergeleri olarak adlandırılabilir. Bu durum, söz konusu göstergeler, test çalışmasına ilişkin unsurlar içerdiği ve test iyileştirme çalışmasında kaydedilen ilerlemeyi değerlendirmek üzere kullanıldığı sürece TMMi açısından yine de kabul edilebilir.



Süreç Alanı 2.2 Test Planlama

Test Planlamanın amacı, tespit edilen risklere ve tanımlanmış test stratejisine dayalı bir test yaklaşımı belirlemekle birlikte test faaliyetlerini gerçekleştirmek ve yönetmek adına iyi yapılandırılmış planlar oluşturarak bunların sürekliliğini sağlamaktır.

Başarılı bir test planlamanın anahtarının, ilgili test planını ("belge") oluşturmak değil de önden düşünme ("süreç") olduğunu unutmayın.

Çevik yaşam döngülerinde, genellikle iki tür planlama gerçekleşir: sürüm planlaması ve döngü planlaması. TMMi® 2. seviyedeki Test Planlama süreç alanı hem sürüm hem de döngü planlamasındaki test aktivitelerine odaklanır. Sürüm planlaması, bir projenin başlangıcında ürünün piyasa sürümünü öngörür. Sürüm planlaması, belirlenmiş bir ürün iş listesi gerektirmekte olup büyük kullanıcı hikayelerini daha küçük hikayelerden oluşan bir koleksiyona dönüştürülebilir. Sürüm planlaması, tüm döngüleri kapsayan test yaklaşımı ve test planı için bir zemin oluşturur. Sürüm planları üst düzey çalışmalardır. Sürüm planlaması yapıldıktan sonra ilk döngüye dair döngü planlaması başlar. Döngü planlaması, tek bir döngünün sonunu öngörmekte olup döngü iş listesiyle ilgilidir.

Test Planlama süreç alanının kendine özgü özel uygulamaları olduğunu unutmamak gerekir. TMMi® bu uygulamaların ne zaman veya nasıl yürütüleceğini belirtmez. Ayrıca aşamalı olarak planlayamayacağınızı da belirtmez. Geleneksel yaklaşım, mümkün olduğu kadar çok kararın peşinen kesinleştirilmesi şeklindedir; böylece ilgili maliyetler ve takvim de kesinleşmiş olur. Bu yaklaşımın gerekçesi işi daha iyi tahmin etmek ve kapsam dışına çıkma riskini azaltmaktır. Çevik yaklaşımlar genel anlamda, en son bilgilere ve müşteri ile devam eden iş birliğine dayanarak planın sürekli iyileştirilmesiyle daha fazla değer kazanılacağı görüşünü benimsemiştir.

2.2.1 ÖH1 Risk Değerlendirmesinin Gerçekleştirilmesi

Her şeyi bütünüyle kapsayacak testler yapmak imkansızdır; daima tercihler yapılması ve önceliklerin belirlenmesi gerekir. Dolayısıyla bu TMMi® hedefi, Çevik projeler için de geçerlidir. Çevik projeler için sürüm planlaması aşamasında ürün tanımlı belgesine veya detaylı kullanıcı hikayelerine göre kapsamlı bir ürün risk değerlendirmesi yapılmalıdır. Her döngü için ilgili döngü planlaması kapsamında, kullanıcı hikayelerine veya söz konusu döngünün diğer gerekliliklerine göre daha ayrıntılı bir ürün riski çalışması yapılmalıdır. Çevik bir projedeki ürün riski değerlendirme süreci, sıralı bir yaşam döngüsü modelini izleyen geleneksel projelerde uygulananlara kıyasla çok daha basit olacaktır. Kullanılabilecek basit ürün riski tekniklerinin bir örneği de risk pokeridir [Van Veenendaal].

Sürüm planlamasında, sürümdeki özellikleri bilen iş kolu temsilcileri, geliştirilecek olan işleve dair genel kapsamlı bilgiler verirler ve test uzman(lar)ı da dahil olmak üzere tüm ekip, risk belirleme ve değerlendirme konusunda destek sağlar.

Döngü planlaması sırasında Çevik ekip, bir sonraki döngüde uygulanacak kullanıcı hikayelerine göre ürün risklerini belirler ve analiz eder. Tercihen, Çevik ekibin tüm üyeleri ve hatta bazı diğer paydaşlar da ürün riski çalışmasında yer alır. Bu çalışma sonucunda, test için kritik alanları belirleyen öncelikli ürün riski unsurlarının bir listesi ortaya çıkar. Bu listede, her bir riski yeteri kadar testle karşılamak için gerekli olan test çalışması miktarının belirlenmesine ve bu testlerin yapılacak test çalışmasının etkinliğini ve verimliliğini en yüksek seviyede tutacak şekilde sıralanmasına yardımcı olacaktır. Görev panosundaki tahmini görevler, kısmen bunlarla ilişkili ürün riski seviyelerine bağlı olarak belirlenebilir. Daha yüksek risk teşkil eden görevler daha erken başlamalı ve bunlar için daha fazla test çalışması yapılmalıdır. Daha düşük risk teşkil eden görevler ise daha geç başlamalı ve bunlar için daha az test çalışması yapılmalıdır.

2.2.2 ÖH2 Test Yaklaşımı Oluşturma

Test yaklaşımı, tespit edilmiş ve önceliklendirilmiş ürün risklerini azaltmak için oluşturulur. Belirli bir döngüye ait test edilecek unsurlar ve özellikler döngü planlaması sırasında tespit edilir. Bu çalışma aynı zamanda ürün risk aşamasının sonucuna dayanmaktadır. Test edilecek öncelikli unsurların listesi, genellikle bu döngüde test edilecek kullanıcı hikayeleri ile ilgilidir. Özellikler genel anlamda, diğerlerinin yanı sıra, test edilecek çeşitli yazılım kalite özellikleri ile ilgilidir. Döngü sırasında ilave testler gerektiren yeni bir ürün riski ortaya çıkabilir. İlave test gerektiren yeni ürün riskleri gibi konular genellikle günlük ayakta toplantılarda tartışılmaktadır.

Riskleri azaltmak amacıyla döngü seviyesinde belirlenmiş test yaklaşımı, kullanıcı hikayeleri ile kabul kriterlerinin; risk seviyesine orantılı test çalışması miktarının; riskin düzeyi ve türüne bağlı olarak uygun test tekniğinin/tekniklerinin seçiminin yeniden gözden geçirilmesini içerebilir. Sürüm düzeyindeki test yaklaşımı çok daha kapsamlı olacaktır ve bu yaklaşım program veya organizasyon seviyesinde belirlenmiş test stratejisine dayandırılmalıdır. Test yaklaşımı genellikle ekip/proje wiki'sinde tutulur veya gösterilir.

Döngüsel geliştirmede her zaman mevcut olan önemli bir risk de regresyon riskidir. Test yaklaşımı, regresyon riskinin nasıl yönetildiğini tarif etmelidir. Genellikle bu, tercihen otomatikleştirilmiş belirli bir regresyon test kümesi oluşturularak yapılır. Bu bağlamda test otomasyon piramidi [Cohn] faydalı olmaktadır. Piramidin en düşük seviyesindeki birim testlerinden başlayıp servis testlerine geçerek regresyon testi otomasyonunun değerinin nasıl en üst düzeye çıkarılacağını gösterir. Kullanıcı arayüzü testi en üstte yer alır. Birim testler hızlı ve güvenilirdir. Servis katmanı, iş mantığının, kullanıcı arayüzü (UI) kısıtlamalarına maruz kalmadığınız API veya servis düzeyinde test edilmesine imkan tanır. Seviye ne kadar yüksek olursa test de o kadar yavaş ve hassas olur.

Normalde belirlenmiş bir test yaklaşımının parçası olan giriş kriterlerinin (özel uygulama 2.3), Çevik yazılım geliştirme ile ilgili olmaması muhtemeldir. Çevik yazılım geliştirmede test çalışması, ekip sürecinin ayrılmaz bir parçası olup neredeyse devamlılığı olan bir faaliyetdir. Bundan hareketle, testin başlatılıp başlatılamayacağını belirlemek için belli bir kontrol listesine veya yöneme ihtiyaç yoktur. Bu durum, Çevik bir ekipte bir test aşamasından (örneğin birim testi) diğerine (örneğin kabul testi) geçen bir bileşen için de geçerlidir.

Test çıkış kriterleri (özel uygulama 2.4), Tamamlandı Tanımı (DoD) olarak bilinen uygulamanın bir parçasıdır. DoD'de teste ilişkin belirli kriterlerin [örneğin, test kapsamı ve ürün kalitesi (hatalar)] olması önemlidir. Döngü, kabul edilmiş kullanıcı hikayeleri setinin uygulanmasıyla sonuçlanmalı ve DoD'de tanımlandığı şekilde (test) çıkış kriterlerini karşılamalıdır. Genellikle çıkış kriterlerini yerine getirmeyen hikayeler iş listesine konur ve bir sonraki döngüde de ele alınabilir. Elbette ki Çevik yaklaşımda, bir test etabından diğerine geçen bir bileşene dair çıkış kriterleri yoktur. Tamamlandı Tanımı yalnızca döngü düzeyinde mevcut olmayıp çoğu zaman sürüm seviyesinde de birden fazla döngüyü kapsayan bir Tamamlandı Tanımı olacaktır. Genel itibarıyla sürüm seviyesindeki DoD'de yine kapsama ve ürün kalitesiyle ilgili kriterler bulunacaktır.

Özel uygulama 2.5 Askıya alma ve devam kriterlerinin belirlenmesi Çevik yaşam döngüleriyle ilgili olmayabilir. Test, Çevik yazılım geliştirme sürecinin ayrılmaz bir parçası olduğundan elbette ki diğer döngü faaliyetlerinden ayrı ve bağımsız bir faaliyet olarak değerlendirilmez. Testin ilerlemesine yönelik olası veya gerçek tehditler olarak kabul edilebilecek engelleme sorunları olduğunda bunlar günlük ayakta toplantılarda tartışılmaktadır. Bu toplantıda ekip, sorunları çözmek için gerektiğinde hangi işlemlerin yapılması gerektiğine karar verecektir. Dolayısıyla, resmi askıya alma ve devam kriterleri gerekli olmayacak ve belirlenmeyecektir; bununla ilgili sorunlar normal Çevik çalışma düzeninin bir parçası olarak ele alınacaktır. Çevik rutin böylelikle bu özel uygulama için alternatif bir uygulama görevi görür.

2.2.3 ÖH3 Test Tahminleri Oluşturma

Çevik ekipler için, döngü planlaması sırasında teste ilişkin ayrıntılı tahminler yapılacaktır. Teste dair detaylı tahminler sürüm planlaması sırasında ve muhtemelen iş listesinin hafifletilmesine dair çalışmalar sırasında yapılır. Elbette tüm tahminler her bir kullanıcı hikayesini hayata geçirmek için ihtiyaç duyulan tüm çalışmayı kapsayan ortak ekip kararıyla ele alınır. Tahmin çalışmaları sırasında test faaliyetlerinin mutlaka dikkate alınmasının sağlanması önemlidir. Bu, test faaliyetlerinin ayrı görevler olarak tanımlanması ve ardından her bir test görevinin tahmin edilmesi yoluyla veya yapılacak testin açık bir şekilde yapıldığı ve dolayısıyla dikkate alınan her bir kullanıcı hikayesinin tahmin edilmesiyle yapılabilir. Test uzmanı, Çevik ekibin bir parçası olarak tahmin çalışmalarına katılacaktır. Poker Planlama ya da tişört bedeni teknikleri, Çevik yazılım geliştirmede kullanılan yaygın tahmin teknikleridir.



Bir sonraki döngü için yapılacak çalışmalar genellikle kullanıcı hikayelerine göre belirlenir. Kullanıcı hikayelerinin tahmin edilebilir olması için küçük olmaları gerekir. Tahmin edilebilirlik, INVEST [Wake] kısıltmasında yer alan kullanıcı hikayesi kriterlerinden biri olup bir döngünün parçası olacak kullanıcı hikayeleri için geçerlidir. Döngü planlaması sırasında Çevik ekipler genellikle testle ilgili görevleri belirler ve tayin eder. Test görevleri, diğer geliştirme görevleriyle beraber bir görev panosunda ele alınacaktır. Bu görev grubu, döngünün tahmini için bir temel oluşturur. Gelecekteki döngü için tayin edilmiş görevler, bir nevi iş kısıtlı yapısı olarak işlev görür (sıralı projelerde kullanıldığı gibi).

Sürüm planlamasında, kullanıcı hikayeleri veya epikler genellikle daha yüzeysel tanımlanmış olup henüz belirli görevler şeklinde detaylandırılmış olmayacaktır. Bu durum elbette tahminlerin daha zor ve daha az doğruluk payıyla yapılmasına sebep olacaktır. Belirtildiği üzere Çevik projelerde, tahminler için bir iş dökmü yapısı oluşturulmayacaktır, ancak sürüm seviyesinde, geliştirilecek ürünün görselleştiren ve böylece işi yeterince kapsamlandıran basit bir diyagramdan faydalanılabilir.

Çevik bir ekip nispeten gayriresmi bir şekilde tahminlerde bulunsun da tahminlerin gerekçesinin (yani hangi faktörlerin dikkate alındığının) açık olması gerekir. Gerekçeye dayalı tartışmalar, tahminlerin daha yüksek bir doğruluk seviyesine ulaşmasına yardımcı olur. Çevik projelerde tahminler genellikle büyüklüğe (hikâye puanları kullanarak) veya iş hacmine (bir tahmin birimi olarak ideal insan çalışma günleri kullanarak) odaklanır. Çevik projelerde, normal koşullarda, maliyetler tahmin çalışmaları kapsamında ele alınmaz.

Dolayısıyla, bu TMMi® özel hedefi ve onun özel uygulamaları, 3.2 Test yaşam döngüsü oluşturma özel uygulaması dışında tamamen uygulanabilir. Döngüsel ve Çevik yazılım geliştiriminin esaslarından biri de işi küçük parçalar halinde yapmaktır. Dolayısıyla, belirlenmiş görevler genel olarak (teste ilişkin) tahminleme için bir dayanak işlevi görebilecek kadar ayrıntılıdır. Bu sebeple, Çeviklik'te, ayrıca tahminleme için ilave bir dayanak işlevi görecektir test faaliyetlerine yönelik bir yaşam döngüsü tanımlamaya gerek yoktur.

2.2.4 ÖH4 Test Planı Geliştirme

Test planlamanın, ilgili test planını ("belge") oluşturmakla değil de önden düşünmeyle ("aktivite") ilgili olduğu görüşünün yeniden belirtilmesidir. Çevik yaklaşımdaki test planlama faaliyetlerinin çoğu bu TMMi® özel hedefinde belirtildiği şekilde, sürüm ve döngü planlamaları sırasında gerçekleştirilecektir. Ancak, bu faaliyetlerin sonucu bir test planında, özellikle de söz konusu faaliyetlerin görev panosuna yansıtılabileceği döngü planlaması için olan bir test planında belgelenmeyecektir.

Test, sürüm ve döngü planlamasının ayrılmaz bir parçası olarak gerçekleştirildiğinden ortaya çıkan "çizelgeler" test faaliyetlerini de içerecektir. Çevik bir projedeki çizelge, sıralı yaşam döngüleri ile geliştirilen ayrıntılı bir çizelgeden çok kullanıcı hikayeleri (birikmiş iş maddeleri) ile sürüm ve döngü önceliklerini yansıtan görevlerin (örneğin istenen iş değeri teslimine dayanan) sıralanması/düzenlenmesi gibidir. Zihin haritaları bu noktada destekleyici teknik olarak sıklıkla kullanılır. Görev panosu döngü önceliklerini yansıtacaktır. Dolayısıyla açık bir test çizelgesi oluşturulmamıştır; net sürüm ve döngü öncelikleri ile test görevleri de dahil olmak üzere sırasıyla yerine getirilecek görevlerin kullanıcı hikayelerine uygun olarak belirlenmesi beklenmektedir.

Proje düzeyinde, test uzmanının belirlenmesi ekibin kurulmasının bir parçasıdır; çoğul becerili kaynaklara veya teste olan ihtiyaç önceden belirlenmektedir. Projeler değiştikçe veya büyüdükçe bir kişinin belirli bir ekibe/projeye dahil edilmesine dair başlangıçtaki gerekçeler kolayca unutulabilir. Bu gerekçeler arasında çoğu zaman ihtiyaç duyulan bazı özel beceriler veya ekibe/projeye özgü belirli deneyimler yer alır. Gerekçelerin unutulmaması açısından kişilerde ihtiyaç duyulan becerilerin yazılması ve bunların, kişilerin neden ilgili ekip/proje için seçildiğine dair bilgilerle desteklenmesi faydalıdır. Çevik ekipler belirlendikten sonra test uzmanlarının kimlerden oluşacağı da aşağı yukarı bellidir. Döngü planlaması sırasında, döngüdeki testin yürütülmesi için gereken (ek) kaynakların ve becerilerin (örneğin fonksiyonel olmayan testler için) belirlenmesi, ekibin gereken testleri gerçekleştirmeye yetecek düzeyde test kaynaklarına, bilgisine ve becerisine sahip olduğundan emin olmak için gerektiğinde tartışılabilir.

Scrum yöneticisi, örneğin soruları yanıtlayarak ve kullanıcı hikayeleri üzerine konuşarak, ürün sahibinin teste gereken şekilde girdi iletilmesini sağlamalıdır. Ürün sahibi yeterince ulaşılabilir olmalı ve bu da yine önceden programlanmalıdır.

İlk proje risk çalışması, sürüm ve döngü planlaması kapsamında yapılmalıdır. Döngü sırasında başka proje risklerinin belirlenmesi (ve yönetimi) günlük ayakta toplantıların bir parçasıdır ve genellikle bir engel günlüğü aracılığıyla belgelenir. Engel günlüğünde test sorunlarının da belirtilmesi önemlidir. Engeller, çözülene kadar günlük ayakta toplantılarda görüşülmelidir.

Belirgin ve detaylı bir test planı geliştirilmemiş ve belgelenmemiş olmasına rağmen, 4.5 Test Planını Oluşturma Özel Uygulaması Çeviklik açısından yine de uygulanabilir. Öte yandan, test planlaması bağlamında yapılan tartışmaların sonuçları büyük ihtimalle sade bir formda, muhtemelen de bir zihin haritasında belirtilecektir.

2.2.5 ÖH5 Test Planına Bağlılığı Sağlama

Çevik yaklaşımda, test yaklaşımı ile test planı geliştirme ve oluşturma süreci ekip çalışmasına dayanan bir uygulamadır ve genellikle test ekibinin üyesi olan bir test uzmanının liderliğinde yapılır. Ürün kalitesi ekibin sorumluluğundadır. Bu nedenle, ekibin doğru süreci izlemesi kaydıyla, test yaklaşımı ile test planına bağlılık bir ekip çalışması olduğu için sürüm ve döngü planlamasının doğal bir sonucudur. Burada elbette test uzmanının test planını hazırlayıp sonrasında açık mutabakat sağlaması gerektiği geleneksel yöntem çalışma şekline göre ciddi bir fark mevcuttur.

Çevik projelerde, Çevik ekip (ürün sahibi de dahil olmak üzere) öncelikli ürün riskleri listesini ve yapılacak risk azaltma faaliyetlerini anlamalı ve bunlar üzerinde uzlaşmaya varmalıdır. Bu anlayış ve bağlılık, ürün risklerini, test yaklaşımını ve testin gerekçesini ekibe açıklayan sürüm veya döngü planlaması sırasında yapılacak bir tartışmanın takip ettiği kısa bir sunumla sağlanabilir.

Tahmin çalışması sırasında (bkz. ÖH 3 Test Tahminleri Oluşturma) iş yükü tahmin edilir. Bir ekibin (test) kaynakları, Çevik ekibin kurulmasıyla sabitlenmiş olur. Tahmin edilen ve geliştirme için seçilen kullanıcı hikayeleri, mevcut kaynakları başlangıç noktası (kısıtlama) olarak alır. Dolayısıyla tekrar iş ve kaynak seviyelerini bağdaştırmak anlamlı bir faaliyet olmayacaktır. Bu sebeple, 3.2 İş ve Kaynak Seviyelerini Bağdaştırma uygulaması, genellikle Çevik bağlamda geçerli olmayan bir uygulamadır. Döngü sırasındaki herhangi bir kaynak sorununu ele almak ve uygun kaynakları derhal (yeniden) tahsis etmek veya döngüden bir çıktıyı, ilerideki bir döngüde veya sürüm planlamasında yer vermek üzere çıkarmak için günlük ayakta toplantılardan faydalanılacaktır.

Süreç Alanı 2.3 Test Gözetimi ve Kontrolü

Test Gözetimi ve Kontrolünün amacı, test ilerlemesi plandan, ürün kalitesi ise beklentilerden önemli ölçüde sapma gösterdiğinde uygun düzeltici çalışmalar yapılabilmesi için testin ilerlemesi ve ürün kalitesinin anlaşılmasını sağlamaktır.

Çevik manifesto ile beraberindeki prensiplere bakıldığında, geleneksel projelere kıyasla çevik projelerde takipten kontrole esas itibarıyla farklı olan bazı şeyler olduğu görülecektir. Her ne kadar takip ve kontrol Çevik bir projenin temel unsurları olsa da bunlar, amacın katı bir plana bağlı kalmak olduğu anlamına gelmez; hatta geçerli olan bunun aksidir zira hem manifesto hem de ilkelerin değişime açık olduğundan bahsedilmektedir. Çevik bir bağlamda Test Gözetimi ve Kontrolü, Çevik yaklaşımların önerdiği şekilde planı güncel tutmak için sürekli yeniden düzenlemek adına en iyi uygulamaları sağlamak olarak yorumlanabilir.

Test gözetimi ve kontrolü açısından bakıldığında, bu, plan odaklı olmadığımız, ancak ilerlememizi ve testten elde ettiğimiz sonuçları sürekli gözden geçirdiğimiz ve yeni ürün riskleri ortaya çıkabileceğinden planımız ile yaklaşımımızı buna uygun olarak uyarladığımız anlamına gelir. Test planı yazıya dökülmüş olsa da canlı bir varlıktır ve yeni bilgilerle karşılaştıkça veya geri bildirimler verildikçe sürekli gözden geçirilmeli ve güncellenmelidir. Çevik projeler aynı zamanda "büyük resmi" akılda tutmalı ve döngü düzeyinde olduğu kadar sürüm düzeyinde de yüksel takip ve kontrol sağlamalıdır.

Test çalışması, Çevik ekibin genel sürecine tamamen entegre bir süreç olduğu için test gözetimi ve kontrolünün de Çevik ekibin genel takip ve kontrol mekanizmalarının ayrılmaz bir parçası olduğunu unutmamak önemlidir. Sonuç olarak test uzmanları, geleneksel projelerde olduğu gibi bir test yöneticisine değil, ekibe karşı sorumludur.

TMMi® 2. seviyedeki Test Planlama Süreç Alanı hem sürüm hem de döngü planlamasına odaklandığından aynı şey test gözetimi ve kontrolü için de geçerlidir. Test gözetimi ve kontrolünün hem planlama hem de sürüm seviyesinde yapılması beklenir.

2.3.1 ÖH1 Plana göre Test İlerlemesi Gözetimi

Çevik ekiplerdeki test uzmanları, test ilerlemesini takip etmek ve kaydetmek için Çevik görev panosundaki test görevlerinin ve hikayelerin ilerlemesi ve iş bitirme çizelgeleri (burndown chart) gibi çeşitli yöntemler kullanır. Bunlar daha sonra wiki gösterge panelleri ve gösterge paneli tarzı e-postalar gibi iletişim araçları kullanılarak ve aynı zamanda ayakta toplantılarda sözlü olarak da ekibin geri kalanına iletilebilir. Ekipler, tüm sürüm boyunca ve her döngüdeki ilerlemeyi izlemek için iş bitirme çizelgelerini kullanabilir. Bir iş bitirme çizelgeleri genel itibarıyla beklenen hız ve uygulanacak özellikler listesine göre kaydedilen ilerlemeyi gösterecektir (ekibin performansı). Test ortamı kaynaklarının yetersiz ve hayati olduğu bazı durumlarda, (örneğin fonksiyonel olmayan testler için) döngü planlaması sırasında kararlaştırılan test ortamı kaynaklarının kullanımını takip etmek için belirli iş bitirme çizelgeleri kullanılır. Bir başka yaygın uygulama da test ortamı sorunlarını görev panosu aracılığıyla takip etmektir (örneğin, hikâye kartları üzerine "engellenmiş test ortamı" yazılı yapışkanlı kağıtlar yerleştirerek veya pano üzerinde ortamlar tarafından engellenmiş tüm hikayelerin, engelleri kaldırılana kadar beklediği ayrı bir sütun oluşturarak). Burada amaç, ilerlemeye engel teşkil eden bir test ortamının etkisini görünür kılmaktır.

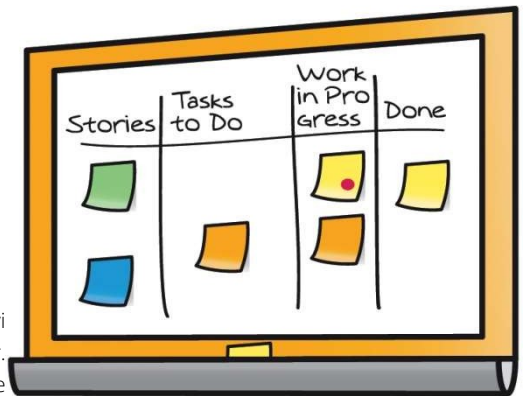
Ekipler, test çalışmasının ne safhada olduğu da dahil olmak üzere tüm ekibin mevcut durumunun anında ve detaylı bir görsel tablosunu ortaya koymak için Çevik görev panolarını kullanabilir. Hikâye kartları, geliştirme görevleri, test görevleri ve döngü planlaması sırasında oluşturulan diğer görevler, genellikle görev türünü belirtecek şekilde renk açısından uyumlu kartlar kullanılarak görev panosunda belirtilir. Döngü sırasında kaydedilen ilerleme, bu görevlerin görev panosu üzerindeki yapılacaklar, yapılmakta olanlar ve bitenler gibi sütunlara aktarılmasıyla yönetilir. Çevik ekipler, Çevik görev panolarındaki hikâye kartlarını düzenlemek için gösterge panellerini ve durum genel bakışını otomatikleştiren araçlar kullanabilir.

Ancak, bazı ekipler bireysel faaliyetler için özel görevler oluşturmayıp yalnızca hikâye kartını kullanabilir ve bu kartlara, test çalışmasının belgelenmiş olabileceği Çevik araçlara veya wiki'lere atıf yaparak yorumlar ekleyebilir. Görev panosundaki test görevleri genellikle, kullanıcı hikayeleri için tanımlanan kabul kriterleri ile ilgilidir. Bir test görevine dair test otomasyon betikleri, manuel testler ve keşif testleri geçer duruma geldiğinde görev, görev panosunun yapılanlar sütununa taşınır.

Tamamlandı Tanımı (DoD), ilerlemenin ölçüleceği çıkış kriterleri olarak işlev görür. DoD, test faaliyetleriyle de ilgili olmalı ve bir kullanıcı hikayesi testi "Tamamlandı" olarak tanımlanmadan önce karşılanması gereken tüm kriterleri göstermelidir.

Test görevleriyle ilgili test kriterlerinin, ekibin "Tamamlandı" olarak belirlemeyi kararlaştırdığı görevlerin yalnızca bir bölümünü oluşturduğu unutulmamalıdır. Tamamlandı Tanımı kriterleri genellikle birden çok seviyede (örneğin döngü seviyesinde ve sürüm seviyesinde) uygulanır. Tüm ekip, görevlerin bir sonraki aşamaya makul oranda taşınmasını sağlamak için görev panosunun durumunu çoğunlukla günlük ayakta toplantılarda olmak üzere düzenli olarak gözden geçirir. Herhangi bir görev (test görevleri de dahil) bir sonraki aşamaya geçmiyor veya çok yavaş ilerliyorsa bu durum, söz konusu görevin ilerlemesini engelleyebilecek konuların analiz edildiği bir ekip tartışmasını tetikleyecektir.

Günlük ayakta toplantıda, test uzmanları da dahil olmak üzere Çevik ekibin tüm üyeleri yer alır. Bu toplantıda mevcut durumlarını ve gerçekleşen ilerlemelerini ekibin geri kalanıyla paylaşırlar. Testin ilerlemesini engelliyor olabilecek tüm sorunlar, ekibin tamamının bu sorunlardan haberdar olabilmesi ve söz konusu sorunlar üzerinde uygun şekilde çalışabilmesi için günlük ayakta toplantılar sırasında paylaşılır. Bu şekilde risk yönetimi de bu günlük toplantılara dahil edilmiş olur. Test ortamlarındaki kullanılabilirlik eksikliği gibi test riskleri de dahil olmak üzere her tür proje riski günlük ayakta toplantı sırasında paylaşılıp tartışılabilir. (Unutmayın ki: ürün risklerini takip etmek, ürün kalitesini takip etmenin bir parçasıdır ve



bundan böyle Plan ve Beklentilere göre Ürün Kalitesinin Gözetimi özel hedefi ile birlikte tartışılacaktır.) Günlük görev yönetimi için günlük ayakta toplantılar ile görev rotasının düzeltilmesine ilişkin Çevik ekip uygulamaları, Test Gözetimi ve Kontrolü süreç alanındaki TMMi®'a özel uygulamaların amacını yerine getiren, kanıtlanmış tekniklerdir.

Çeviklik'te kilometre taşı niteliğindeki gözden geçirme bir döngünün tamamlanmasıyla yapılır. Test çalışmasının gerçekleştirilmesi (Tamamlandı Tanımına göre), döngü gözden geçirmesinin bir parçası olacaktır. Teslim edilmekte olan ürünün iş değeri ve kalitesini tartışmak üzere paydaşların katılımıyla demolar düzenlenir. Paydaşlar; döngü planlaması, döngü gözden geçirmesi (demolar) ve geriye dönük gözden geçirmelerde ürün sahibi tarafından temsil edilir. Ürün sahibi, ürün iş listesi konusundaki tartışmalar vasıtasıyla sürece dahil olur ve döngü süresince kullanıcı hikayelerinin detaylandırılması ile testlerin tasarlanması konusunda geri bildirim ve girdi sağlar. Diğer paydaşlar ise her döngünün sonundaki döngü gözden geçirmesi sırasında sürece dahil olurlar. Paydaşların bir ürün sahibi tarafından temsil edilmesi, Çevik çalışma biçiminde yerleşik bir uygulama olduğundan paydaş katılımının özel olarak takip edilmesine gerek yoktur. Çevik projelerde ürün sahibinin aktif katılımının çok önemli bir başarı faktörü olduğunu, ancak bazen uygulamada gerçekleştirilmesinin çok güç olabileceğini unutmamak gerekir. Böyle bir durum söz konusu ise günlük ayakta toplantıda belirtilerek tartışılacak ve tüm ekibin etkinliğini ve verimliliğini etkileyecek bir proje riski (yukarıda belirtildiği üzere) olarak değerlendirilecektir.

2.3.2 ÖH2 Plan ve Beklentilere göre Ürün Kalitesinin Gözetimi

Ürün kalitesi gözetimi için ilerleme gözetiminde kullanılan mekanizmaların aynaları kullanılır (bkz. ÖH1). Çeviklik'te, ürün risklerinin izlenmesindeki odak noktası, ayrıntılı risk belgelerinin incelenmesinden ziyade, düzenli toplantılarda ürün riskleri listesinin incelenmesidir. Yeni tespit edilen ürün riskleri veya değişen ürün riskleri (örneğin keşif testi sonucunda) tartışılacak ve gerekli test adımları üzerinde anlaşmaya varılacaktır. Çeşitli ürün risklerinin durumu genellikle gösterge panelindeki tablolarla gösterilir.

En iyi uygulamalardan biri de ilgili fonksiyonlute sisteme başarıyla entegre edilip test edilene kadar hiçbir fonksiyonlitenin tamamlanmış olarak kabul edilmemesidir. Bazı durumlarda, devam eden hataları ve diğer teknik borç biçimlerini gidermek için periyodik olarak güçlendirme veya stabilizasyon döngüleri gerçekleştirilir. Çevik ekipler, ürün kalitesini takip etmek ve iyileştirmek için test başarı/başarısızlık oranları, hata bulma oranları, bulunan ve düzeltilen hatalar gibi geleneksel geliştirme metodolojilerinde kaydı tutulanlara benzer hata esaslı metrikler kullanır. Bir döngü sırasında bulunan ve düzeltilen hataların sayısı ile muhtemelen bir sonraki döngünün iş listesine girecek olan düzeltilmemiş hataların sayısı günlük ayakta toplantılarda takip edilmelidir. Genel ürün kalitesini izlemek ve iyileştirmek için birçok Çevik ekip aynı zamanda, ürünün müşteri beklentilerini karşılayıp karşılamadığına dair geri bildirim almak amacıyla müşteri memnuniyeti anketlerini kullanır.

Test çıkış kriterleri [örneğin test kapsamı ve ürün kalitesine (hatalar) ilişkin], Tamamlandı Tanımının (DoD) unsurlarıdır. Kabul edilen çıkış kriterlerine uyum genellikle görev panosu aracılığıyla takip edilmekte olup burada bir hikâye, yalnızca DoD kriterlerine uygun olması halinde "Tamamlandı" olarak gösterilebilir.

Günlük ayakta toplantı, ürün kalitesi gözden geçirmelerini neredeyse sürekli olarak gerçekleştirmek için kullanılan mekanizmadır. Çeviklik'teki kilometre taşı niteliğindeki ürün kalite değerlendirmesi bir döngünün tamamlanmasıyla yapılır. Teslim edilmekte olan ürünün iş değerini ve kalitesini tartışmak üzere paydaşların katılımıyla demolar düzenlenir. Ürün kalitesi, tanımlanan DoD kalite kriterlerine göre doğrulanır ve onaylanır.

Test Planlama süreç alanının ardından, giriş izleme, askıya alma ve devam ettirme kriterleriyle ilgili bazı uygulamaların alakasız olması muhtemeldir. Daha fazla bilgi için Test Planlama süreç alanındaki özel hedef 2 Test Yaklaşımı Oluşturma'ya bakınız. Burada, bu kriterlerin neden Çevik bir ortamda uygun olmayacağı açıklanmıştır.

2.3.3 ÖH3 Düzeltici Faaliyetleri Kapanışa Kadar Yönetme

Çevik ekipler, beklentilerden sapmalar gibi bir sorunu iş bitirme tablosundan ve/veya (test) görevlerinin ve hikayelerin ilerleme kaydetmemesi gibi sorunları Çevik görev panosundan çok hızlı bir şekilde fark edeceklerdir. Bu ve bunun gibi sorunlar (örneğin testin ilerlemesini engelleyen sorunlar) ekibin tamamının bu sorunlardan haberdar olabilmesini sağlamak için günlük ayakta toplantılar sırasında paylaşılır (yukarıda belirtildiği üzere). Bu, sorunların analiz edildiği ekip tabanlı bir tartışmaya yol açmaktan farklıdır. Bu tartışmadan, ana hattın bir güncellemeye ihtiyacı olduğu (örneğin bir veya daha fazla kullanıcı hikayesinin döngüye ait iş listesinden çıkarılması), Tamamlandı Tanımının belki de çok katı olduğu veya çalışma biçiminde değişiklikler yapılması gerektiği sonucu çıkabilir. Ekip, yapılması gereken düzeltici faaliyetlere ürün sahibi ile birlikte karar verecektir. Döngüye ait iş listesinin değiştirildiği durumlarda, ürün sahibi ekibe güncellenmiş bir iş listesi sunar ve listeden çıkarılan maddeler bir sonraki döngü işlemine aktararak döngü planlaması sırasında tartışılır.

Çevik projelerde, düzeltici faaliyetlerin yönetilmesi esasen kendi kendini organize eden ekibin sorumluluğundadır. Ekip uygun düzeltici faaliyetleri tanımlayıp uygulayabilir veya herhangi bir konuyu "engel" olarak Scrum yöneticisine taşıyabilir. Scrum yöneticisi ekibi desteklemekten sorumludur. Düzeltici faaliyetlerin tartışıldığı ve yönetildiği en yaygın etkinlikler günlük ayakta toplantılar ile geçmişe dönük toplantılardır.

Kararlaştırılan düzeltici aksiyonlar, kapanışa kadar ürün iş listesi veya (döngü kapsamında) görev panosu aracılığıyla "görevler" veya "iş listesi maddeleri" olarak yönetilebilir.

Süreç Alanı 2.4 Test Tasarımı ve Yürütme

Test Tasarımı ve Yürütmenin amacı, test tasarımı özelliklerini belirleyerek, test tasarım tekniklerini kullanarak, yapılandırılmış bir test yürütme sürecini hayata geçirerek ve kapanışa kadar test olaylarını yöneterek test tasarımı ve yürütülmesi sırasındaki test süreci kapasitesini geliştirmektir.

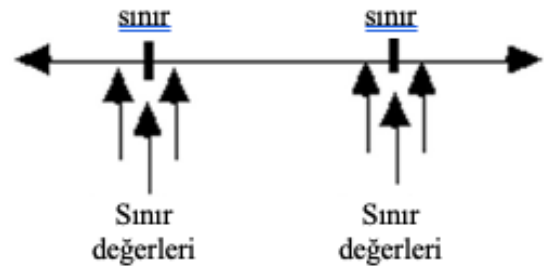
Her ne kadar temel amaç ("riskleri azaltmak ve yazılımı test etmek") Çevik ve geleneksel sıralı bir proje için aynı olsa da testin nasıl yapılacağına dair yaklaşımlar genellikle birbirinden çok farklıdır. Çeviklik'te, esneklik ve değişime cevap verebilmek önemli başlangıç noktalarıdır. Ayrıca, test analizi ve tasarımı, test uyarlaması ve test yürütmesi birbirini takip eden test fazları olmayıp daha ziyade birbirine paralel olarak, birbiriyle örtüşen biçimde ve döngüsel olarak gerçekleştirilirler. Oluşturulan test dokümantasyonunun detay seviyesi de bir diğer önemli farktır. Çevik projelerde genellikle daha fazla deneyim temelli ve hata bazı teknikler kullanılsa da gereksinim bazı test tasarım teknikleri de hala geçerli olup kullanılabilir. Birim ve entegrasyon testine daha fazla odaklanan komut ve karar testi gibi beyaz kutu test teknikleri de oldukça popülerdir. Son bir önemli fark da çeşitli test seviyelerinde daha fazla regresyon testi gerektiren regresyon riski seviyesidir. İdealde regresyon testi büyük ölçüde otomatize edilmiştir. Birçok fark mevcuttur, ancak en nihayetinde Test Tasarımı ve Yürütme süreç alanı bağlamında amaç, her zaman test senaryoları oluşturmak, ürün risklerini azaltmak, testleri koşturmak ve hataları bulmaktır.

2.4.1 ÖH1 Test Tasarım Teknikleri Kullanarak Test Analizi ve Tasarımı Yapma

Çeviklik'te, test analizi ve tasarımı ile test yürütme, genellikle bir döngü boyunca birbirine paralel olarak ilerleyen ve birbirini karşılıklı olarak destekleyen faaliyetlerdir. Sıralı yaşam döngüsü projelerinde test analizi, test uzmanlarının test esasını (örneğin gereksinimleri) gözden geçirmesi ve oluşturulmasını takiben test esasının test edilebilirliğini değerlendirmesiyle gerçekleşir. Çevik projelerde test uzmanları, iş birliği içinde kullanıcı hikayeleri oluşturan ve bunları geliştiren ekibin bir parçasıdır. Her kullanıcı hikayesi için kabul kriterleri de dahil olmak üzere gereksinimler geliştirilirken sık sık kayda geçmeden gözden geçirmeler yapılır. Bu kriterler iş kolu temsilcileri, yazılımcılar ve test uzmanlarının iş birliğiyle belirlenmektedir. Test uzmanı genellikle, kendine has bakış açısıyla eksik ayrıntıları tespit edip bunları test edilebilir hale getirerek kullanıcı hikayesini geliştirecektir. Bir test uzmanı, iş kolu temsilcilerine kullanıcı hikayesi ve ona dair kabul kriterleri hakkında açık uçlu sorular sorarak ve kullanıcı hikayesini test etme yöntemleri önererek sürece katkıda bulunabilir. Böylece test analizi açık ve ayrı bir faaliyet olmaktan çıkıp test uzmanlarının, iş birliğine dayalı kullanıcı hikayesi geliştirme kapsamındaki görevlerinin bir parçası olarak gerçekleştirdikleri dolaylı bir etkinlik olmaktadır.

Kullanıcı hikayeleri analizine dayanarak test koşulları¹ belirlenir. Test çalışması açısından bakıldığında, neyin test edilebileceğini görmek için test esasını analiz edilir; ki bunlar da test koşullarıdır. Test koşulları (bazen test durumları olarak da adlandırılır) temel olarak test edilmesi/kapsama alınması gereken "şeylerin" bir tanımıdır [Black, Van Veenendaal]. Tanımlanan kabul kriterleri, yeterince detaylı ve açık olmaları kaydıyla geleneksel test koşullarının işlevini rahatça üstlenebilir.

Test koşulları daha sonra tanımlanmış ve karşılaştırılmış kabul kriterlerini kapsaması gereken testlere dönüştürülür. Test analizini yalnızca kullanıcı hikayeleri yerine daha üst bir seviyede yapmak da çoğu zaman faydalıdır. Örneğin, kullanıcı hikayesi düzeyindekinden daha soyut ve aynı zamanda birden fazla kullanıcı hikayesini kapsayan test koşullarını belirlemek için bir özelliği veya epiği veya bir hikaye setini analiz etmek. Gereksinime dayalı test tasarım teknikleri genellikle kullanıcı hikayeleri ve kabul kriterlerinden test koşulları üretilmesinde faydalıdır.



Ancak Çeviklik'te bu test tasarım teknikleri çoğu zaman doğrudan değil de dolaylı olarak kullanılır, böylece test uzmanları deneyimleri sayesinde test tasarım tekniklerine hakim olur ve bunları bağlam içinde esnek bir şekilde kullanabilirler. Test koşulları, bir test tasarım spesifikasyonu kapsamında belgelendirildikleri geleneksel çalışma biçiminin aksine daha sade bir biçimde belgelendirileceklerdir. Bazen, özellikle de keşif testinin yaygın olarak kullanıldığı projelerde, test koşulları sadece beyin fırtınası yoluyla belirlenir ve test fikirleri olarak belgelenir (test başlatma belgesinin bir parçası olmak üzere). Test koşullarının belirlenmesi, yatay izlenebilirlik (aşağıda açıklanmaktadır) oluşturulması ve belirli test koşullarını kapsayacak şekilde otomatik testlerin geliştirildiği (otomatik) regresyon testlerinin kapsamının ele alınması için de bir zemin oluşturur.

¹ Test koşulu ifadesi bu paragrafta TMMi® uygulamalarına göre kullanılmaktadır, ancak Çevik yaklaşımda yaygın bir ifade değildir. TMMi® test koşulunu, "Bir veya daha fazla test senaryosu ile doğrulanabilen bir bileşen veya sistemin ögesi veya olgusu (örneğin, işlev, işlem, özellik, kalite özelliği veya yapısal unsur)" olarak tanımlar. Burada, Çevik ekibe ekstradan bir test dokümanı oluşturması gerektiği dayatması amaçlanmamıştır. Ancak önemli olan, ekibin hangi öğelerin (veya olguların) test edilmesi gerektiğini net bir şekilde anlamasıdır. Bazı projelerde gerçekte neleri kapsadıkları ve neleri kapsamaları gerektiği konusunda anlaşılmayan birçok test çalışması vardır. Çevik yaklaşımda TMMi® uygulamalarının test koşulları üzerindeki amacına, genellikle kabul kriterleri (gerekli detay seviyesinde belirlendikleri takdirde) ve keşif testindeki test başlatma belgeleri gibi alternatif uygulamalar aracılığıyla ulaşılır.

"Önce test" prensibinin Çeviklik'te uygulanmasıyla test koşullarını kapsayan testler, kodun geliştirilmesinden önce veya en azından onunla eş zamanlı olarak oluşturulacaktır (ve muhtemelen otomatik hale getirilecektir). Otomatik birim testi için Test GÜdümlü Geliştirme gibi bir yaklaşım düşünülebilir. Daha yüksek test seviyeleri için Davranış GÜdümlü Geliştirme (BDD) ve Kabul Testi GÜdümlü Geliştirme (ATDD) yaklaşımları, testi de destekleyen popüler Çevik yazılım geliştirme yaklaşımlarıdır. Hem BDD hem de ATDD de test otomasyonu ile son derece ilişkilidir.

Çoğu manuel testte, ekip test yürütme konusunda ilerleme kaydettikçe testler şekillenecek/iyileştirilecektir. Testler geleneksel projelerde olduğu kadar detaylı bir şekilde değil de daha çok keşif testlerinde kullanıldığı gibi test fikirleri biçiminde belgelenir. Karmaşık ve kritik alanlarda, test tasarımı ve test senaryosu geliştirme konusunda daha geleneksel bir yaklaşım benimsenerek resmi test tasarım tekniklerini kullanmak tüm risklerin dikkate alınabilmesi açısından en iyi yol olabilir. Bununla birlikte bu yaklaşımla geleneksel sıralı yaşam döngüsü ortamındaki testlere kıyasla dokümantasyon miktarı sınırlı olacaktır. Testlerin önceliklendirilmesi, esas aldıkları kullanıcı hikayesinin önceliklendirilmesine uyum sağlar. Kullanıcı hikayelerinin önceliklendirilmesi iş değerine bağlıdır, en yüksek öncelik en yüksek iş değerini ifade eder. Testlerin fonksiyonel ve fonksiyonel olmayan riskleri kapsayacak şekilde, özellikle de Çeviklik özelindeki regresyon riskini kapsayacak şekilde oluşturulması önemlidir.

Test koşullarını ve testlerin yürütülmesini desteklemek için gerekli spesifik test verileri oluşturulur. Öte yandan Çeviklik'te, geleneksel test ortamlarının aksine, gerekli test verileri ilk seferde test spesifikasyonu dokümanının bir parçası olarak belirtilmez. Bunun yerine, gerekli araçların ve/veya fonksiyonların mevcut olması kaydıyla, manuel testlerin koşumunun hemen başlamasını sağlamak için test verileri hızlıca oluşturulur. Ancak, otomatik testler için kullanılacak verilerin genellikle önceden belirlenmesi gerekecektir.

Gereksinimler, test koşulları ve testler arasında izlenebilirliğin oluşturulması ve sürdürülmesi gerekir. Test ekipleri, testlerinin bir parçası olarak çeşitli kullanıcı hikayelerini ve kabul kriterlerini dikkate aldıklarından emin olmalıdır. Resmi bir gereksinim yönetim aracına ihtiyaç duyulmasa da ekibin, gereksinimlerinin, her birine tanımlayıcıların atanmasını destekleyecek şekilde, düzenlenmesi ve yönetilmesi için desteğe ihtiyacı vardır. Böylece bu tanımlayıcılar testin mutabık kalınan kriterlere göre tamamlandığından emin olmak için kullanılabilir. Birçok Çevik organizasyonda, kullanıcı hikayeleri ilgili kabul kriteri ve daha sonra yapılacak testlerin geliştirilmesinin temelini oluşturur. Bu testler bir kez oluşturulduğunda, testlerin kendileri çoğu zaman kabul görmüş ayrıntılı gereksinimlere dönüşür. Bu yaklaşımın kullanılması, TMMi® özel uygulaması olan yatay izlenebilirliğe ilişkin gereksinim yönetiminin amacına ulaşmak için yeterli olabilir.

2.4.1 ÖH2 Test Uyarılama

Test uyarılama süreci, testlerin yürütülmesini başlatmak için gereken her şeyi yerine getirmekle ilgilidir. Genellikle, testin yürütülmesini destekleyecek test belgelerinin (örneğin test prosedürleri) geliştirilmesi için gereken çaba en aza indirilir. Bunun yerine otomatik (regresyon) test betikleri geliştirilir ve önceliklendirilir. Ayrıca regresyon testi hazırlığı, diğer test faaliyetlerine paralel olarak mümkün olan en kısa sürede başlar.

Çevik bağlamda ayrıntılı test prosedürleri oluşturulması yaygın bir uygulama değildir. Yeterli bilgi birikimine sahip bir ekibin çalışmasında, testler büyük olasılıkla çok daha yüksek bir soyutlama seviyesinde belgelenecektir. Bu belgeleme, testleri yürütenler için yeterli olacaktır, çünkü onların testleri gerçekleştirmek için gerekli alan bilgisi ve teknik bilgiye sahip oldukları varsayılmaktadır. Testleri yürütenler ekip olarak çalışırlar, dolayısıyla hem neyin, nasıl kodlandığını hem de işlevselliğin amaca nasıl uygun olduğunu daha iyi anlayacaklardır. Döngüler dahilinde ve döngü boyunca yapılan değişikliklerin seviyesi çoğu zaman geniş kapsamlı olduğundan, ayrıntılı test prosedürlerinin geliştirilmesi daha fazla bakım çalışmasına neden olacaktır. Testlerin çoğu genellikle keşif testi kullanılarak yapılır. Keşif testlerinde ayrıntılı test prosedürleri geliştirilmez; onun yerine, test uzmanı için test fikirlerini ve ilkelerini genel hatlarıyla açıklayan tek sayfalık test başlatma belgeleri oluşturulur. Ancak, testleri gerçekleştirmek için gerekli olan spesifik test verilerinin tabii ki önceden oluşturulması gerekir.

Birçok Çevik ekip otomatik testleri kullanır. En çok kullanılan yaklaşımlar, Test GÜdümlü Geliştirme (TDD), Davranış GÜdümlü Geliştirme (BDD) ve Kabul Testi GÜdümlü Geliştirme (ATDD). Bu yaklaşımlardan biri veya daha fazlası kullanılarak test uyarılama faaliyetlerinin bir parçası olarak otomatik test betikleri oluşturulacaktır.

Sürekli entegrasyon, Çevik projeler için önemli bir uygulamadır. Test çalışması, ekip tarafından gerçekleştirilecek eş zamanlı ve tamamen entegre bir faaliyettir. Bağımsız, ayrı bir faaliyet veya aşama değildir. Dolayısıyla 2.3 Alım Testi Prosedürünü Belirleme özel uygulaması, Çevik bir ekip için anlamsız olacaktır. Buna gerek yoktur ve hatta spesifik bir resmi alım testinin oluşturulması ve gerçekleştirilmesi mantıklı değildir. Ancak, sistem entegrasyon testi veya donanım/yazılım entegrasyon testi gibi bazı testlerin Çevik ekibin kapsamı dışında gerçekleştirilmesi durumunda Çevik ekibin çıktı kalitesini değerlendirmek ve teslim edilen ürünün testlerine hazır olup olmadığını belirlemek için bir alım testi belirleyebilirler.

Çevik yaklaşımda belirli bir test koşum çizelgesi oluşturulmaz, dolayısıyla 2.4 Test Koşum Çizelgesi Oluşturma özel uygulaması büyük olasılıkla gereksiz olacaktır. Bir döngü süresince testlerin hangi sırayla yürütüleceği konusunda yüksek düzeyde bir esneklik olsa da söz konusu sıralamaya çeşitli kullanıcı hikayelerinin önceliği yön verecektir. Yürütülecek çeşitli testler, döngü planlaması sırasında tanımlanacak ve Çevik görev panosu aracılığıyla test görevleri olarak ele alınacaktır.

2.4.1 ÖH3 Test Koşumu

Döngüde geliştirilen yazılım öğelerinin elbette ki test edilmesi gerekir. Çevik manifestonun takip edilmesiyle bu süreç alanındaki diğer hedeflerde olduğu gibi Test Koşumu da geleneksel bir sıralı projeye göre daha az dokümantasyon yoğunluğu ile gerçekleştirilecektir. Çoğunlukla ayrıntılı test prosedürleri ve test kayıtları üretilmez. Başta regresyon testleri olmak üzere tüm testler, mümkün olduğunca ve uygulanabilir olduğu ölçüde

otomatize edilmelidir.

Çevik bir projede, yazılım asgari düzeyde günlük olarak teslim edilir ve test, kaliteli bir ürün sunmak için karşılıklı iş birliği içinde yapılan Çevik yazılım geliştirme sürecinin ayrılmaz bir parçasıdır. Ayrı olarak tahsis edilmiş bir geliştirme ve test ekibi yoktur. Dolayısıyla, 3.1 Alım Testi Gerçekleştirme özel uygulamasında tanımlanan spesifik bir alım testinin, tamamıyla Çevik bir ortamda anlamsız olması muhtemeldir. Ancak, (çoğu zaman olduğu gibi) Çevik ekip dışında bazı testler yapıldığında, alım testi Çevik ekip dışındaki testten sorumlu test ekibi tarafından gerçekleştirilecektir. Çevik ekipte, belirli bir kullanıcı hikayesine dair kabul testi başlamadan önce tamamlanması beklenen faaliyetler olduğunu unutmayın. Bu faaliyetler genellikle Tamamlandı Tanımı'nın bir parçasıdır; örneğin, kabul edilen birim testi miktarı tamamlanmış ve başarılı olmuştur.

Test Koşumu, döngü planlaması sırasında tanımlanan önceliklere uygun olarak yapılır. Bazı testler dokümanlara dayalı bir test prosedürü kullanılarak yürütülebilir, ancak genel olarak birçok test kendi çerçevesinde keşif ve oturum tabanlı testler kullanılarak yürütülecektir. Keşif testinde, test tasarımı ve test koşumu, hazırlanmış bir test başlatma belgesinin rehberliğinde eş zamanlı olarak gerçekleşir. Test başlatma belgesi, belirli bir zaman dilimindeki test oturumu sırasında dikkate alınacak olan test hedefleri ile test koşullarını belirtir. Keşif testi sırasında, en son testlerin sonuçları bir sonraki testlere yön verir.

Döngüsel gelişmeyle birlikte, düzenlenmiş ve yapılandırılmış regresyon testine olan ihtiyaç artmaktadır. Regresyon testleri bazen manuel olarak yapılır, ancak tercihen otomatik regresyon testleri ve destek araçları kullanılarak yapılacaktır. Özellikle birim ve entegrasyon testlerindeki regresyon testi, çoğunlukla sürekli entegrasyon sürecinin bir parçasıdır. Çevik yazılım geliştirmede sürekli entegrasyon önemli bir uygulamadır. Sürekli entegrasyon temel olarak, en az günlük olarak gerçekleşen, erken ve hızlı tespitler sağlayan otomatik bir yapı ve test sürecidir. Sürekli entegrasyon, otomatik regresyon testlerini düzenli olarak çalıştırmaya, ekibe kodun kalitesi ve elde edilen kod kapsamı konusunda hızlı geri bildirim gönderilmesine olanak tanır.

Test sırasında tespit edilen olaylar günlüğe kaydedilip raporlanabilir. Çevik projelerde genellikle, tespit edilen tüm olayların kaydedilip kaydedilmemesi gerektiği konusunda bir tartışma vardır. Özellikle de ekibin bir arada bulunduğu durumlarda, test uzmanlarının yazılımcılarla olan görüşmesine istinaden hemen düzeltilip bir sonraki yapıya dahil edilebilecek olayların/hataların günlüğe kaydedilmesi gerekemeyebilir, sadece düzeltilmeleri yeterlidir. Genellikle, Çeviklik'te tespit edilen tüm olaylar günlüğe kaydedilmeyecek ve buna göre yönetilmeyecektir. Belirtildiği gibi birçoğu kayıt dışı olarak ekibin bir üyesi (test uzmanı olabilir) tarafından gündeme getirilir ve hatadan sorumlu olan yazılımcı tarafından derhal düzeltilir. Bazı veriler kaybedilir, ancak aynı zamanda ek yük de azaltılmış olur. Bazı ekipler yalnızca döngülerde gözden kaçan olayları, bazıları tespit edildikleri günde düzeltilemeyen olayları, bazıları ise yalnızca yüksek öncelikli olayları günlüğe kaydeder. Tespit edilen tüm olaylar günlüğe kaydedilmiyorsa hangilerinin kaydedilmesi gerektiğini ve hangilerinin kaydedilmemesi gerektiğini belirleyen kriterler olmalıdır. Unutmayın, amaç olayları kayıt altına almak değil, hatanın doğru şekilde düzeltilmesini sağlamaktır.

Bu tip bir kritere dair bir örnek şu şekilde olabilir: "Bir hata bir sonraki günlük ayakta toplantıdan ve günlük sürümden önce çözümlenebiliyorsa belgelenmesine gerek yoktur. Gündeme getirilen bir hata bir gün içinde giderilemiyorsa veya ekip dışındaki bir paydaşı etkiliyorsa belgelenmeli ve kayıt altına alınmalıdır."

Olaylar görev panosuna bazen görev üzerindeki yapışkanlı bir not şeklinde veya ayrı bir görev olarak kaydedilerek bir hikayenin ve o hikayeye ilişkin görevlerin tamamlanmasının engellendiğini göstermiş olur. Hataların yönetimi için görev panosunda yapışkanlı not kağıtları kullanmak alışıldık bir uygulama olup genellikle birlikte çalışan ekipler için tercih edilen bir yöntemdir. Çevik ekip daha sonra olaylar üzerinde çalışacak ve kapanışa kadar bunları yönetecektir (bu süreç alanının aşağıdaki 4. hedefine bakınız). Bazı ekipler, tespit edilen olayları hata yönetimi veya hata izleme aracı gibi bir araçta kaydetmeyi ve belgelemeyi tercih eder. Böyle bir durumda, bu tip bir araç mümkün olduğunca yalın bir biçimde kullanılmalı ve katma değeri çok az olan veya hiç olmayan test olay raporlarının herhangi bir unsurunu iletmeye zorlamamalıdır.

Çevik ekiplerde, test edilen öğelerin tanımlanmış kabul kriterlerini karşılayıp karşılamadığını ve gerçekten "tamamlandı" olarak işaretlenip işaretlenemeyeceğini belirlemek için test koşumu sırasında veri kaydı yapmak iyi bir uygulama olarak kabul edilir. Bu durum, keşif testi gibi tecrübeye dayalı teknikler uygulanırken de geçerlidir. Belgelenmesi yararlı olabilecek bilgilere örnek olarak test kapsamı (ne kadar konu kapsama alındı ve daha ne kadarının test edilmesi gerekiyor?), test sırasında yapılan gözlemler (örneğin test edilmekte olan sistem ve kullanıcı hikayesi istikrarlı görünüyor mu?), risk listesi (hangi riskler kapsama alındı ve hangileri en önemlileri arasında yer alıyor?), tespit edilen hatalar ve diğer sorunlar ile olası açık sorular gösterilebilir.

Kaydedilen bilgiler, ekibin ve paydaşların, yapılan tüm test çalışmalarının mevcut durumunu anlamasını kolaylaştıracak şekilde bir tür durum yönetimi aracı (örneğin test yönetim araçları, görev yönetim araçları, görev panosu) kayıt altına alınmalı ve/veya özetlenmelidir. Elbette ki, ekibin durum bilgisine ihtiyaç duyduğu tek alan, sadece test görevlerinin durumu değildir; ekibin daha ziyade kullanıcı hikayesinin genel durumunu bilmesi gerekir. Yalnızca test değil, ilgili tüm faaliyetler tamamlanmadığı sürece hikaye açısından hiçbir şey tamamlanmış olmaz.

2.4.2 ÖH4 Test Olaylarını Kapanışa Kadar Yönetme

Yukarıda belirtildiği üzere Çevik yaklaşımda tespit edilen tüm olaylar çoğunlukla kaydedilmez. Bu TMMi® hedefi yalnızca kaydedilen ve dolayısıyla kapanışa kadar yönetilmesi gereken olaylar için geçerlidir. Prensipte, Çeviklik'teki olayları yönetmek basittir. Bir olay Çevik görev panosunda, bir görev üzerindeki yapışkanlı not şeklinde veya ayrı bir görev olarak kaydedildiğinde bir hikayeyi ve o hikayeye ilişkin görevlerin tamamlanmasını engelleyen bir hata olarak görselleştirilmiş olur. İlerlemeyi önleyen diğer engel veya görevlerde olduğu gibi bu durumda da standart Çevik yaklaşım uygulanacaktır. İlgili olay ayakta toplantılarda tartışılacak ve ekip tarafından çözümlenmek üzere atanacak, aynı anda da görev panosundan silinecektir.

Tespit edilen olayın başka bir döngüye ertelenmesine karar verilmesi durumunda, bunlar sadece ürün için istenen başka bir seçenek (veya değişiklik) haline gelir. Bunları iş listesine eklemek ve uygun şekilde önceliklendirmek gerekir. Öncelik yeterince yüksek bir seviyeye ayarlandığında,

ilgili olay bir sonraki döngüde ekip tarafından düzeltilecektir. Önceliğin genellikle iş kolu tarafından belirlendiğini, ancak bazen teknik borç olarak adlandırılan kavramla ilgili olaylara biraz daha öncelik verilmesi gerektiğini unutmamak gerekir. Bazı ekiplerin, teknik borçla ilgili olayları/hataları gidermeye yönelik özel döngüleri vardır. Ancak bu, tavsiye edilen standart bir Çevik uygulaması değildir, çünkü borcun düzenli olarak ödenmesi gerekir.

Çevik yaklaşımda olayların nasıl ele alınabileceğine örnek olarak öncelikle ekibin bir döngünün ortasında olduğunu hayal edin. Ekip, kullanıcı hikayelerinden biri üzerinde çalışırken o hikayeyi "tamamlandı" durumuna getiremediğini tespit eder, çünkü kabul kriterlerinden biri karşılanmamıştır (örneğin geliştirme sırasında oluşan bir hata sebebiyle). Çevik ekipler genellikle bu türden hataları mevcut döngüde mümkün olan en kısa zamanda düzeltirler, aksi takdirde yarım kalmış işleri sürdürmeye devam edeceklerdir (yalın envantere eşdeğer). Bu tür olaylar görev panosu üzerinde görselleştirilecek ve bu pano aracılığıyla yönetilecektir.

İkinci olarak, döngüye ilişkin tüm hikayelerin döngünün sonunda Tamamlandı Tanımı'na göre sonuçlandırıldığını varsayın. Gözden geçirme toplantısı sırasında bir hikayeyi gösterirken müşterinin istediği gibi davranmayan bir şey olduğunu ve bu beklenmedik davranışın, döngü başlamadan önce iş kolu ile beraber tartışılmadığını düşünün. Bu durumda, üretilen yazılımın işlevselliği çalıştığı takdirde, ve muhtemelen yapılması gereken şey, olayı ek bir hikayeye dönüştürmek ve ürüne ait iş listesine eklemektir.

Bir döngü sırasında bulunan ve düzeltilen hataların sayısı ile muhtemelen bir sonraki döngünün iş listesine girecek olan düzeltilmemiş hataların sayısını günlük ayakta toplantıların yanı sıra geçmişe dönük toplantılarda da takip etmek faydalı bir uygulamadır.

Geleneksel sıralı yaşam döngüsünde genellikle, bir araya gelerek olayı gözden geçiren ve olayın çözümüne ilişkin karar veren belirli bir Yapılandırma Kontrol Kurulu (CCB) bulunur. Çevik yaklaşımda ise bu görevi yetkilendirilmiş ekip üstlenir. Elbette iş birimi temsilcisi de (örneğin ürün sahibi), olayın çözümüne ilişkin önemli bir rol oynar. Olaylar üzerinde çalışmak ekstra bir iş yüküdür; başka konular için gösterilecek dikkat ve çabayı eksiltir. Bu olayları listelemek, sınıflandırmak, sıralamak ve nihayetinde planlamak için resmi bir süreç oluşturma konusunda ne kadar çok uğraşarsak iş yükü o kadar artar. Bunun gibi karmaşık bir olay yönetimi süreci tam anlamıyla bir zaman kaybıdır. Olayların Çevik ekipler tarafından nasıl ele alındığı ve yönetildiğine dair bazı temel prensipler ortaya koymak ve bunlara göre hareket etmek daha iyidir.

Süreç Alanı 2.5 Test Ortamı

Test Ortamının amacı, testleri yönetilebilir ve tekrarlanabilir bir şekilde gerçekleştirmenin mümkün olduğu, test verilerinin de dahil olduğu, yeterli bir ortam oluşturmak ve sürdürmektir.

Test Ortamı süreç alanı vasıtasıyla testleri yönetilebilir ve tekrarlanabilir bir şekilde gerçekleştirmenin mümkün olduğu yeterli bir test ortamı (genel test verileri de dahil olmak üzere) oluşturulur ve sürdürülür. Çevik yaşam döngüsü kullanan bir yazılım geliştirme projesinde de yeterli bir test ortamı elbette ki vazgeçilmezdir. Bir döngünün kısa döngü süreleri nedeniyle test ortamının oldukça istikrarlı ve kullanılabilir olması gerekir. Test ortamındaki problemler daima döngünün ilerlemesi ve sonuçları üzerinde derhal etkili olacaktır. Dolayısıyla, test ortamındaki ve test verilerindeki yapılandırma ile değişikliklerin doğru bir şekilde yönetilmesi de hayati önem taşır.

Test ortamlarının, başka faktörlerin yanı sıra test edilen sisteme de bağlı olarak birçok şekli vardır, ancak genellikle test ortamının spesifikasyonları, uygulanması ve yönetimi, Çevik yazılım geliştirme projesine paralel olarak ilerler. Bir test ortamının uygulanması genellikle çok zaman almakta olup oldukça karmaşık olabilir, bu da bu görevi, döngünün sınırlı döngü süresi içinde gerçekleştirmeyi neredeyse imkansız hale getirir [Van der Aalst ve Davis]. Bu faaliyet aynı zamanda genel itibarıyla bir Çevik ekibi oluşturan farklı paydaşların ve mühendislerin katılımını gerektirir. Çevik metodolojiler, test ortamı veya altyapı geliştirme ve yönetimi için değil, yazılım geliştirmeyi desteklemek için geliştirilmiştir. Özetle, bu faaliyet için pek uygun değildir ve test ortamı veya altyapı geliştirme ve yönetiminin uygulanma biçimi geleneksel bir ortamdakiyle büyük ölçüde aynıdır, ancak Çeviklik'in uygulandığı durumlarda süreçte bazı değişiklikler olabilir.

2.5.1 ÖH1 Test Ortamı Gereksinimlerini Belirleme

Test ortamı gereksinimlerinin belirlenmesi projenin başlarında gerçekleştirilir. Belirlenen gereksinimler, doğruluk, kullanılabilirlik, fizibilite ve "gerçek hayattaki" bir operasyonel ortamı doğru şekilde temsil edebilme niteliklerine uyumun teyit edilmesi açısından gözden geçirilir. Erkenden oluşturulan gereksinimler, simülasyon, taklit uygulama veya sürücüler gibi gerekli test ortamı ve bileşenlerini yakalamak ve/veya geliştirmek için daha fazla zaman sağlamak gibi bir avantajı vardır.

Bazı Çevik yazılım geliştirme projelerinde, test ortamı gereksinimlerinin ortaya çıkarılması ve belirlenmesinin gerçekleştirildiği ilk döngü (döngü 0) olarak adlandırılan bir döngü uygulanır.

2.5.2 ÖH2 Test Ortamının Uyarlaması

Bazen uyarlama (veya bir kısmı), diğer faaliyetlere (örneğin eğitim, üst düzey ürün riski değerlendirmesi ve tamamlandı tanımının oluşturulması) ek olarak ilk döngüde gerçekleştirilir. Bu şekilde, ilk yazılım geliştirme döngülerinin bile halihazırda (kısmen) çalışır durumda bir test



ortamina sahip olması amaçlanmaktadır. Teknik ortamsal faaliyetler veya sorunlar da ürün iş listesinin bir parçası olabilir.

Test ortamının tasarlanması (genel test verileri de dahil olmak üzere), sıralı bir yaşam döngüsünde olduğu gibi kapsamlı bir plan aracılığıyla başlayabilir, ancak uyarlamaya en kısa sürede başlanması ve ilk döngünün başlamasını takiben test ortamının ön bir sürümünün hazır halde olması genellikle daha iyidir.

Test ortamlarında genelleşici olmak gerçekten de zordur. Bazı alanlarda taklit uygulamalar ile sürücüler hızla servis sanallaştırması ile değiştirilir, bu da genel olarak çok daha hızlı bir test ortamı uyarlamasına imkan tanır. Bazı Çevik projelerde test ortamlarının sağlanması son derece otomatiktir. Test ortamları gün yerine dakika veya saatler içinde devreye alınabilir. Günümüzde makinelerin, sunucuların ve hizmetlerin sanallaştırılması da çok daha fazla kullanılmaktadır. Bulut da gerekli ortamların sağlanmasına yardımcı olarak yine erişimi hızlandırır.

2.5.3 ÖH3 Test Ortamlarını Yönetme ve Kontrol Etme

Test ortamının yönetimi ve kontrolü Çevik bir proje süresince gerçekleşecektir. Bazen Çevik ekibinin test ortamını kendilerinin kurması ve sürdürmesi gerekir, ancak bu çoğu zaman Çevik ekibin dışındaki ayrı bir birim tarafından yapılır. Faaliyetler Çevik ekibin kendisi tarafından yapıldığında, bu durum elbette ki ekibin bu görevleri yerine getirmek için yeterli teknik bilgiye sahip olduğunu gösterir. Ayrıca, görevlerin Çevik sürecin bir parçası haline geldiği anlamına da gelir; yani, bunların bir planlama çalışmasında ele alınması, görev panosuna konulması ve herhangi bir sorun çıkması halinde muhtemelen günlük ayakta toplantılarda tartışılması gerekir.

Test Ortamı süreç alanının temel sonuçlarından biri de belirli hedefler ile uygulamaların tamamının halen geçerli olduğu ve özünde değişmedikleridir. Değişen şey, yaşam döngüsündeki zamanlamalarıdır.

3) TMMi® 3. Tanımlı

Süreç Alanı 3.1 Test Organizasyonu

Test Organizasyonu süreç alanının amacı, testten sorumlu olan ileri seviyede becerilere sahip bir grup insanı belirlemek ve organize etmektir. Test grubu, testin yanı sıra organizasyonun test süreci ile test süreci varlıkları üzerindeki iyileştirmeleri, organizasyonun mevcut test süreci ile test süreci varlıklarının güçlü ve zayıf yanları konusunda sahip olduğu detaylı bilgilere dayanarak yönetir.

3.1.1 ÖH1 Test Organizasyonunu Oluşturma

Test Organizasyonu çoğunlukla yanlış anlaşılan bir süreç alanıdır. Birçok kişi bu süreç alanını, TMMi®'in bağımsız bir test grubu veya bağımsız testler yapan bir departmana ihtiyacı olduğu şeklinde yorumlamaktadır. Bu olasılığın yanında, TMMi® gereksinimlerine uygun başka yapılanma modelleri de vardır. Testleri hazırlayan ve yürüten bağımsız test organizasyonunun yanı sıra test yeterlilik merkezi de bir seçenektir. Test yeterlilik merkezi olarak adlandırılan yapılanmanın tipik görev ve sorumlulukları, test süreçlerini oluşturmak, yönetmek ve iyileştirmek ve test deneyimi, bilgi ve becerileri ile projelere kaynak sağlamaktır. Bir test yeterlilik merkezi genel itibarıyla test metodolojileri ve süreçleri konusunda yetkilidir. Bir test uzmanının, günlük faaliyetlerde Çevik ekibin bir parçası olduğunu, ancak bunun yanında test yeterlilik merkezi veya test birliği gibi bir test organizasyonunun üyesi olmasının da yaygın olduğunu unutmamak gerekir.

Bir test organizasyonu, test ve kalite sorunları ile ilgili alanlarda liderliğe ihtiyaç duymaktadır. Böyle bir grubun personeline test uzmanları adı verilir. Test yeterlilik merkezi seçeneği Çeviklik ile iyi uyum içindedir ve testin bir disiplin olarak korunmasını ve buna göre ciddiye alınmasını sağlamaktadır. Elbette ki özel disiplinleri bulundurmaktan yana olmayan Çevik düşünce savunucuları da vardır. Bu bazıları için tercih edilen bir seçenek olsa da TMMi®'i uygulamak testin bir disiplin olarak ele alınmasını ve öyle de devam ettirilmesini gerektirir. Elbette ki bu sadece iş perspektifi açısından mantıklıysa yapılmalıdır, mantıklı olmadığı durumda da TMMi® muhtemelen tercih edilmesi gereken iyileştirme yolu değildir.

Çeviklik'te, test organizasyonu bir test birliği şeklini alabilir. Test birliği genellikle farklı Çevik ekiplerin üyelerinden oluşan, resmi olmayan ve kendi kendini yöneten bir test grubudur. Test birliğinin önemli faaliyetleri arasında bilgi paylaşımı, grup genelinde öğrenme, trend izleme vs. yer alabilir. Birlik üyeliği gönüllülük esasına dayalıdır. Pratikte test birliği çoğunlukla iki haftada bir toplanır. Test birliğine ayrıca, test alanında kariyer imkanları oluşturma, eğitim düzenleme ve test süreci iyileştirmelerinin belirlenmesi, planlanması ve uygulanması gibi daha resmi görevler de üstlenebilir. Test birliğinin büyük ölçüde gayriresmi ve özerk kalmasını sağlamak önemlidir. Test birliğinin başarılı olmasındaki önemli mecburiyet de üyelerine gerçekleştirilecek çeşitli test birliği faaliyetlerine harcanmak üzere zaman tahsis edilmiş olmasıdır.

Bağımsız test uzmanları, hata bulma konusunda çoğunlukla daha etkilidirler. Bağımsızlık ile Çeviklik'i dengelemek zordur. Bazı Çevik ekipler tamamen bağımsız ayrı test ekipleri bulundurur ve her test koşumunun son günlerinde talep üzerine test uzmanları atar. Bu yaklaşım bağımsızlığı koruyabilir ve bu test uzmanları, yazılıma dair nesnel ve tarafsız bir değerlendirme sunabilir. Ancak zaman baskıları, ürünlerdeki yeni özelliklerin anlaşılabilmesi ve iş kolu paydaşları ile yazılımcılarla ilişkilerde yaşanan sorunlar genellikle bu yaklaşımdan kaynaklı problemlere yol açar. Başka bir seçenek de projenin başlangıcında test uzmanlarının uzun vadeli olarak Çevik ekiplere tahsis edildiği bağımsız, ayrı bir test organizasyonuna sahip olunması ve böylelikle, test uzmanlarının bağımsızlıklarını korumasına ve aynı zamanda ürünle ilgili detaylı bilgi edinip diğer ekip üyeleriyle de güçlü ilişkiler geliştirmelerine imkan tanınmasıdır. Ayrıca, bu bağımsız test organizasyonunda, otomatik test araçlarının geliştirilmesi, fonksiyonel olmayan testlerin yapılması, test ortamları ile verilerin oluşturulması ve desteklenmesi ve bir test koşumu kapsamına pek iyi uymayabilecek test seviyelerinin (örneğin sistem entegrasyon testi) gerçekleştirilmesi gibi uzun vadeli ve/veya döngüden bağımsız faaliyetler üzerinde, Çevik ekipler dışında çalışacak uzmanlaşmış test mühendisleri yer alabilir. Çevik bir ekip bünyesinde de bağımsız test gerçekleştirme imkanının bulunduğunu unutmamak gerekir. Test varlıkları veya kodun yaratıcısından farklı bir kişi de ürünü test ettiği sürece bu hala bağımsızlıktır. Dolayısıyla yazılımcılar birbirlerinin kodunu, kod gözden geçirmeleri ve birim testi vasıtasıyla veya bir kullanıcının bakış açısından teste katkıda bulunan (örneğin farklı kişiler kullanılarak) iş analistleri aracılığıyla test edebilirler. Son ama en az diğerleri kadar önemli olan bağımsızlık, organizasyonel sorumluluk esaslı bir yapı ile düzenlenebilir ancak her şeyden önce doğru eleştirel zihniyete sahip olmakla ilgilidir.

3.12 ÖH2 Test Uzmanları için Test Fonksiyonları Oluşturma

Test, değerli bir meslek ve disiplin olarak görülür. Çevik ekip, test bilgisi ve becerilerine ihtiyaç duymaktadır.

Test uzmanları bunları sağlayacak ve aynı zamanda diğer ekip üyelerine de test faaliyetlerini gerçekleştirirken yol gösterecektir. Test fonksiyonları ve test kariyeri yolları, bir test eğitim programı aracılığıyla belirlenir ve desteklenir. Tipik bir Çevik test uzmanının sahip olması gereken bilgi ve beceriler, geleneksel bir organizasyondaki test uzmanına göre farklıdır. Test uzmanının tabii ki yine de "geleneksel" test bilgisi ve becerilerine sahip olması gerekir, fakat bunun yanı sıra test otomasyonuna ilişkin bilgi ve becerilere de sahip olmalı ve kod yazımı veya iş analistliği gibi test dışında görevleri de yerine getirebilmelidir. Çevik bir test uzmanı ekip içinde çalıştığından, sosyal beceriler teknik becerileri artırmak kadar önemlidir. Çevik test uzmanı, T şeklinde bir yetkinlik alanına sahip olmalıdır. Test organizasyonunda, Çevik bağlamda iyi bir test uzmanı olmak için gerekli beceri ve motivasyona sahip kişiler yer alır. Bu kişiler belirli bir test fonksiyonuna atanır. Özellikle de Çevik bağlamda, tüm ekip test faaliyetleri gerçekleştireceğinden diğer test dışı fonksiyonlara dair açıklamaları, test faaliyetleri, roller, gereken test bilgisi ve becerilerine dair beklentilerle detaylandırmak önemlidir.



3.13 ÖH3 Test Kariyeri Yolları Oluşturma

Test uzmanlarının bilgi, beceri, mevki ve kazanımlarını geliştirmelerine ve böylece kariyerlerini ilerletmelerine olanak sağlamak üzere test kariyeri yolları oluşturulur. Belirlenen test kariyeri yollarına göre test organizasyonunun her üyesi için kişisel test kariyeri geliştirme planları oluşturulur ve takip edilir. Bu özel hedef ve ona eşlik eden uygulamalar Çevik bağlamda da tamamen geçerlidir. Çeviklik'in insan odaklı yaklaşımı dikkate alındığında, bu bariz şekilde önemli bir hedeftir. Çevik ekibin, zorlu işlerini anlayan ve diğer ekip üyelerine yol gösterebilecek motive test uzmanlarına ihtiyacı vardır. Bunların birçoğu test kariyeri yollarıyla ele alınır.

Ancak Çevik bir organizasyondaki test kariyeri yollarının gerçek tanımı, sıralı bir yaşam döngüsünü takip eden bir organizasyondan biraz farklı olacaktır. Geleneksel organizasyonlardaki test kariyeri yolları genellikle dikey büyümeye (test uzmanından test yöneticisine) odaklanırken, Çevik organizasyonlardaki test kariyeri yolları daha çok yatay büyümeye odaklanır (test uzmanından kıdemli test uzmanına ve belki de test koçuna). Test yönetimi gibi çeşitli hiyerarşik işlev ve rollere genellikle daha az ihtiyaç olup Ürün Sahibi (Product Owner) veya Scrum yöneticisi (Scrum master) tarzındaki pozisyonlara yükselme ihtiyacı ise daha fazladır. Geleneksel test yönetimi görevlerinin birçoğunun Çeviklik'te hala geçerli olduğunu, ancak bunların başkaları tarafından (Scrum master, Çevik ekip gibi) üstlenildiğini unutmamak gerekir.

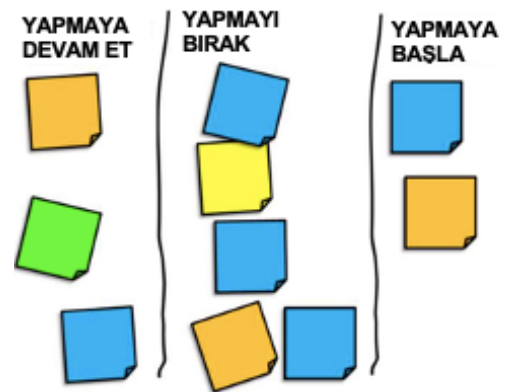
3.14 ÖH4 Test Süreci İyileştirmelerini Belirleme, Planlama ve Uygulama

Bu TMMi® hedefi ve ona ait uygulamalarla belirlenmiş olan iyileştirme süreci çoğunlukla Çevik ekibin gerçekleştirdiği geçmişe dönük toplantıda dikkate alınır. Döngüye dair geçmişe dönük bir toplantı, Çevik ekibin kendini denetlemesi ve bir sonraki döngü sırasında hayata geçirilecek iyileştirme aksiyonlarını alması için bir fırsattır. Geçmişe dönük toplantı genellikle döngü gözden geçirmesinden sonra ve bir sonraki döngü planlamasından önce gerçekleştirilir.

Ekip geçmişe dönük toplantı sırasında döngüde nelerin iyi gittiğini, nelerin iyileştirilebileceğini ve bir sonraki döngüde nelerin iyileştirilmesine eğilineceğini/öncelik verileceğini tartışır. Çevik ekip, her geçmişe dönük toplantıda iş süreçlerini iyileştirerek veya uygunsa Tamamlandı Tanımı'nı uyarlayarak ürün kalitesini artırmanın yollarını belirler ve planlar. Geçmişe dönük toplantılar, test etkinliği, verimliliği ve testlerin kalitesine odaklanan test iyileştirme kararlarının alınmasıyla sonuçlanabilir. Bu toplantılarda ayrıca, uygulamaların, kullanıcı hikayelerinin, fonksiyonların veya sistem ara yüzlerinin test edilebilirliği de ele alınabilir. Hataların kök neden analizinin yapılması, test ve geliştirme konusunda iyileştirmelerin kapısını açabilir. Geçmişe dönük toplantılarda test uzmanları önemli bir rol oynar. Onlar ekibin bir parçasıdır ve kendilerine has bakış açılarıyla katkı sağlarlar. Test uzmanları ve test uzmanı olmayanlar da dahil olmak üzere tüm ekip üyeleri hem test faaliyetlerinde hem de test dışı faaliyetlerde katkı sağlayabilirler.

Geçmişe dönük toplantının sonunda Çevik ekip, bir sonraki döngüde uygulayacağı teste ilişkin iyileştirmeleri belirlemiş olmalıdır. Bu iyileştirmelerin bir sonraki döngüde uygulanması, Çevik ekibin kendini denetlemesinin bir uyarılamasıdır. Ekip bir sonraki döngüde yapılacak en önemli iyileştirmeleri oylar, böylece test iyileştirmeleri iş listesinde bir döngü süresinden daha uzun süre kalabilir, ama sonuçta hayata geçirilir. Ekibin, yalnızca bir sonraki döngüye uygulayabileceği kadar iyileştirme üstlenmesi önemlidir, dolayısıyla her döngüde muhtemelen en fazla iki veya üç iyileştirme olmalıdır. Test uzmanları, test iyileştirmelerinin ekip içinde uygulanmasını sağlamak için bu iyileştirmeler konusunda sorumluluk almalıdır. Geçmişe dönük toplantı, bir ekibin proje ömrü boyunca sürekli gelişmesine ve iyileşmesine imkan tanıyan önemli bir mekanizmadır.

Kapsam çoğu zaman önceki döngü ile sınırlı olduğundan daha çok belirli proje sorunlarını çözmeye odaklanan küçük ama sık iyileştirmeler yapılır. Bu iyileştirmelerin odağı genellikle projeler arası öğrenme ile iyileştirmelerin kurumsallaştırılması değildir. Test süreç iyileştirmesinin nasıl



düzenlendiğine ve yönetildiğine bakıldığında muhtemelen organizasyonel düzeydeki bir test süreci grubuna daha az odaklanıldığı ve proje içindeki ekiplerin kendi kendilerini yönetmesine daha fazla önem verildiği görülecektir. Bunun kötü bir şey olmamasıyla birlikte başarılı olan yerel test iyileştirmelerinin organizasyonun geri kalanıyla paylaşıldığı ve Çevik ekibin yetki alanının dışında kalan iyileştirme alanlarının da ele alınabildiği bir mekanizma oluşturmak önemlidir. Olası bir çözüm, test organizasyonunun bir temsilcisinin, iyileştirmelerin paylaşılmasını ve uygun olduğunda organizasyon genelinde yaygınlaştırılmalarını sağlamak üzere yerel geçmişe dönük toplantılara katılmasını sağlamak olabilir. Test süreci iyileştiricisi de organizasyon genelinde test faaliyetlerini etkileyen daha geniş kapsamlı ve temel konuları üstlenebilir.

3.15 ÖH5 Organizasyonel Test Sürecini Uygulamaya Alma ve Tecrübeleri İçselleştirme

Çevik bir projede standart organizasyonel test süreci ile test süreci varlıklarına ihtiyaç olduğu sürece erişimin sağlanması önemlidir. (Standart organizasyonel test süreci ile test süreci varlıkları ve bunların Çevik projelerle ilişkisi hakkında daha fazla bilgi için Test Yaşam Döngüsü ve Entegrasyonu süreci alanına bakınız.)

Çevik projelerde organizasyonun standart test sürecinin uygulanması ve test süreci varlıklarının kullanımı, kendi kendini yöneten ekipler tarafından takip edilebilir (örneğin bu konuların geçmişe dönük toplantılarda ele alınmasıyla). Test süreci iyileştiricisi gibi test organizasyonunun bir temsilcisi, bu toplantılarda hazır bulunarak uygulama durumunu takip edebilir ve projeler arası sorunları ele alabilir. Bu kişi ayrıca Çevik ekipten test organizasyonuna aktaracağı tecrübeleri ve test iyileştirmelerini edinebilir. Tecrübeler ve test iyileştirmeleri daha sonra test süreci iyileştirme önerileri olarak sunulup buna göre yönetilebilir. Böylece Çevik ekiple yapılan testlerden elde edilen tecrübeler uygun olması durumunda standart organizasyonel (Çevik) test süreci ile test süreci varlıklarına uygulanır.

Süreç Alanı 3.2 Test Eğitim Programı

Test Eğitim Programı süreç alanının amacı, test görevleri ile rollerinin etkin ve verimli bir şekilde gerçekleştirilebilmesi için kişilerin bilgi ve becerilerinin geliştirilmesini kolaylaştıran bir eğitim programı geliştirmektir.

Test Eğitim Programının temel amacı, test uzmanları ile diğer ekip üyelerine gerekli yazılım test eğitimini sağlamaktır. Çevik manifestonun "süreçler ve araçlardan ziyade bireyler ve etkileşimlere" ilkesinin bir sonucu olarak Çevik ortamdaki süreçler genel olarak yüzeysel şekilde belgelenecek ve olası tüm senaryoları kapsamayacaktır. Bununla birlikte, kişilerin eğitimi özen gösterilmesi gereken kritik bir başarı faktörü haline gelmekte olup neredeyse yüzeysel süreçlere karşı bir önlem gibidir. Kaliteli bir eğitim programı, test çalışmasında yer alanların test bilgi ve becerilerini geliştirmeye devam etmelerini ve testle ilgili gerekli alan bilgisi ile diğer bilgi ve becerileri kazanmalarını sağlar. Kalite ve test artık ekip sorumluluğu olduğundan testle ilgili eğitiminin sadece test uzmanlarına değil, tüm Çevik ekibe sağlanması beklenir.

3.2.1 ÖH1 Organizasyonel Test Eğitim Olanakları Oluşturma

Süreç, organizasyonun stratejik test eğitim ihtiyaçlarının belirlenmesi ile başlar. Test uzmanlarına yönelik test tasarım teknikleri ve kapsam ölçütleri gibi geleneksel eğitim ihtiyaçlarının yanı sıra artık Çevik teste özel bilgi ve beceri kazanılmasına da ihtiyaç vardır. Buna ilişkin örnekler arasında Çevik manifesto ile ilkeleri, Scrum, Kanban ve XP gibi çerçeveler, test piramidi ve test çeyrekleri gibi Çevik test kavramları, Test GÜDÜMLÜ Geliştirme (TDD), Davranış GÜDÜMLÜ Geliştirme (BDD), Kabul Test GÜDÜMLÜ Geliştirme (ATDD) ve kabul kriterlerinin tanımının desteklenmesi ile bunların test edilebilirlik, test otomasyonu ve keşif testi açısından gözden geçirilmesinin de dahil olduğu kullanıcı hikayesi geliştirme yer alır. Ayrıca planlama, tahmin (örneğin poker planlama tekniği), izleme ve risk analizi (örneğin risk pokeri) gibi test yönetimine dair bazı tipik bilgi ve beceri alanları artık yalnızca test yöneticisinin değil, Çevik bir ekipte çalışan çoğu test uzmanının da bilgi ve beceri dağarcığının bir parçası olması gereken alanlar haline gelecektir. Stratejik test eğitimi ihtiyaçlarının belirlenmesi sadece test uzmanları ile sınırlı değildir, Çevik ekiplerin diğer üyelerine yönelik testle ilgili eğitim ihtiyaçlarını belirlemek de son derece önemlidir.



Diğer tüm organizasyonlarda olduğu gibi, test eğitim ihtiyaçları muhtemelen bazı projeye özgü test eğitim ihtiyaçlarını da ele alan (sade) bir eğitim planına dönüştürülecektir. Eğitim planının gözden geçirilmesi ve sorumlulukların oluşturulması gerekmektedir. Özellikle Çeviklik'te bazı becerilerin iş başında eğitim, öğle yemeği eğitimleri, koçluk ve mentorluk gibi gayriresmi yollarla etkili ve verimli bir şekilde aktarıldığını, diğer beceriler için ise daha resmi eğitimler gerektiğini unutmamak gerekir. Belirli test eğitim ihtiyaçlarını karşılamaya yönelik uygun yaklaşım, bilgi ve beceri alanına göre belirlenmelidir (örneğin keşif testi için iş başında eğitim genellikle önemli bir husustur).

3.2.2 ÖH2 Gerekli Test Eğitimini Sağlama

ÖH2'ye dair özel uygulamalar geleneksel ve Çevik bir organizasyonda büyük ölçüde aynıdır. Test görevlerini etkili bir şekilde yerine getirebilmek için gerekli eğitimi alması gereken ekip üyeleri organizasyonel eğitim planı esas alınarak belirlenir. Ardından, gerekli kaynaklar da dahil olmak üzere eğitim programları ve gerçekleştirilir. Bu süreç kulağa oldukça basit gelse de bazı Çevik organizasyonlar eğitim için zaman bulmakta zorlanır. Çevik ekipler her zaman yoğun olup döngü teslim tarihine kadar işlerini yetiştirmeye çalışmaktadırlar.

Yukarıda belirtildiği gibi gayriresmi eğitim araçları Çevik organizasyonlarda sıkça kullanılır. Örneğin, resmi eğitimin yanı sıra iş başında mentorluk ve koçluk da sürekli olarak teşvik edilir ve uygulanır. Hatta mentorluk, herkesin sorumluluğundadır ve organizasyonun tüm seviyelerinde beklenmektedir. İkili gruplar halinde çalışmak Çevik ekipler tarafından beceri geliştirme ve bilgi aktarımı amacıyla yaygın olarak kullanılan bir başka yöntemdir. Scrum kullanılırken Scrum uygulamaları konusunda ekibe koçluk yapmak Scrum Master'ın sorumluluğundadır.

Gayriresmi veya resmi olarak yapılmış olan test eğitimlerinin kayıtları oluşturulacak ve test eğitiminin etkinliği değerlendirilecektir. Alınan eğitim, katılımcıların bilgi ve becerilerinin artık test görevlerini yerine getirmek için daha yeterli olup olmadığının tartışılacağı geçmişe dönük bir toplantı kapsamında da değerlendirilebilir.

Süreç Alanı 3.3 Test Yaşam Döngüsü ve Entegrasyonu

Test Yaşam Döngüsü ve Entegrasyonunun amacı, kullanılabilir bir dizi organizasyonel test süreci varlıkları (örneğin standart bir test yaşam döngüsü) ve çalışma ortamı standartları oluşturmak ve test yaşam döngüsünü geliştirme yaşam döngüsüyle bütünleştirmek ve eşgüdümlemektir. Entegre yaşam döngüsü, bir projede test çalışmalarının erken başlamasını sağlar. Test Yaşam Döngüsü ve Entegrasyonunun amacı aynı zamanda belirlenen risklere ve planlanmış test stratejisine dayalı olarak birden fazla test seviyesinde tutarlı bir test yaklaşımı oluşturmak ve belirlenmiş test yaşam döngüsüne dayalı olarak genel bir test planı oluşturmaktır.

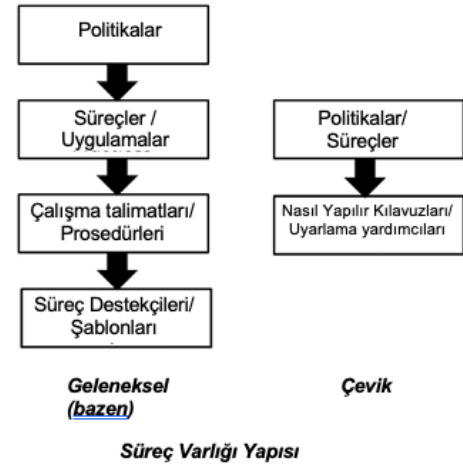
3.3.1 ÖH1 Organizasyonel Test Süreci Varlıklarını Oluşturma

Test organizasyonunun önemli sorumluluklarından biri de organizasyonun test politikası ve hedefleri doğrultusunda standart bir test süreci oluşturmak, belgelemek ve söz konusu süreci devam ettirmektir. Bu faaliyetler diğerlerinin yanında, koşullar çeşitli test türlerinin ve seviyelerinin açıklamalarını da içerir.

TMMi®, oldukça kapsamlı, belirlenmiş ve belgelenmiş test süreçlerini zorunlu kılmaya da çoğu zaman örneklerle detaylandırılan odaklanılmış şablonlar, ortak bir çalışma yöntemi oluşturmak için harika bir yaklaşımdır. Bob Galen'in burada, kontrol listeleri, şablonlar ve havuzlardan oluşan standartları Çevik yazılım testinin temel direklerinden biri olarak tanımladığı Çevik kalitesi ve testine dair üç temel ilkeye bakınız [Galen]. Şablon, ortaya çıkan bilgileri ve belgeleri toplamak için gerekli etkinlikleri tanımlar. Bir sürecin katı bir aktiviteler dizisinden oluşması gerekmemektedir. Bağımlılıklar mevcut olduğunda, bunlar şablonda notlar halinde veya bir formdaki, önkoşulu bulunan diğer alanlar tamamlanana kadar erişilemeyecek alanlarda tutulabilir. Şablonlar, bir sürecin gerçek amacını göstermeleri sebebiyle pratik değere sahiptirler. Şablonlar ayrıca "kelime kalabalığı bulunan" süreç belgelerinde sıklıkla rastlanan belirsizlikleri de önler. Çevik bir organizasyon olarak etkin süreçleriniz varsa bunları yalın bir şekilde belgelemeniz ve belki birkaç şey eklemeniz gerekebilir. Süreç oluştururken Çevik organizasyonda sıklıkla kullanılan bazı temel ilkeler şunlardır:

- Sürece dair "yapılması gerekenler" kılavuzlardan ayrı olarak gruplanır
- Hiçbir süreç iki sayfadan fazla olmaz (hedef, esnek kural)
- Büyüklük ve ölçekleri ne olursa olsun tüm projelerde süreçler içinde, "nasıl yapılır" bilgisinin veya kullanılan araçlara ilişkin bilgilerin yer almasına karar verilmedikçe süreçler bu bilgileri içermez.
- Uyarılama/planlama seçenekleri ile "nasıl yapılır" bilgileri ayrı kılavuzlarda yer alır. "Nasıl yapılır" bilgisi konusunda, yalnızca mevcut bir kitaba veya yazıya atıfta bulunulmasına karar verilebileceğini unutmamak gerekir.

Pek çok büyük organizasyonda politikalar, süreçler/uygulamalar, çalışma talimatları/prosedürleri ve süreç destekçileri/şablonları gibi dört seviyede süreç varlığı bulunması alışıldık bir durumdur. Bu durum, TMMi®'in gerektirdiği herhangi bir şeyden değil, birçok büyük kuruluşun süreç varlıklarını uygulamayı tercih ettiği yöntemden kaynaklanır. Süreç varlıklarının seçimi organizasyona bağlı olsa da Çevik organizasyonların çoğu iki seviyeli süreç varlıklarının yeterli olduğunu keşfetmiştir. İki seviyeli süreç varlıkları, sürecin işletilmesi için "ne yapılması gerektiğini" açıklayan ilişkili süreç tanımının bir politika beyanıyla birleştirilmesiyle oluşturulur. İkinci seviye, sürecin işletilmesi ve uyarlanması için gereken "nasıl yapılır" yönergelerini içerir. Bu seviye, sürecin uyarlanmasına yardımcı olan ve ekibe, tanımlı çerçeve içinde tam olarak nasıl çalışacağına karar vermesi için özerklik sağlayan araçlar olarak görülebilecek olup genellikle destekleyici şablonları içerir. Birçok Çevik organizasyonda adım adım açıklanan prosedürlerin yerini araç kılavuzları ile eğitim/mentorluk almıştır. Yukarıda belirtildiği üzere test başlatma belgesi şablonu, test oturumu çizelgesi şablonu veya test planı şablonu gibi bir şablon, şablonla ifade edilen gerekli süreç etkinliklerini içeren bir süreç olarak hizmet edebilir. Bu, Etkin Çevik süreç tanımları geliştirmek için gözlemlenen yaygın bir tekniktir. [McMahon]



Organizasyonun standart test süreçleri grubu, belirli tanımlanmış süreçlerini oluşturmak üzere projeler tarafından özelleştirilebilir. ÖU1.3 Uyarılama kriterleri ile yönergeleri oluşturma kapsamında, günümüzde birçok kuruluş yukarıdan aşağıya terzi yaklaşımı olarak da adlandırılan bir yaklaşım kullanmaktadır. Bu yaklaşım kişilerin, bir şeye neden ihtiyaç olmadığını açıklamak için çaba göstermesini gerektirmektedir. Böyle bir yaklaşım da en basit projelere dahi müthiş bir iş yükü getirmekte olup Çevik dostu bir yaklaşım değildir. Yukarıdan aşağıya terzi yaklaşımı yapıldığında ne kadar ileri gidilebilir? Belirlenmiş bir minimum seviye var mıdır? Ancak minimum seviyenin herkesin yapması gereken şey olduğu bir minimum seviyeden başlarsanız "yapılması gereken" yaklaşımı riskini ortadan kaldırmış olursunuz. Aşağıdan yukarıya terzi yaklaşımı tarzında bir yaklaşım ise Çevik çerçevelere çok daha uygun olup aynı zamanda TMMi®'la da tamamen uyumludur. Bu tarz yaklaşım, çevikliklerini korumak isteyen Çevik organizasyonlar veya çevikliklerini artırmak isteyen yüksek olgunluktaki test organizasyonları için son derece güçlü bir mekanizmadır. Bir dizi basit kriter, bu yaklaşım hayata geçirilirken doğru soruların sorulmasına yardımcı olabilir. Yardımcı kriterler olmadığı takdirde, bazıları çok fazla yukarı uyarılama yapma eğiliminde olur. .

Standart test sürecinin uygulamaya alınmasını desteklemek için genellikle bir wiki üzerinde, test uzmanlarının günlük işlerinde kendilerine yardımcı olacak örnek şablonları, en iyi uygulamaları ve kontrol listelerini bulabilecekleri bir test süreci varlık kütüphanesi oluşturulacak ve sürdürülecektir. Test süreci varlık kütüphanesi oluşturmak, ekipler test faaliyetleri sırasında sağlanan varlıkları kullanırken katma değer elde ettikleri sürece Çevik bağlam için de mantıklıdır. Tahminler, kapsam, kalite ölçüleri gibi test verilerinin toplandığı ve ekiplerin kullanımına sunulduğu bir test süreci veritabanı için de neredeyse aynı mantık geçerlidir. Ekipler arası öğrenme ile deneyimlerin paylaşımına olanak sağlamak için toplanan ve kullanıma sunulan her şeyin ekipler için katma değer sağladığından emin olmak gerekir.

Elbette Çevik bağlamda da çalışma ortamı standartları, proje çalışma ortamlarının oluşturulmasına rehberlik etmek için kullanılır.

3.3.2 ÖH2 Test Yaşam Döngüsü Modellerini Yazılım Geliştirme Modelleriyle Entegre Etme

Standart test yaşam döngüsü modeli, çeşitli test seviyeleri için ana faaliyetler ile çıktıları tanımlar. Standart test yaşam döngüsü modeli, test faaliyetlerini aşamalandırma, kilometre taşları, çıktılar ve faaliyetler açısından entegre edecek şekilde yazılım geliştirme yaşam döngüsü modelleri ile uyumludur. Yaşam döngüsü entegrasyonu, projelerde testlerin erkenden sürece dahil edilmesini sağlayacak şekilde yapılır. Tüm bunlar, Çevik bir yaşam döngüsünün benimsenmiş olması halinde sanki yapılacak hiçbir şey olmadığını, her şeyin halihazırda halledilmiş olduğunu ifade ediyor gibi görünebilir. Çeviklik'te, geliştirme ile test faaliyeti yaşam döngüsüne tamamen entegre olmuş olmalı ve test faaliyeti de mümkün olan en erken zamanda gerçekleştiriliyor olmalıdır.

Bu her ne kadar Çevik metodoloji kullanan birçok organizasyon için geçerli olsa da bu açıdan zorlanan organizasyonlar da vardır. Bu organizasyonların döngüleri, testin ancak döngünün sonlarına doğru başladığı, genellikle zaman baskısı altında gerçekleştirildiği ve hatta bazen kısa kesildiği, geliştirmeye dair mini bir V modeli yaşam döngüsüne benzer. Bu tabii ki entegre bir Çevik yazılım geliştirme ve test yaşam döngüsünün olması gerektiği gibi bir uygulaması değildir. Ancak yine de "Çevik gibi" olan bazı organizasyonlar için bu özel hedefin iyileştirmeye açık bir alan olduğunu gösterir. Testin döngünün erken aşamalarında sürece nasıl dahil edilip uygulanabileceğini kararlaştırarak ortak bir anlayış oluşturulması önemlidir.

Dolayısıyla bu özel hedef, Çevik yazılım geliştirme prensibini uygulayan bir organizasyon için de geçerlidir. Doğru yapılması halinde geliştirme ile test faaliyeti tamamen birbirine entegre olacak ve bunun nasıl yapıldığı konusunda ortak bir anlayış olacaktır. Bu gibi durumlarda başka bir şey yapılmasına gerek yoktur. Başka durumlarda ise yukarıda açıklandığı üzere bu özel hedef iyileştirmeye açık çok önemli bir alan olup bir döngüde gerçekleştirilen geliştirme ve test faaliyetlerinin birbiriyle çok daha uyumlu olmasını sağlamalıdır.

3.3.3 ÖH3 Master Test Planı Oluşturma

TMMi® 3. seviyede test faaliyeti, tüm test seviyelerindeki test görevleri, sorumlulukları ve test yaklaşımının koordinasyonunu belirleyen master test planıyla ilgilidir. Master test planı, çeşitli test seviyeleri arasındaki gereksiz test fazlalığını veya testlerden vazgeçilmesini önlemekte olup genel test sürecinin verimliliğini ve kalitesini önemli ölçüde artırabilir. Master test planında, gerçekleştirilecek belirli seviyeler ve bu seviyeler arasındaki ilişki de dahil olmak üzere belirli bir projeye dair test stratejisinin nasıl uygulanacağı açıklanır. TMMi® 2. seviyede test planlaması, Test Planlama süreç alanı kapsamında hem sürüm hem de döngü seviyesinde halihazırda ele alınmış durumdadır. Çevik bir ekip genellikle, tamamı sürüm ve döngü planlaması kapsamında uygun şekilde ele alınması gereken birden çok test seviyesi gerçekleştirir.

Ancak Çevik yaşam döngüsü modeline göre çalışılan bazı projelerde Çevik ekip bünyesinde gerçekleştirilen testlerin yanı sıra bazı testler ekip dışında yürütüldüğünden bir nevi hibrit yaklaşım benimsenmiştir. Bu gibi durumlarda master test planı Çevik ekiplerden, onların kapsamı dışında gerçekleştirilen fonksiyonel olmayan testler (güvenlik, performans vb.), donanım/yazılım entegrasyon testi, sistem entegrasyon testi veya beta testi gibi test seviyeleriyle olan ilişkiyi tanımlayıp yönetecektir. Bu özel hedefe ilişkin tüm özel uygulamalar da genellikle dolayısıyla geçerli olacaktır. Özellikle daha büyük projelerde ve organizasyonlar büyüdükçe genel bir master test planını ve ilgili kararları tutarlı bir biçimde belgelemek daha büyük bir değere sahip olmaya başlar. Bu özellikle bir organizasyonun ölçeklendirilmiş Çevik çerçeveler kullandığı durumlarda daha da belirgindir.

Tüm test çalışmalarının Çevik ekipler tarafından gerçekleştirildiği durumlarda özel hazırlanmış master test planının bir katma değeri yoktur ve dolayısıyla bu TMMi® hedefi bu gibi durumlarda büyük olasılıkla uygulanabilir olmayacaktır. Bu durumda, sürüm ve döngü seviyesinde yapılan planlamanın gerekli tüm test hususlarını kapsaması beklenir.

Süreç Alanı 3.4 Fonksiyonel Olmayan Test

Fonksiyonel Olmayan Test süreç alanının amacı, test planlama, test tasarımı ve test yürütme sırasında fonksiyonel olmayan testi de kapsayacak şekilde test süreci olanaklarını iyileştirmektir.

Fonksiyonel olmayan testin önemli olup olmaması, uygulanan yaşam döngüsünden bağımsızdır. Geliştirilmekte olan ürün, hangi fonksiyonel olmayan unsurların testle ilgili olduğunu belirleyecektir. Bu durum tabii ki hem sıralı yaşam döngüsü modelleri hem de Çevik yazılım geliştirme modelleri için geçerlidir. Dolayısıyla, fonksiyonel olmayan testler de Çevik yazılım geliştirme için geçerlidir.



Ancak, fonksiyonel olmayan unsurların niteliğine ve kullanılan test yaklaşımına bağlı olarak, performans ve güvenilirlik testi gibi bazı fonksiyonel olmayan testler geleneksel bir şekilde yapıldıklarında, kısa bir döngüde test edilemezler. Bu fonksiyonel olmayan testlerin kurulumu, düzenlemesi ve gerçekleştirilmesi kullanılan yöntem ve tekniklere bağlı olarak haftalar alabileceğinden iş yapma zamanları döngülerin ilerleme ritmine daha az uygundur. Bu yüzden, fonksiyonel olmayan özelliklerin test edilmesi için çoğunlukla farklı bir yaklaşım gerekir. Bu, çeviklik açısından iyi yöndeki temel değişikliklerinden biridir, çünkü kalite özellikleri döngüler boyunca erkenden test edilir ve geleneksel bir yaşam döngüsü modelinde olduğu gibi bunların testi sona bırakılmaz.

Tam bir fonksiyonel olmayan test yapmak için bileşenlerin/fonksiyonların tümü döngüden elde edilemeyebilir, ancak bu testin sonuçları, kalite problemlerinin erken belirtilerini görmek için kullanılabilir.

Kullanıcı hikayeleri, kullanıcılar ve müşteriler için doğru ürünün geliştirilmesini sağlamak açısından hem fonksiyonel hem de fonksiyonel olmayan öğeleri içermelidir. ISO 25010 kalite standartları, gereksinimlerin yapılandırılmasına yardımcı olabilir; test uzmanları bu fonksiyonel olmayan unsurları dikkate almalıdır [ISO25010].

3.4.1 ÖH1 Fonksiyonel Olmayan Ürün Riski Değerlendirmesi Yapma

Test Planlama süreç alanı kapsamındaki ÖH1 Risk Değerlendirmesi Yapma özel hedefinde tanımlandığı ve açıklandığı üzere Çevik projelere dair ürün risk oturumu/oturumları artık fonksiyonel olmayan unsurlar ve riskleri de içerecek şekilde açıkça genişletilecektir. Sürüm düzeyindeki risk değerlendirme, artık fonksiyonel olmayan test çeşidinde de ürün vizyonuna bağlı olarak gerçekleştirilebilir. Döngü seviyesinde ise bu değerlendirme, fonksiyonel olmayan gereksinimleri ana bir girdi olarak tanımlayan kullanıcı hikayeleri kullanılarak yapılır.

Ürün sahibi dahil tüm ekip üyeleri ve muhtemelen bazı diğer paydaşların da tercihen ürün riski oturumlarına katılması beklenmektedir. Fonksiyonel olmayan bazı alanlarda ise yardımcı olmaları için uzmanlara ihtiyaç duyulabilir. Ürün risk oturumları genellikle önceliklendirilmiş fonksiyonel (ve fonksiyonel olmayan) ürün risklerinin belgelenmiş bir listesiyle sonuçlanır. Test Planlama süreç alanında belirtildiği gibi, Çevik projelerde hem kullanılan süreç hem de ortaya çıkan dokümantasyon, sıralı bir yaşam döngüsü modelini takip eden geleneksel bir projeye kıyasla çok daha yüzeysel olacaktır.

3.4.2 ÖH2 Fonksiyonel Olmayan Test Yaklaşımı Oluşturma

Tanımlanmış ve önceliklendirilmiş fonksiyonel olmayan ürün risklerini azaltmak üzere fonksiyonel olmayan ilgili kalite özellikleri için bir test yaklaşımı belirlenir. Belirli bir test döngüsü için test edilecek fonksiyonel olmayan özellikler döngü planlaması sırasında saptanır. Test edilecek fonksiyonel olmayan özelliklerin öncelikli listesi, genellikle bu döngüde test edilecek kullanıcı hikayeleri ile ilgilidir. Ek testler gerektiren döngü sırasında yeni fonksiyonel olmayan ürün riskleri ortaya çıkabilir. İlave test gerektiren yeni fonksiyonel olmayan ürün riskleri gibi konular genellikle günlük ayakta toplantılarda tartışılmaktadır.

Fonksiyonel olmayan riskleri azaltmak için döngü seviyesinde belirlenen fonksiyonel olmayan test yaklaşımı genel itibarıyla fonksiyonel olmayan risklerin seviyesi ve çeşidine göre uygun fonksiyonel olmayan test yöntemlerinin ve test tekniğinin/tekniklerinin oluşturulmasını kapsayacaktır. Genellikle destekleyici araçların kullanımı ile fonksiyonel olmayan testlere yönelik test otomasyonu yaklaşımı konularını da ele alacaktır. Sürüm seviyesindeki fonksiyonel olmayan teste yönelik test yaklaşımı çok daha yüksek bir düzeyde olacağından program veya organizasyon düzeyinde belirlenmiş test stratejisi ile varsa oluşturulmuş master test planını da esas almalıdır. (Master test planı oluşturma ya da oluşturmamaya dair gerekçe için Test Yaşam Döngüsü ve Entegrasyonu süreç alanındaki ÖH3 Master Test Planı Oluşturma konu başlığına bakınız). Hem sürüm hem de döngü düzeyindeki fonksiyonel olmayan test yaklaşımı, genel test yaklaşımının bir parçası olup ekip/proje wiki'sinde tutulacak veya ortaya konulacaktır.

Fonksiyonel olmayan çıkış kriterleri Tamamlandı Tanımı (DoD) olarak bilinen uygulamanın bir parçasıdır. DoD'un fonksiyonel olmayan testlerle ilgili belirli kriterlere sahip olması önemlidir; [örneğin Arızalar Arası Ortalama Süre (MTBF) veya "ön yüz web sayfaları Açık Web Uygulaması Projesi'nin (OWASP) ilk 10 risk listesi için test edilmiştir"] ifadesinin bulunması önemlidir.

Döngü, kararlaştırılmış fonksiyonel olmayan kullanıcı hikayeleri grubunun (bunların kabul kriterleri de dahil olmak üzere) uygulanmasıyla sonuçlanmalı ve DoD'de tanımlandığı şekilde fonksiyonel olmayan test çıkış kriterlerini karşılamalıdır. Fonksiyonel olmayan özelliklerin, fonksiyonel kullanıcı hikayelerine dair kabul kriterlerinin bir parçası olabileceğini ve mutlaka ayrı fonksiyonel olmayan kullanıcı hikayeleri olarak belirlenmesi gerektiğini unutmamak gerekir. Döngü düzeyindeki Tamamlandı Tanımının yanı sıra birden fazla döngüyü kapsayan sürüm seviyesinde de çoğu zaman bir DoD bulunmaktadır. Ayrıca, sürüm seviyesindeki DoD de, fonksiyonel olmayan özelliklere ilişkin kriterler de bulunabilir.

3.4.3 ÖH3 Fonksiyonel Olmayan Test Analizi ve Tasarımı Yapma

Bu özel hedef büyük ölçüde aynı uygulamaları içermektedir, ancak bu kez, Test Tasarımı ve Yürütmesi süreç alanındaki ÖH1 Test Tasarım Tekniklerini kullanarak Test Analizi ve Tasarımı Gerçekleştirme özel hedefinde olduğu gibi fonksiyonel olmayan bir bakış açısı söz konusudur. Test analizi ve tasarımı sırasında, fonksiyonel olmayan testlere dair test yaklaşımı, somut test koşullarına ve testlere dönüştürülür. Çeviklik'te, test analizi ve tasarımı ile test koşumu, genellikle bir döngü boyunca birbirine paralel olarak ilerleyen ve birbirini karşılıklı olarak destekleyen faaliyetlerdir. Bu durum fonksiyonel olmayan testlerin birçoğu için de geçerlidir. Dolayısıyla fonksiyonel olmayan test analizi açık ve ayrı bir faaliyet olmayıp daha çok test uzmanlarının, iş birliğine dayalı kullanıcı hikayesi geliştirme kapsamındaki görevlerinin bir parçası olarak gerçekleştirdikleri dolaylı bir etkinliktir.

Fonksiyonel olmayan kullanıcı hikayelerinin analizi esas alınarak fonksiyonel olmayan test koşulları oluşturulur. Test koşulları temel olarak test edilmesi/dikkate alınması gereken "şeylerin" bir tespittir. Belirlenen kabul kriterleri, yeterince detaylı ve açık olmaları kaydıyla geleneksel test koşullarının işlevini üstlenecektir. Kabul kriterleri daha sonra fonksiyonel olmayan testlere dönüştürülür. Fonksiyonel olmayan testlerde, test analizini yalnızca kullanıcı hikayeleri yerine daha üst düzeyde gerçekleştirmek çoğu zaman faydalıdır. Kullanıcı hikayesi düzeyindekilerden daha soyutlanmış olan ve aynı zamanda birden fazla kullanıcı hikayesini kapsayan fonksiyonel olmayan test koşullarını belirlemek için bir özelliği veya epiği veya bir hikaye koleksiyonunu analiz etmek bunun bir örneği olabilir. "Önce test" prensibinin Çeviklik'te uygulanmasıyla fonksiyonel olmayan test koşulları grubunu kapsayan fonksiyonel olmayan testler, kodun geliştirilmesinden önce veya en azından onunla eş zamanlı olarak belirlenecektir (ve muhtemelen otomatik hale getirilecektir).

Çoğu manuel, fonksiyonel olmayan testte, ekip fonksiyonel olmayan test koşumu konusunda ilerleme kaydettikçe testler tasarlanacaktır/iyleştirilecektir. Testler çoğunlukla geleneksel projelerde olduğu kadar ayrıntılı olarak değil de daha çok keşif testi kullanılan durumlardaki gibi test fikirleri biçiminde belirlenir. Karmaşık ve kritik fonksiyonel olmayan alanlarda, test tasarımı ve test senaryosu geliştirme konusunda daha geleneksel bir yaklaşım göstererek resmi fonksiyonel olmayan test tasarım tekniklerini kullanmak, tüm risklerin dikkate alınabilmesi açısından en iyi yol olabilir. Ancak bu yaklaşımda da dokümantasyon miktarı, geleneksel bir sıralı yaşam döngüsü ortamındaki fonksiyonel olmayan testlere kıyasla sınırlı olacaktır. Fonksiyonel olmayan testlerin önceliklendirilmesi genel olarak, esas aldıkları kullanıcı hikayesinin önceliklendirilmesini takip eder. Bununla birlikte önceliklendirme, belirli bir fonksiyonel olmayan testin hazırlanması ve gerçekleştirilmesi için gereken süreye göre de belirlenebilir.

Fonksiyonel olmayan testlerin yürütülmesini desteklemek için gereken belirli test verileri belirlenir. Çeviklik'te ihtiyaç duyulan test verileri genellikle ilk başta bir test spesifikasyon belgesinde tam olarak belirtilmez. Belirtilmediği durumlarda da çoğunlukla kullanıcı hikayesinin bir parçası olarak kayıt altına alınır. Ancak, gerekli araçlar ve/veya işlevselliklerin mevcut olması durumunda fonksiyonel olmayan testleri desteklemek için ihtiyaç duyulan test verileri, çoğunlukla fonksiyonel olmayan manuel testlerin yürütülmesine hızla başlanmasını sağlamak için anında oluşturulur. Fonksiyonel olmayan otomatik testlerde ise bunun aksine verilerin genellikle önceden belirlenmesi gerekir.

Fonksiyonel olmayan gereksinimler, test koşulları ve testler arasında izlenebilirliğin oluşturulması ve sürdürülmesi gerekir. Ekiplerin, testleri kapsamında çeşitli fonksiyonel olmayan kullanıcı hikayelerini ve kabul kriterlerini dikkate aldıklarından emin olmaları gerekir. Birçok Çevik organizasyonda, kullanıcı hikayeleri bir dizi ilgili kabul kriteri ile daha sonra yapılacak testlerin geliştirilmesinin temelini oluşturur. Bu yaklaşım kullanılarak gereksinimlerden teste kadar yata izlenebilirlik elde edilebilir.

3.4.4 ÖH4 Fonksiyonel Olmayan Test Uyarlaması

Test uyarlama, test koşumunu başlatmak için gereken her şeyi yerine getirmekle ilgilidir. Genellikle, test koşumunu destekleyecek test prosedürleri gibi test belgelerinin geliştirilmesi için gereken çaba en aza indirilir. Bunun yerine otomatik (regresyon) test senaryoları geliştirilir ve önceliklendirilir.

Fonksiyonel olmayan test uyarlaması, Test Tasarımı ve Yürütme süreç alanı kapsamındaki ÖH2 Test Uyarlamayı Gerçekleştirme özel hedefinde daha önce açıklanan uygulamaların birçoğunu takip edecektir.

Özel olarak ne yapılması gerektiği ve fonksiyonel olmayan test uyarlamasının nasıl yapıldığı, büyük ölçüde belirlenen yaklaşıma, kullanılan tekniklere ve hangi fonksiyonel olmayan özelliklerin hangi seviyede test edilmesi gerektiğine bağlıdır. Örneğin, daha az hazırlık gerektiren keşif testi, kullanılabilirlik testi için uygun olabilir, fakat çoğu zaman kapsamlı güvenilirlik ve performans testleri için pek uygun değildir.

Bazı fonksiyonel olmayan kalite standartları açısından test verilerinin varlığı çok önemli olup bunların test uyarlaması sırasında oluşturulmaları gerekir. Bu durum büyük ölçüde geleneksel projeler için de geçerli olup belki tek fark bunların daha özet bir biçimde belgelenmesi ve böylelikle regresyon testinde yeniden kullanıma imkan sağlanması olabilir.

Elbette ki test uyarlaması ile hazırlığı, diğer test faaliyetlerine paralel olarak en kısa zamanda başlayacaktır. Burada ayrı bir safha söz konusu olmayıp daha ziyade, verimli ve etkin bir test koşumu gerçekleştirilebilmesi için yapılması gereken bir grup faaliyetten (görev panosunda listelenmiş olan) bahsedilebilir.

3.4.5 ÖH5 Fonksiyonel Olmayan Test Koşumu

Bu süreç alanındaki önceki hedefte olduğu gibi, büyük ölçüde bu dokümanda daha önce üzerinde durulan, Test Tasarımı ve Yürütme süreç alanı kapsamındaki ÖH3 Test Yürütmesi Gerçekleştirme özel hedefine atıfta bulunacağız. Çevik ortamda fonksiyonel olmayan testlerin yürütülmesi, test olaylarının bildirilmesi ve test günlüklerinin tutulması uygulamaları, Çevik ortamdaki fonksiyonel test uygulamalarıyla temel olarak aynıdır. Bunlar geleneksel bir projede olduğundan çok daha düşük bir dokümantasyon yoğunluğunda yapılacaktır. Çoğu zaman ayrıntılı test prosedürleri ve test kayıtları üretilmeyecektir.

Fonksiyonel olmayan testlerin yürütmesi, döngü planlaması sırasında tanımlanan önceliklere uygun olarak yapılır. Bazı testler dokümante edilmiş test prosedürü kullanılarak yürütülebilir, fakat aynı zamanda çerçeve olarak keşif ve oturum tabanlı testler kullanılarak birçok fonksiyonel olmayan test de yürütülecektir. Döngüsel geliştirmede, regresyon testi düzenleme ve yapılandırmasına daha fazla ihtiyaç vardır. Bu bazen manuel olarak yapılır, ancak otomatik regresyon testleri ile destek araçları kullanılarak yapılması tercih edilir. Regresyon testi elbette sistemin, test edilmesi önemli olarak belirlenmiş fonksiyonel olmayan unsurları için de geçerlidir.

Test sırasında tespit edilen fonksiyonel olmayan olaylar ekip tarafından günlüğe kaydedilip raporlanabilir. Çevik projelerde genellikle, tespit edilen tüm olayların kaydedilmesinin gerçekten gerekli olup olmadığı konusunda bir tartışma mevcuttur. Bazı ekipler yalnızca döngülerde gözden kaçan olayları, bazıları tespit edildikleri günde düzeltilemeyen olayları, bazıları ise yalnızca yüksek öncelikli olayları günlüğe kaydeder. Tespit edilen tüm olaylar günlüğe kaydedilmiyorsa hangi olayların günlüğe kaydedilmesi gerektiğini ve hangilerinin kaydedilmesi gerekmediğini belirleyen kriterler olmalıdır. Olaylar bazen Çevik görev panosunda, bir görev üzerine yerleştirilen yapışkanlı bir kağıt şeklinde veya ayrı bir görev olarak kaydedilir, böylelikle söz konusu olayın, bir hikayeyi ve o hikayeye ilişkin görevlerin tamamlanmasını engellediği gösterilmiş olur. Bazı ekipler, hata yönetimi veya hata takip aracı gibi bir araç kullanılarak tespit edilen olayları kaydetmeyi ve belgelemeyi tercih eder. Böyle bir durumda, bu tip bir araç mümkün olduğunca yalın bir biçimde kullanılmalı ve katma değeri çok az olan veya hiç olmayan fonksiyonel olmayan test olay raporlarının herhangi bir unsurunu iletmeye zorlamamalıdır.

Çevik ekiplerde de test edilen öğelerin belirlenmiş kabul kriterlerini karşılayıp karşılamadığını ve gerçekten de "tamamlandı" olarak işaretlenip işaretlenemeyeceğini belirlemek için fonksiyonel olmayan test yürütme sırasında veri kaydı yapmak iyi bir uygulama olarak kabul edilir. Günlüğe kaydedilen bilgiler, ekibin ve paydaşların, yapılan tüm test çalışmalarının mevcut durumunu anlamasını kolaylaştıracak şekilde bir tür durum yönetimi aracıdır (örneğin test yönetim araçları, görev yönetim araçları, görev panosu) kayıt altına alınmalı ve/veya özetlenmelidir.

Süreç Alanı 3.5 Eş-Gözden Geçirmeler

Eş-Gözden Geçirme süreç alanının amacı, çalışma ürünlerinin, kendileri için belirlenen gereksinimleri karşıladığını doğrulamak ve seçili çalışma ürünlerindeki hataları erkenden ve verimli bir biçimde düzeltmektir. Önemli bir sonucu ise çalışma ürünlerinin ve önlenebilecek hataların daha iyi anlaşılmasını sağlamasıdır.

3.5.1 ÖH1 Eş-Gözden Geçirme Yaklaşımı Oluşturma

Çevik ekipler, insanların bir ürün hakkında geri bildirim sağlamak üzere bir araya geldiği belirli bir zaman dilimi olmadığından genellikle resmi eş-gözden geçirme yapmazlar. Ancak geliştirme boyunca sürekli daha az resmi eş-gözden geçirmeler yaparak eş gözden geçirmenin amacına ulaşırlar. Bu, birçok Çevik organizasyonda yaygın bir uygulamadır. Ancak yine de bu faaliyetler yürütülürken bir disiplin olması gerekir. Bu TMMi® hedefi, bir projede eş-gözden geçirme yaklaşımının oluşturulmasına yönelik uygulamaları içerir. Bir gözden geçirme yaklaşımı, gözden geçirme faaliyetlerinin nasıl, nerede ve ne zaman yapılacağını ve bu faaliyetlerin resmi mi yoksa gayri resmi mi olduğunu belirler. Eş-gözden geçirme yaklaşımı oluşturma Çevik projeler için de geçerlidir, ancak uygulanan gözden geçirme teknikleri ile gözden geçirmelerin düzenlenme şekli çok farklıdır.

Çevik projelerde tipik olarak uygulanan eş-gözden geçirme örnekleri arasında şunlar yer alır:

- Bir döngü boyunca ekip ve iş paydaşları ile gereksinimler (örneğin kullanıcı hikayeleri) konusunda düzenli olarak iyileştirme / düzenleme oturumları yapmak;
- Geliştirilmekte olan çalışma ürünlerini (örneğin kodlar veya testler) açıkça tartışmak ve geri bildirimde bulunmak için diğer ekip üyeleriyle günlük toplantılar.
- En azından bir döngünün sonundaki döngü gözden geçirmesi esnasında ürünlerin erkenden ve sık sık müşterilere gösterilmesi;

Zayıf gereksinimler genellikle proje başarısızlığının önemli bir nedenidir. Gereksinim problemleri, kullanıcıların kendi gerçek ihtiyaçlarına ilişkin içgörü eksikliği, sisteme dair küresel bir vizyonun olmaması, gereksiz veya çelişkili gereksinimler ve diğer iletişim eksikliklerinden kaynaklanabilir. Çevik yazılım geliştirmede, iş kolu temsilcileri, yazılımcılar ve test uzmanlarının bakış açısından gereksinimlerin kayıt altına alınması için kullanıcı hikayeleri yazılır. Sıralı geliştirmede, bir özelliğin paylaşılan bu vizyonu, gereksinimler yazıldıktan sonra yapılan resmi gözden geçirmelerle gerçekleştirilir; Çevik yazılım geliştirmede ise bu paylaşılan vizyon, gereksinimler belirlenirken sık aralıklarla yapılan gayri resmi gözden geçirmelerle gerçekleştirilir. Bu gayri resmi gözden geçirme oturumlarına genellikle iş listesi iyileştirme veya iş listesi düzenleme oturumları adı verilir. İyileştirme toplantıları sırasında, iş kolu temsilcisi ve geliştirme ekibi (ve varsa paydaş), uygulama için gerekli ayrıntı düzeyini tespit etmek ve açık konuları netliğe kavuşturmak için gözden geçirme teknikleri kullanılır.

Ekip tarafından geliştirilmekte olan iş ürünlerinin neredeyse sürekli gözden geçirilmesi, tüm ekip yaklaşımının bir parçasıdır. Bu çalışmalar, hataları erken tespit etmek ve aynı zamanda iyileştirme fırsatlarını belirlemek amacıyla yapılır. XP veya pair-programming gibi Çevik yöntem ve teknikler, ekipler için geri bildirim döngüleri oluşturmak adına temel bir uygulama olarak eş-gözden geçirmeleri de içerir. Tabii ki yüksek düzeyde bir karmaşıklık veya riskin söz konusu olduğu durumlarda, ekip yarı resmi veya resmi bir gözden geçirme tekniği (örneğin teftiş) uygulamayı tercih edebilir. Bu gibi durumlarda, daha resmi bir gözden geçirme için çaba harcamanın ve daha disiplinli bir çalışma yöntemi uygulamanın açık bir gereksinimi vardır. Gözden geçirme yaklaşımı kriterlerinin bir parçası olarak, ne zaman daha resmi bir gözden geçirme tekniği kullanılacağı kararlaştırılabilir.

İş analistliği, gayri resmi gözden geçirmeler, izlenecek yollar, teftişler veya perspektif esaslı okuma gibi yöntemlerle gereksinimlerin doğrulanmasına odaklanır. Çevik yöntemler ise bunun yerine, değerli ürün artışlarını hızla hayata geçirebilmek amacıyla gereksinimleri sık aralıklarla yapılan erken geri bildirimler vasıtasıyla doğrulamaya gayret eder. Erken resmi doğrulama ihtiyacı, entegre ürün artışları şeklinde hızlı sonuçlar gösterilerek azaltılır. Artış, paydaşların gereksinimlerini tam olarak karşılamıyorsa, yeni gereksinimler şeklinde ürün iş listesine geri konur ve diğer tüm iş listesi kalemleriyle beraber önceliklendirilir. Bir döngü sırasında fiilen ne yapıldığının gösterilmesi, somut bir şey etrafında yapılacak doğrulamaya dayalı bir açıklamaya katkı sağlamanın çok etkili bir yoludur. Açıklananı öne çıkarma konusunda hiçbir şey, bir şeyin gerçekte nasıl çalıştığını gösterme fırsatının yerini alamaz. Gösterim, döngü gözden geçirmesi sırasında gerçekleştirilen bir aktivitedir. Özellikler paydaşlara gösterilir ve özellikler konusunda paydaşlarla tartışılır sonrasında ise ürün iş listesi veya sürüm planı üzerinde bu tartışmadan elde edilen yeni bilgileri yansıtacak şekilde gereken uyarlamalar yapılır.

3.5.2 ÖH2 Eş-Gözden Geçirmeler

Elbette her yaklaşım gibi bu yaklaşım da yalnızca kararlaştırılmış ve üzerinde mutabakata varılmış bir yaklaşım olarak var olmamalı, ona bağlı kalınmalıdır. Ekip, günlük rutinlerinin bir parçası olarak gayriresmi (ve muhtemelen) resmi gözden geçirmeler uygulayıp düzenli olarak paydaş demoları yaparak döngü sırasında iş listesi olgunlaştırma oturumlarına yönelik hatırı sayılır zaman harcamalıdır. Paydaş/müşteri gösterimlerinin en azından her döngü sonunda yapılması beklenir.

Giriş kriterleri sadece resmi gözden geçirmeler için geçerlidir bu nedenle Çevik projelerde muhtemelen pek uygulanabilir olmayacak veya tamamen uygulanamayacaktır. Ancak çıkış kriterlerinin kullanımı uygulanabilir olup aynı zamanda katma değere sahiptir. INVEST [Wake], Çevik projelerde kullanıcı hikayeleri gözden geçirilip güncellenirken sıklıkla sözü edilen bir dizi çıkış kriteri örneğidir. INVEST, iyi bir kullanıcı hikayesini oluşturan aşağıdaki kavramlardan oluşan bir kısaltmadır:

- Bağımsız (Independent)
- Müzakere edilebilir (Negotiable)
- Değerli (Valuable)
- Tahmin edilebilir (Estimable)
- Küçük (Small)
- Test edilebilir. (Testable)



Elbette kaliteyle ilgili başka çıkış kriterleri de vardır ve bunlar da faydalı bir şekilde kullanılabilir.

Test uzmanının, Çevik ekibin bir parçası olarak gözden geçirme oturumlarına katılması önemlidir. Bu durum özellikle de döngü boyunca test etme için bir dayanak olarak kullanılan çalışma ürünlerinin tartışılıp gözden geçirildiği oturumlar (örneğin, iş listesi olgunlaştırma oturumları) için geçerlidir. İş listesi olgunlaştırılırken sisteme dair alternatif bakış açılarının oluşmasını sağlamak açısından en az bir yazılımcıyla bir test uzmanının bu oturumlarda hazır bulunması tavsiye edilir. Test uzmanı benzersiz bakış açısıyla gözden kaçan ayrıntıları veya fonksiyonel olmayan gereksinimleri tespit ederek kullanıcı hikayesini iyileştirecektir. Bir test uzmanı özellikle, belirli bir kullanıcı hikayesine ilişkin kabul kriterlerinin belirlenmesini ve tanımlanmasını destekleyebilir. Bir test uzmanı, iş kolu temsilcilerine kullanıcı hikayesi hakkında açık uçlu ve farklı seçenekleri değerlendiren "farzedelim..." şeklinde sorular sorarak, kullanıcı hikayesini test etme yöntemleri önererek ve kabul kriterlerini teyit ederek sürece katkıda bulunur.

Özel uygulama 2.3 Eş-gözden geçirme verilerini analiz etme, özellikle resmi gözden geçirmelerle ilgili bir uygulamadır. Resmi gözden geçirmeler konusunda fazlasıyla çaba harcanır; harcanan bu çabanın hem verimli hem de etkin olması için eş-gözden geçirme verileri, gözden geçirme sürecini öğrenmek ve iyileştirmek amacıyla toplanır ve ekip ile paylaşılır. Belirtiliği üzere resmi gözden geçirmeler (örneğin, teftiş), Çevik projelerde pek yaygın değildir. Veri toplama resmi gözden geçirmelerin ana unsurlarından biri iken gayri resmi gözden geçirmelerde nadiren uygulanır. Sonuç olarak detaylı eş-gözden geçirme verilerinin toplanması, analizi ve paylaşımının Çevik bir bağlamda daha az kullanıldığından hatta hiç kullanılmadığından bahsedilebilir. Eş-gözden geçirme yaparken veri toplamaya karar verme bağlamında aşağıdaki sorular sorulabilir:

- Veriler toplanıyorsa bu verileri kim kullanacaktır?
- Bu veriler iş hedeflerimizle nasıl ilişkilidir?

Hiç kimse bu verileri gerektiği şekilde kullanmayacaksa değerli kaynakları bunları toplamaya ayırmamak gerekir. Sonuç itibarıyla Çevik projelerin çoğu için Özel Uygulama 2.3 Eş-gözden geçirme verilerini analiz etme, muhtemelen alakasız olarak değerlendirilecektir. Yine de bazı temel, değerli gözden geçirme verilerinin toplanacağını ve ilgili uygulamalar (örneğin, GP 2.8 Süreci Takip ve Kontrol Etme) bağlamında kullanılacağını unutmamak gerekir.

4) TMMi® 4. Seviye - Ölçülen

Daha önce de belirtildiği üzere TMMi® 3. Seviye ile TMMi® 4. ve 5. seviyeler arasında olağan bir bekleme süresi olduğunu unutmamak gerekir. Bir organizasyonu TMMi® 4. ve 5. seviyelere taşımanın değeri konusunda halen ciddi anlaşmazlıklar vardır. Özellikle Çevik organizasyonların TMMi® 4. ve 5. seviyelerdeki fark yaratabilecek ve katma değerli uygulamaları büyük bir özenle seçmeleri ve yalnızca bunları uygulamaları önerilir. TMMi® yeterince kapsamlı olsa da organizasyonların, başarılı olabilmek için odaklanma gerektiren temel test uygulamaları ve iyileştirmelerini belirlemeleri gerekir. İlginçtir ki, Scrum'ın kurucu ortağı Jeff Sutherland, Scrum'ı, daha üst olgunluk seviyelerindeki uygulamalarla beraber kullanmanın faydası üzerine bir makale yayınlamıştır [Sutherland ve diğerleri]. Dokümanın bundan sonrasında, 4. ve 5. seviye özelindeki bu daha yüksek TMMi® seviyeleri ve uygulamalarının Çevik yaklaşımla beraber daha az resmi biçimde kullanılmasıyla nasıl değer kazanabileceği hakkında bilgi verilecektir.

İlk olarak, şu ana kadar belirtilenlerle biraz ters düşecek olsa da TMMi® 2. ve 3. seviyeye ulaşmaya çalışırken aynı zamanda temel proje hedeflerine ulaşmak için Çevik tekniklerle TMMi® 4. ve 5. seviye uygulamalarından uygun olanı seçerek kullanmayı da düşünebilirsiniz. Beklemenize gerek yoktur ve beklememeniz de tavsiye edilir. Esnek davranın ve TMMi® 2. ve 3. seviye test uygulamalarına takılıp kalmayın. Modeli projenize yön veren amaçlara göre kullanın ve en fazla fayda sağlayacak süreç alanlarını, hedeflerini ve uygulamalarını kullanın. Bu doğrultuda, TMMi®'ı bir ölçüye kadar daha istikrarlı kullanın ve olgunluk seviyelerini çok katı bir şekilde kullanmayın.

Süreç Alanı 4.1 Test Ölçümü

Test Ölçümünün amacı, bir organizasyona, test sürecinin etkinliği ve verimliliğini, test personelinin verimliliğini, ortaya çıkan ürün kalitesini ve test iyileştirme sonuçlarını objektif olarak değerlendirme konusunda destek sağlamak için ölçümler yapmak, toplamak, analiz etmek ve uygulamaktır. Test organizasyonu bu doğrultuda, yönetim bilgisi ihtiyaçlarını desteklemek için kullanılan bir test ölçüm mekanizması geliştirecek ve bunu sürdürecektir.

Çevik yaklaşımlar, projelerde işi gerçekleştirmesi ve doğru bir şekilde ölçmesi gereken kişilerin bağlılığını kazanmayı destekler, ancak Çevik yaklaşımlar, organizasyon genelinde doğru ölçümlerin kullanılmasını sağlamak için geri adım atma ve bu ölçümlerin organizasyonun proje ihtiyaçlarından elde edilen ölçüm hedefleriyle uyumlu olmasını sağlamak konusunda çok fazla destek verme eğiliminde değildir. Bunlar, TMMi® test ölçümü süreci alanının Çevik organizasyonlara getirebileceği güçlü yönlerdir.

4.1.1 ÖH1 Test Ölçüm ve Analiz Faaliyetlerini Uyumlu Hale Getirme

Ölçüm programları çoğu zaman olabildiğince fazla veri toplamakla ilgili gibi görünür. Bir ölçüm programı son derece hedef odaklı olmalı ve yalnızca önemli olan şeyleri toplamalıdır. Ölçüm programı için net hedefler belirleyerek işe başlanmalıdır ve düzenli olarak bu hedeflerin üzerinde tekrar durulmalıdır (örneğin, "Ölçüm programı projelerimizdeki testlerde daha başarılı olmamızı sağlamalı). Ardından paydaşlarla bu hedefleri hangi test metriklerinin gerçekten destekleyeceğini tartışın. Bu beyin fırtınasının sonucuna göre sınırlı bir dizi ana metrik belirlenecektir.

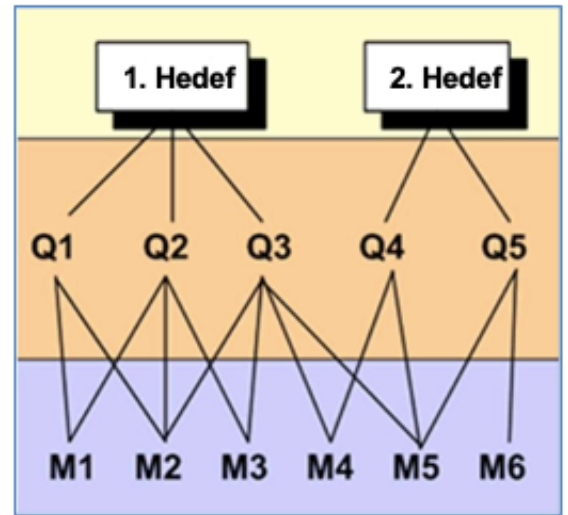
Amaç Soru Metriği (ASM) [Van Solingen ve Berghout] bu yaklaşımı destekleyecek olan oldukça kullanışlı bir yöntemdir. Proje ihtiyaçlarına bağlı olarak bağlamla ilgili özel ölçümlere ihtiyaç vardır. Şirketin standart metrikleri çoğunlukla gerçekteki test süreci iyileştirmesi için yetersizdir. Ölçümlerin amacı kararlara yön vermektir. Kayda değer ölçümler elde etmek için donanımlı küçük ekipler kullanın ve daha sonra bu ölçümleri kısa aralıklarla gözden geçirin ve geliştirin.

Katma iş değerinden yola çıkan hedef odaklı bir test ölçüm programı ile TMMi® Test Ölçümü süreç alanı Çevik basitlik ilkesiyle tamamen uyumludur. Organizasyon bünyesinde yalın bir test ölçümü ve veri toplama stratejisi olduğundan emin olun. Sadece birkaç odaklı ölçüm hedefi ve birkaç temel ölçümle işe başlayarak bir test ölçüm programının geliştirebileceğini ve proje ihtiyaçları gelişip değiştiğinde bunların genişletilebileceğini veya değiştirilebileceğini vurgulamakta fayda vardır.

Çeviklik'te odağın daha çok ekip temelli ve sistem odaklı düşünmeye dayalı olacağını unutmamak gerekir. Bu durum, bazı metriklerin, yalnızca test çalışmasının kendi özel koşullarıyla sınırlı kalmasından ziyade ekibe ve sistemin geneline yayılmasına sebep olabilir. Bu da ölçüm analiz safhasında daha fazla zorluk yaşanmasına yol açabilir.

Elbette ki oluşturulan metrikler ile verilerin kimler için toplanacağı ölçüm hedeflerine bağlı olacaktır, ancak çevik ekipler için tipik bazı test metriği örnekleri şöyledir:

- Hata döngü süresi; Çevik ekipler hataları olabildiğince çabuk gidermeye çalışmalıdır. Hatta, işbirlikçi Çevik yaklaşımın temel



amaçlarından biri, yazılımın daha erken uygulamaya alınması için hataların daha hızlı düzeltilmesidir.

- Hata taşması (Bir sonraki sürüme ertelenen hata sayısı); Çevik ekipler her döngüde çalışan bir yazılım üretmeyi hedefler. Hata taşması, basitçe her sprint veya döngünün sonunda kalan hataların sayılması suretiyle belirli bir döngü veya sprint sırasında düzeltilmeyen hataları ölçer. Buna bazı Çevik kuruluşlarda teknik borç adı verilir.
- Bir döngü sırasında bulunan hataların sayısı
- Üretime geçiş/müşteriye gönderim sonrasında bulunan hataların sayısı
- Ekip dışındaki kişiler tarafından bulunan hataların sayısı
- Müşteri destek taleplerinin sayısı
- Otomatik test kapsamının yüzdesi
- Kod kapsamının yüzdesi.

4.1.2 ÖH2 Test Ölçüm Sonucunu İletme

İlgili ölçümler proje ihtiyaçlarına göre belirlendikten sonra bu özel hedef içinde yer alan özel uygulamaların tümü, geleneksel bir organizasyondakilerde olduğu gibi uygulanabilir. Aradaki tek fark, bunun uygulamaları mümkün olduğunca yalın tutularak farklı bir zihniyetle gerçekleştirilecek olmasıdır.

Gerekli test ölçüm verileri toplanarak eksik olup olmadığı ve bütünlükleri açısından kontrol edilir, veriler daha sonra planlanan şekilde analiz edilir ve sonuçlar ilgili tüm paydaşlara iletilir.

Süreç Alanı 4.2 Ürün Kalitesi Değerlendirme

Ürün Kalitesi Değerlendirmenin amacı, ürünlerin kalitesi hakkında nicel bir anlayış geliştirmek ve böylece belirli projelerin ürün kalitesi hedeflerine ulaşmasını desteklemektir.

4.2.1 ÖH1 Ürün Kalitesine Yönelik Ölçülebilir Proje Hedefleri ve Öncelikleri Belirleme

Çevik bağlamda ürün kalitesine yönelik ölçülebilir öncelikli hedefler belirleme ve kararlaştırma, temelde geleneksel bir ortamdakinden pek de farklı değildir. Genel amaç, müşteriler ile son kullanıcıların kaliteli ürünler konusundaki ihtiyaç ve isteklerinin karşılanmasına katkı sağlamaktır. Çevik bağlamda, bunu yapmanın birçok yolu vardır. Bazıları ürün kalitesine dair ölçülebilir hedefler belirlemek için Tamamlandı Tanımını kullanırken diğerleri özellik, epik ve kullanıcı hikayelerini ve bunları ölçülebilir hale getirmek için de kabul kriterlerini kullanırlar. Nasıl çalışılacağı kararı başka faktörlerin yanı sıra ürün kalitesi hedefinin niteliğine bağlıdır. (Kalite özellikleriyle ilgili genel bilgi için [ISO25010] bakınız.)

Tamamlandı Tanımı (DoD), yalnızca fonksiyonlari açısından değil ürün kalitesi açısından da gerçekten tamamlanmış olan kullanıcı hikayelerinin teslim edilmesini sağlayan gerekli ekip faaliyetlerinin ve/veya iş ürünlerinin kapsamlı bir listesidir. DoD'de ürün kalitesi hedeflerinin belirtilmesinin avantajı, bunları tüm ekip için görünür kılmayı ve dolayısıyla geliştirmeye devam ederken ekibin hedeflere yönelik olarak çalışabilmesidir. Kalite hedeflerinin DoD'de belirtilebilmeleri için uygulama açısından küresel düzeyde geçerli olmaları gerektiği açıktır. Kalite hedefleri küresel düzeyde geçerli olmazsa muhtemelen en iyi alternatif çalışma şekli bunları bir özellik veya epik olarak tanımlamaktır.

Bazen ilk başta, ilgili ürün kalitesi özelliklerinin "yalnızca" geniş kapsamlı bir listesi sürüm seviyesinde belirlenir. Bunlar olgunlaştırma oturumu sırasında kademeli olarak bir özellik, epik veya kullanıcı hikayesi olarak detaylandırılacaktır. Ardından, bir döngü başlangıcında ekip, belirlenen ürün kalitesi hedeflerinin söz konusu döngüye dair hedefin bir parçası olup olmayacağını değerlendirecektir.

4.2.2 ÖH2 Projenin Ürün Kalitesi Hedeflerine Ulaşması yönünde Kaydedilen İlerlemenin Ölçülmesi ve Yönetilmesi

Nicel ürün kalitesi hedefleri, belirlendikten sonra takip edilecek, kontrol edilecek ve muhtemelen gerektiğinde yeniden düzenlenecektir. Ürün kalitesi, DoD'in bir parçası olarak ve/veya ölçülebilir kabul kriterleri bulunan belirli özellikler, epikler veya kullanıcı hikayeleri olarak tanımlanacağından TMMi® 2. Seviyesindeki süreç alanı 2.3 Test Gözetimi ve Kontrolü'nde açıklandığı ve belirtildiği üzere ürün kalitesi hedeflerine ulaşmaya yönelik ilerlemenin takip edilmesi ve yönetilmesi için aynı yöntem ve teknikler kullanılabilir. İlerleme günlük olarak ölçülecek ve ilerleme, kalite ve riske ilişkin gelişmelerin bildirildiği günlük ayakta toplantı bildirimleri aracılığıyla tartışılacaktır. Ayrıca Ürün Kalite Değerlendirmesi, genellikle bir ölçüm altyapısı sağlayan Test Ölçümü süreç alanı tarafından da desteklenmektedir.

Süreç Alanı 4.3 İleri Gözden Geçirmeler

TMMi® 3. seviye süreç alanı Eş-Gözden Geçirmelerin uygulamalarına dayanan İleri Gözden Geçirmelerin amacı, ürün kalitesini yaşam döngüsünün başlarında ölçmek ve eş-gözden geçirmeler (statik test) ile dinamik testi birbirine uyumlu hale getirerek test stratejisi ve test yaklaşımını iyileştirmektir.

4.3.1 ÖH1 Eş-Gözden Geçirme Yaklaşımı ile Dinamik Test Yaklaşımını Koordine Etme

Geleneksel projelerde statik testler (örneğin, gözden geçirmeler) çoğu zaman bağımsız bir test ekibi tarafından yapılan dinamik testlerle bağlantılı olmasa da , bu durumun Çevik bağlamda tamamen farklı olması beklenir. Çevik ortamda, ürün kalitesinin sağlanmasına yönelik yaklaşım, testle ilgili statik veya dinamik tüm faaliyetlerin tek bir tutarlı ve koordineli yaklaşım olarak düzenlendiği bir yaklaşım olmalıdır. Elbette ki statik ve dinamik testleri entegre etmek ve en uygun karışımı bulmak Çevik ekibin sorumluluğundadır. Çevik ekibin, bu tutarlı ve etkin bütünleşik yaklaşımın oluşturulmasına öncülük etmesi beklenir. Dolayısıyla, statik test (eş-gözden geçirme) yaklaşımı, dinamik test yaklaşımı ile uyumlu hale getirilecek ve koordine edilecektir.

4.3.2 ÖH2 Ürün Kalitesini Yaşam Döngüsünün Başlarında Eş-Gözden Geçirmeler Vasıtasıyla Ölçme

TMMi® 4. seviyede, organizasyon yazılım ürünleri ve ilgili proje ürünleri için nicel hedefler belirler. Eş-gözden geçirmeler, bu hedeflere ulaşılmasında önemli bir rol oynar. TMMi® 3. seviyede eş-gözden geçirmeler daha çok

hataların tespiti için yapılırken bu seviyede, ürün kalitesini yaşam döngüsünün başlarında kontrol edebilmek için ürün/belge kalitesinin ölçülmesine yoğunlaşılır. Gözden geçirme uygulamaları, örnekleme, çıkış kriterleri uygulama ve kurallar belirleme gibi uygulamaları içerecek şekilde geliştirilir. Bu gerçekten de Tom Gilb'in Çevik teftiş olarak adlandırdığı durumdur ve amaç, odağı temizlemeden alıp örneklemeye, ölçüme, motivasyona ve hataların önlenmesine yönlendirmektir. Bu açıkça yüksek olgunluktaki Çevik kuruluşlara uyacak bir uygulamadır. Bu uygulamalara başka bir örnek de kullanıcı hikayeleri için Hazır Tanımı (DoR) olarak INVEST kurallarını [Wake] kullanmak ve kullanıcı hikayelerinin kalitesini, bunları bu kurallara göre gözden geçirerek ölçmektir. Hazır Tanımının bulunması, hikayelerin hemen eyleme geçirilebilir olması gerektiğini ifade eder.

4.3.3 ÖH3 Test Yaklaşımını Gözden Geçirme Sonuçlarına göre Yaşam Döngüsünün Başlarında Uyarılma

Eş-gözden geçirme sonuçları ile dinamik testler koordine edildiğinde, erken gözden geçirme sonuçları ve verileri ürün riskleri ile test yaklaşımını etkileyebilir. Bu süreç alanının 1. özel hedefinde belirtildiği üzere Çeviklik'te kalite bir ekip çabası olup doğrulama ve sağlama esaslı faaliyetlerin sonuçları ekip toplantılarında tartışılacaktır. Dolayısıyla, erken test sonuçlarını, sonraki test faaliyetlerini etkilemek üzere kullanma ilkesi, Çevik ekipler ve projelerdeki çalışma biçimiyle neredeyse mükemmel bir uyum içindedir.



5) TMMi® 5. Seviye Optimize Edilen

TMMi® 5. seviyedeki hedefler ile uygulamalar, yüksek olgunluktaki kalite yönetimi ve süreç iyileştirme yaklaşımlarından elde edilmiştir. Bu yüksek olgunluk düzeyinde hedefler ile uygulamalar, uygulanmakta olan yazılım yaşam döngüsünden neredeyse bağımsızdır. Tabii ki yalın Çevik tabanlı bir organizasyonun olaylara yaklaşımı, geleneksel hiyerarşi esaslı bir kuruluşunkinden farklı olacaktır, ancak kalite kontrol, süreç optimizasyonu ve hata önleme ile ilgili temel uygulamalar büyük ölçüde değişmeden kalacaktır. Bunun gerekçesi, bu uygulamaların genellikle operasyonel ürün geliştirme seviyesinden daha yüksek bir seviyede gerçekleştirilecek olan kalite yönetimi uygulamaları olarak kabul edilmesidir. Dolayısıyla bu uygulamalar, bir proje veya ürün teslim akışı içindeki yazılım geliştirme ve/veya test etme ile ilgili ana faaliyetlerle doğrudan ilgili uygulamalar değildir.

Süreç Alanı 5.1 Hata Önleme

Hata Önlemenin amacı, geliştirme yaşam döngüsü boyunca hataların ortak nedenlerini belirleyip analiz etmek ve gelecekte benzer hataların meydana gelmesini önlemek için yapılması gerekenleri kararlaştırmaktır.

Çevik organizasyonlar, genellikle bu süreç alanının özel hedeflerinin ve özel uygulamalarının üzerine inşa edilebileceği önemli bir temel olan hata önlemeye odaklanır. Etkin çevik organizasyonlar sadece hata tespitinden ziyade hata önleme üzerine yoğunlaşırlar. Hata önleme kültürü, "çalışması devam eden" hataları, kullanıcı hikayelerinin geliştirildiği döngünün ötesine geçen hatalardan ayırmayı içerir.

Çevik geçmişe dönük toplantı, ekiplerin yaptıkları işler üzerinden geçmesi, öğrenmesi ve yaptıkları işte sürekli olarak daha iyi olmalarına yönelik bir uygulamadır. Geçmişe dönük toplantılar çoğunlukla mevcut çalışma şeklini araştırmak için kullanılsa da kalite sorunlarını araştırmak veya üretilen yazılımın kalitesini artıracak eylemler üzerinde anlaşmak için de kullanılabilirler.

Hata Önleme süreç alanı, ekipler, ürünler ve değer akışları genelinde yaygın hata nedenlerini belirlemeye ve analiz etmeye ve gelecekte bu tür hataların ortak nedenlerini ortadan kaldırmak için belirli aksiyonları almaya yönelik uygulamaları ele almaktadır. Geliştirme veya test sırasında ya da sahada tespit edilmiş olmasına bakılmaksızın tüm hatalar, süreç alanı kapsamındadır. Bu seviyedeki hata önleme ölçüm verilerine ihtiyaç duyduğundan hata önleme, TMMi® 4. seviye ölçüm uygulamalarına ve geliştirme, test ve ürün kalitesiyle ilgili mevcut ölçüm verilerine dayanır.

Çevik organizasyonlarda temel olarak iki ayrı türde hata önleme faaliyeti vardır: takım seviyesi ve takımlar arası seviye. Ekiplerle ilgili hata önleme faaliyetleri ekibin sorumluluğundadır ve bunlar öncesinde TMMi® 3. seviyede ele alınmıştır. Ekipler arası hata önleme faaliyetleri ise bir Test Süreç Grubu veya test birliği tarafından gerçekleştirilmektedir ve bu süreç alanının özellikle üzerinde durduğu alanlar da bunlardır.

5.1.1 ÖH1 Hataların Ortak Nedenlerini Belirleme

Test süreci iyileştirmeleri de dahil olmak üzere Çevik iyileştirmeler genellikle sık geri bildirim döngülerine dayalı olarak gerçekleşir. Kapsam çoğu zaman önceki döngü ile sınırlı olduğu için temel olarak belirli yerel sorunları çözmeye odaklanan küçük ama sık iyileştirmeler yapılır. Bu iyileştirmelerin odağı genellikle ekipler arası öğrenme ve iyileştirmelerin kurumsallaştırılması üzerine değildir. Sorunları yalnızca ekip seviyesinde çözmek de kolayca alt düzey optimizasyona ve büyük resmi görmekten uzaklaşılmasına yol açabilir. Bu TMMi® süreç alanının, mevcut Çevik uygulamaları geliştirerek katma değer sağladığı alan da budur; hata önleme artık organizasyon düzeyinde de (ekipler, ürünler ve değer akışları genelinde) uygulanmakta ve genellikle Test Süreç Grubu veya test birliği tarafından yönetilmektedir. (Bkz. bölüm 3.1.1)

Analiz edilecek hataların seçimi, risk ve proje değeri de dahil olmak üzere çeşitli faktörlere dayanmalıdır. Hataların önlenmesinin en fazla katma değer sağlayacağı (genellikle azalan maliyet veya riskler açısından) ve/veya hataların en kritik olduğu alanlara odaklanılmalıdır. Hata önleme süreç alanı yaygın olarak hatalar için kullanılsa da hız sorunları gibi problemler için de kullanılabilir.

Bir Test Süreç Grubu, hata önleme faaliyetlerini koordine edebilir, ancak iyileştirme eylemleri genellikle diğer birimleri de etkileyeceği için bu faaliyetlerin farklı birimlerden bir ekip (örneğin, iş analistiği, sistem mühendisliği ve/veya yazılım geliştirme temsilcileri) tarafından yürütülmesi gerekir. Elbette bu yaklaşım, Çevik teslim yöntemleri ve ekiplerinde halihazırda yaygındır. Ancak, her bir iyileştirme işlemine göre gerekli birimlerin bu ekipte yer aldığından emin olunmalıdır.

Neden-sonuç diyagramları, balık kılçığı diyagramları veya 5 neden gibi hataların kök nedenlerini değerlendirmeye yönelik analitik yöntemler genellikle hata önleme için kullanılmakta olup halihazırda yaygın olan Çevik uygulamalardır.

Bu özel hedef için TMMi® modelinde açıklanan üç özel uygulamanın (1.1 Hata seçim parametrelerini tanımlama, 1.2 Analiz için hata seçme ve 1.3 Seçilen hataların kök ve ortak nedenlerini analiz etme) tamamı Çevik bir organizasyon için de geçerlidir.



5.1.2 ÖH2 Hataların Ortak Nedenlerini Sistematik Olarak Ortadan Kaldırabilmek için Yapılacak İşlemleri Önceliklendirme ve Belirleme

Bu özel hedefe dair özel uygulamaların, Çevik'te veya sıralı bir yaşam döngüsünde gerçekleştirilmesi arasında pek fark yoktur. Elbette ki önerilen fiili çözüm ile ardından gelen iyileştirme önerisi farklı olacaktır, çünkü bu önerilerin Çevik çalışma şekline uygun olması gerekir (örneğin, onaylı, resmileştirilmiş eylem önerileri yerine ilave iş listesi maddeleri oluşturma). Problemin tekrarlama olasılığını en aza indirmek için gerekli önlemler alınır. Bu yaklaşım çoğunlukla süreç iyileştirmesini, ekip üyeleri ve diğer personele ilave mentorluk ve/veya eğitim sağlanması ile birleştirmek anlamına gelir. Kişiler ile süreç sorunlarını yapay bir şekilde ayırmaya çalışmak yerine bunların her ikisini de entegre bir biçimde ele almak gerekir.

Ayrıca Çevik organizasyonlar kendi kendini yöneten ekipler üzerine kuruludur. Buna göre ekipler, hangi iyileştirmeleri uygulayıp uygulamayacaklarını seçebilirler. Dolayısıyla, eylemlerin önceliklendirilmesi ve belirlenmesi sırasında, söz konusu iyileştirmelerin gerçekten de ekiplere katma değer sağlayıp sağlamadığının tartışılabilmesi için ilgili ekiplerden yeterli sayıda temsilcinin bu süreçte dahil edilmesi gerekir.

Ayrıca, iyileştirmelerin hayata geçirilme şekli de farklı olacaktır; hiyerarşik yapıya pek uyulmayacak ve daha ziyade kendi kendini yöneten ekiplere gönüllülük esasına bağlı olarak öneride bulunma şeklinde olacaktır. Test iyileştirmelerinin hayata geçirilmesi konusunda daha fazla bilgiye aşağıda açıklanacak olan Test Süreci Optimizasyonu süreci alanında erişilebilir.

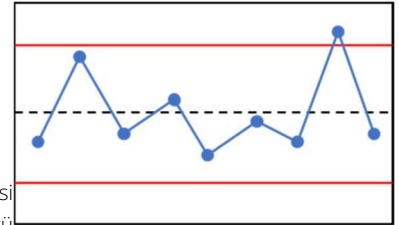
Süreç Alanı 5.2 Kalite Kontrol

Kalite Kontrolün amacı, test sürecini istatistiki olarak yönetmek ve kontrol etmektir. Bu düzeyde test süreci performansı tamamıyla tahmin edilebilir olup kabul edilebilir limitler içinde dengelenir. Proje düzeyindeki test, ürün kalitesini tahmin edebilmek ve testi daha verimli hale getirmek için temsili örneklerle dayalı istatistiksel yöntemler kullanılarak yapılır.

5.2.1 ÖH1 İstatistiksel Olarak Kontrol Edilen Bir Test Süreci Oluşturma

Kalite kontrol ilke ve esasları uygulanmakta olan yaşam döngüsü yöntemine bakılmaksızın aynıdır. Süreç kalite kontrolü, standart test sürecinin gerçekleştirilmesi için hedeflerin belirlenmesini içerir. Ekiplerden test süreci performansı üzerine alınan ölçümlere dayanarak analiz yapılır ve test süreci performansını kabul edilebilir limitler içerisinde tutmak için ayarlamalar yapılır. Test süreci performansı kabul edilebilir limitler içinde dengelendiğinde oluşturulmuş test süreci, ilgili ölçümler ve ölçümler için kabul edilebilir limitler, temel çizgi olarak belirlenir ve test süreci performansını istatistiksel olarak kontrol etmek için kullanılır.

Organizasyonun standart test sürecinin test süreci yeterliliği, yani, bir ekibin elde etmeyi bekleyebileceği test süreci performansı artık tam olarak anlaşılmış ve bilinmektedir. Sonuç olarak ekip, bu beklentilerden sapmaların kabul edilebilir limitler içinde performans göstermelerini sağlamak için hızlı ve tutarlı bir biçimde çalışmaya başlayabilir.



İstatistiksel kontrol için iş açısından kritik olan ve geçmişte problemlerin yaşandığı alt süreçlerin seçilmesi önemlidir. Deneyimler, bu kriterleri en iyi karşılayan alt süreçlerin çoğunlukla tek bir ekibe özgü olmadığını, daha ziyade birden çok çapraz ekip, proje ve ürün teslim akışında görüldüğünü göstermektedir. Çevik yöntem ve teknikler kullanılırken iş kısa döngüleri bölünür; her döngüde tamamlanan işlerde birçok geliştirme ve test faaliyeti gerçekleştirilir. Bu nedenle Çevik yazılım geliştirme, istatistiksel kontrol için en değerli olan alt süreçlerin izlenmesini, geleneksel gelişimden daha etkili bir şekilde destekler, çünkü döngüler çok daha kısadır ve bu durum, test süreci performansı hakkında neredeyse anında geri bildirim alınmasını sağlar.

Hızlı geri bildirim ekipler için çok değerli olup aynı zamanda önemli bir Çevik prensibidir.

5.2.2 ÖH2 İstatistiksel Yöntemler Kullanarak Test Yapma

Ürün kalite kontrolü, testlerin temsili bir örneklemyle sonuçlanan istatistiksel olarak geçerli çıkarımlar yapmak için ürünün operasyonel profilleri [Musa] ile amaçlanan ortamdaki kullanım modellerini esas alır. Özellikle sistem düzeyinde faydalı olan bu yaklaşımda, söz konusu temsili örneklemeye dayalı olarak ürün kalitesinin tahmin edilebilmesi için istatistiksel test yöntemleri kullanılır. Başka bir deyişle, kullanım profili veya operasyonel profil tarafından temsil edilen tüm olası kullanımların bir alt kümesi test edilirken test sonuçları, ürünün genel performansı hakkında varılacak sonuçların temelini oluşturabilir.

Bu temel olarak Çevik ile iyi uyum sağlayan bir yaklaşımdır, çünkü doğası gereği yalındır ve ürünle ilgili operasyonel kullanım ve gözden geçirme oturumlarından gelen geri bildirimlere göre şekillenir. Ancak, bu yaklaşımı uygulayabilmek için Çevik organizasyonun doğru olgunluk seviyesinde olması gerekir. Bu yaklaşım ayrıca, toplanan ve mevcut olan ölçüm ile verilerden önemli ölçüde faydalanmaktadır ve bunlara bağlıdır.

Testlerde istatistiksel yöntemler kullanan bir organizasyon güven düzeylerini ve güvenilirliği nicel olarak ölçebilir (bu terimlerle ilgili daha fazla bilgi için bkz. TMMi® çerçevesi). İstatistiksel testler uygulanırken hem güven hem de güvenilirlik düzeyi genellikle Tamamlandı Tanımı olarak kullanılır.

Süreç Alanı 5.3 Test Süreci Optimizasyonu

Test Süreci Optimizasyonunun amacı, kuruluştaki kullanılan mevcut test süreçlerini sürekli olarak iyileştirmek ve uygun olabilecek yeni test teknolojilerini (örneğin, test araçları veya test yöntemleri) belirleyerek bunları düzenli bir şekilde organizasyona uyarlamaktır. Test Süreci Optimizasyonu ayrıca test varlıklarının organizasyon genelinde yeniden kullanımını da destekler. İyileştirmeler, organizasyonun iş hedeflerine bağlı olarak oluşturulan ürün kalitesi ve test süreci performans hedeflerini destekler.

5.3.1 ÖH1 Test Süreci İyileştirmelerini Seçme

Test iyileştirme önerileri toplanıp analiz edilir. Test iyileştirme önerileri organizasyon içi kaynaklar ile genellikle ekipler arası, projeler arası ve ürünler arası akıştan elde edilir. Organizasyon içi kaynaklar, hata önleme faaliyetleri, ekip gözden geçirmeleri, geçmişe dönük toplantılar, topluluklar ve birliklerden gelen sorun ve fikirleri içerir. Test iyileştirme önerileri, maliyet ve fayda analizleri yapıldıktan sonra organizasyon genelindeki ekiplerde uygulamaya geçirilmeden önce yeni ve henüz fark edilmemiş önemli değişikliklerin değerlendirilebilmesi için pilot proje olarak uygulanır. Son olarak, organizasyon genelinde uygulanacak olan test iyileştirmeleri seçilir.

5.3.2 ÖH2 Yeni Test Teknolojilerinin, Test Süreci üzerindeki Etkilerinin Belirlenebilmesi için Değerlendirilmesi

Test süreci optimizasyonu kapsamında, merkezi olarak yapılanmış olan test süreç grubu (veya test birliği) piyasadaki araçlar, yöntemler, teknikler veya teknik yenilikler gibi Çevik ekiplerin test olanaklarını geliştirebilecek yeni teknolojileri proaktif bir biçimde araştırır. Organizasyon ekip temsilcilerine danışarak testle ilgili teknolojik yenilikler konusunda bir farkındalık sağlayarak ve bunları sistematik olarak değerlendirip denemeler yaparak ürünlerinin kalitesini ve test faaliyetlerinin verimliliğini artırmak amacıyla uygun test teknolojilerini seçer. Standart uygulamalara dahil edilmeden önce yeni ve kanıtlanmamış test teknolojilerini değerlendirilebilmek için ekip bünyesinde pilot uygulamalar yapılır.

5.3.3 ÖH3 Test İyileştirmelerini Hayata Geçirme

Test iyileştirmeleri ile uygun bulunan yeni test teknolojileri, test çalışmalarının iyileştirilmesi için Çevik ekipler genelinde hayata geçirilir. Bunların faydaları ölçülür ve yeni uygulanmaya başlanan yeniliklerle ilgili bilgiler organizasyon geneline yayılır. Elbette geleneksel ve çevik organizasyondaki hayata geçirme süreçleri arasında fark vardır. Geleneksel bir organizasyonda hayata geçirme çoğunlukla, ilgili süreçlerde yer alan herkes tarafından uygulanacak yeni uygulamalar oluşmasına sebep olur. Çevik bir organizasyonda ise ekip daha fazla özerkliğe sahiptir ve kendi kendini yönetmektedir. Bu aynı zamanda hayata geçirme sürecinin de farklı olacağını gösterir. Test iyileştirmeleri ile yeni test teknolojileri, organizasyonel test süreci varlıklarına uygun biçimde dahil edilir, ayrıca eğitim ve destek programları planlanır ve sağlanır.

Ancak nihayetinde, söz konusu test iyileştirmesinin çalışma şekillerinin bir parçası olup olmayacağına Çevik ekibin kendisi karar vermelidir.

5.3.4 ÖH4 Kaliteli Test Varlıklarının Yeniden Kullanımını Sağlama

TMMi® 3. seviyede, test yazılımlarının ekipler, projeler ve ürün akışları genelinde bir ölçüde yeniden kullanımı söz konusu olabilir, ancak test varlıklarının yeniden kullanımı TMMi® 5. seviyede önemli bir hedef haline gelmektedir. Ekibin tekerleği yeniden icat etmesine gerek yoktur; bunun yerine halihazırda mevcut olan uzmanlık ve varlıklardan faydalanarak zaman ve çabadan tasarruf edebilirler. Organizasyon genelinde muhtemelen tekrar kullanılacak kaliteli test varlıkları, örneğin geçmişe dönük toplantılar, öğrenilen tecrübeler oturumları, test birliği içindeki tartışmalar veya test değerlendirmelerinde tespit edilir. Daha sonra yeniden kullanılacak test varlıkları seçilir ve ilerideki ürün teslim ve projelerinde yeniden kullanım için değiştirilebilir bir formatta merkezi test havuzuna eklenir.

6) TMMi® Özel Hedef ve Uygulamalarının Uygulanabilirliğine Dair Genel Değerlendirme

6.1 TMMi® Değerlendirmeleri

Baş denetçi ile denetçilere değerlendirmelerinde yardımcı olmak üzere bu bölümde, süreç alanlarının, özel hedeflerin ve özel uygulamaların Çevik bir bağlamda uygulanabilirliğine ilişkin genel bilgiler verilmiştir. Önceki bölümlerde, kısmen geçerli veya geçersiz sınırlı sayıda özel uygulamaya yer verilmişti. Bu özel uygulamalar, buradan itibaren sıralanmaktadır. TMMi® değerlendirmesi sırasında baş denetçi veya denetçinin bu maddeleri, söz konusu Çevik bağlamdaki anlamları açısından açık bir biçimde değerlendirmesi gerekecektir. Bir maddenin süreç için kısmen geçerli olduğu veya tamamen geçersiz olduğuna karar verilmesi halinde, bu kararın gerekçeleriyle beraber hem kuruluşun süreçlerinde hem de değerlendirme raporunda net bir şekilde belgelenmesi beklenir.

6.2 TMMi® 2. seviye Yönetilen

Süreç Alanı 2.2 Test Planlama

ÖH2 Test Yaklaşımı Oluşturma

- ÖU2.3 Giriş kriterlerinin belirlenmesi
- ÖU2.5 Askıya alma ve devam kriterlerinin belirlenmesi

ÖH3 Test Tahminleri Oluşturma

- ÖU3.2 Test yaşam döngüsü oluşturma

ÖH5 Test Planına Bağlılığı Sağlama

- ÖU5.2 İş ve kaynak seviyelerini bağdaştırma

Geçerli

Kısmen geçerli/geçersiz

Kısmen geçerli/geçersiz

Geçerli

Kısmen geçerli/geçersiz

Geçerli

Kısmen geçerli/geçersiz

Süreç Alanı 2.3 Test Gözetimi ve Kontrolü

ÖH2 Plan ve Beklentilere göre Ürün Kalitesinin Gözetimi

- ÖU2.3 Giriş kriterlerinin gözetimi
- ÖU2.5 Askıya alma ve devam kriterlerinin gözetimi

Geçerli

Kısmen geçerli/geçersiz

Kısmen geçerli/geçersiz

Süreç Alanı 2.4 Test Tasarımı ve Yürütme

ÖH2 Test Uyarlama

- ÖU2.3 Alım testi prosedürünü belirleme
- ÖU2.4 Test koşum çizelgesi oluşturma

ÖH3 Test Koşumu Yapma

- ÖU3.1 Alım Testi Gerçekleştirme

Geçerli

Kısmen geçerli/geçersiz

Kısmen geçerli/geçersiz

Geçerli

Kısmen geçerli/geçersiz

6.3 TMMi® 3. seviye Tanımlı

Süreç Alanı 3.5 Eş-Gözden Geçirmeler

ÖH2 Eş-Gözden Geçirmeleri Gerçekleştirme

- ÖU2.3 Eş-gözden geçirme verilerini analiz etme

Geçerli

Kısmen geçerli/geçersiz

Referanslar

- [Black, Van Veenendaal] Rex Black, Erik van Veenendaal and Dorothy Graham (2012), Foundations of Software Testing ISTQB Certification (3. baskı), Cengage
- [Cohn] M. Cohn (2009), Succeeding with Agile: Software Development using Scrum, Addison-Wesley
- [Galen] R. Galen (2015), Three Pillars of Agile Quality and Testing: Achieving Balanced Results in your Journey Towards Agile Quality, RGCG, LLC
- [Gilb] T. Gilb (2004), Agile Specification Quality Control, International Council on Systems Engineering (INCOSE)
- [ISO25010] ISO/IEC 25010 (2011), Systems and Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) System and Software Quality Models, International Organization for Standardization
- [Sutherland et al] C.R. Jakobsen, K. Johnson and J. Sutherland (2007), Scrum and CMMI level 5: The Magic Portion for Code Warriors in: IEEE Agile 2007 Conference, Washington, D.C.
- [McMahon] Paul. E. McMahon (2011), Integrating CMMI with Agile Development – Case Studies and Proven Techniques for Faster Performance Improvement, Addison Wesley
- [Musa] J. Musa (1998), Software Reliability Engineering Testing, McGraw-Hill Education
- [Van der Aalst and Davis] L. van der Aalst and Cecile Davis (2013), TMap NEXT in scrum – Effective testing in agile projects, Sogeti Nederland B.V.
- [Van Solingen and Berghout] R. van Solingen and E. Berghout (1999), The Goal/Question/Metric method, McGrawHill
- [van Veenendaal] E. van Veenendaal (2014), PRISMA: Product Risk Assessment for Agile Projects, in: Testing Experience, Sayı 04/12, Aralık 2012
- [Wake] B. Wake (2003), INVEST in Good Stories, and SMART Tasks, xp123.com/articles/invest-in-good-stories-and- smart-tasks