



YAZILIM GELİŞTİRMEDE ÇEVİK YÖNTEMLER

EXTREME PROGRAMMING (XP)

Extreme Programming (XP)

- Extreme Programming (XP), yazılım geliştirmeye yönelik bilinçli ve disiplinli bir yaklaşımdır.
- XP başarılıdır çünkü müşteri memnuniyetini vurgular.
- Metodoloji, müşterinizin ihtiyaç duyduğu yazılımı ihtiyaç duyduğu anda sunmak üzere tasarlanmıştır.
- Bu metodoloji ayrıca ekip çalışmasını vurgular.
- Yöneticiler, müşteriler ve geliştiriciler, kaliteli yazılım sunmaya adanmış bir ekibin parçasıdır.
- XP, grup yazılımı tarzı geliştirmeyi etkinleştirmenin basit ancak etkili bir yolunu uygular.
- XP programcıları müşterileri ve diğer programcılarla iletişim kurar. Tasarımlarını basit ve temiz tutarlar. Yazılımlarını ilk günden itibaren test ederek geri bildirim alırlar.

- XP, küçük ve orta ölçekli ekiplerin değişen ve gelişen gereksinimlere uyum sağlarken üstün kalitede yazılımlar oluşturmasını sağlamak için yazılım geliştirmenin teknik yönünü vurgulayan bir dizi ilke, uygulama ve değer üzerine kuruludur.

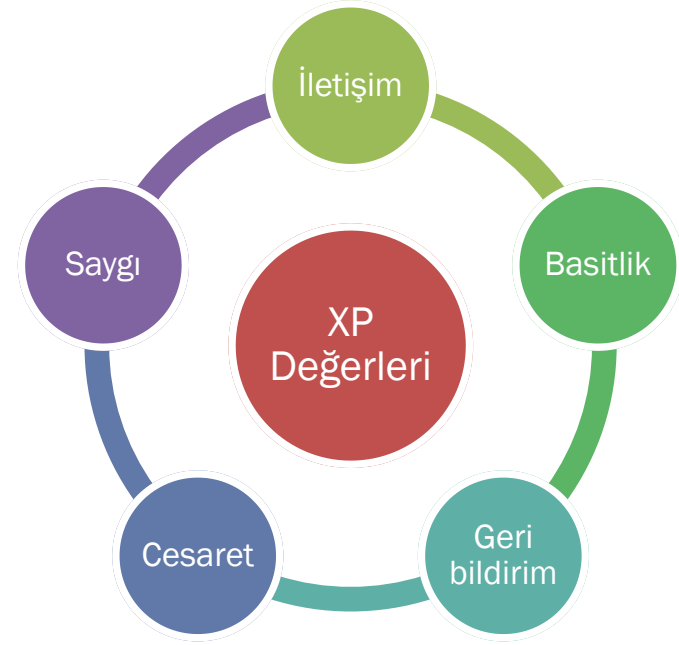


XP Değerleri

- XP programlama değerlere dayanır.
- Kurallar, değerleri maksimize etmenin doğal uzantısı ve sonucudur.
- XP aslında bir dizi kural değil, kişisel ve kurumsal değerlerinizle uyum içinde çalışmanın bir yoludur.
- XP değerleri:
 - Basitlik
 - İletişim
 - Geri bildirim
 - Saygı
 - Cesaret

XP Programlama deęerleri

- XP, bir yazılım projesini beş temel yolla iyileştirir; basitlik, iletişim, geri bildirim, saygı ve cesaret.
- Programcılar, müşteriler ve diğer programcılarla sürekli iletişim halindedir. Tasarımlarını basit ve temiz tutarlar. Yazılımlarını ilk günden itibaren test ederek geri bildirim alırlar. Sistemi mümkün olan en erken zamanda müşterilere teslim ederler ve önerilen deęişiklikleri uygularlar. Her küçük başarı, her bir ekip üyesinin benzersiz katkılarına olan saygılarını derinleştirir. Bu temel sayesinde programcılar deęişen gereksinimlere ve teknolojiye cesurca yanıt verebilir.



Basitlik

- XP modelinin temel ilkesi, zaman alacak ve asla kullanılmayacak bir şey inşa etmeye çalışmaktan ziyade, şu anda verimli bir şekilde çalışacak basit bir sistem geliştirmektir.
- Gelecekteki gereksinimler üzerine spekülasyonlar yapmak için zaman ve çaba harcamak yerine, hemen ihtiyaç duyulan bazı belirli özelliklere odaklanır.
- Basitlik, "işe yarayacak en basit şey nedir?" anlamına gelir.
- Bunun amacı israfı önlemek ve yalnızca gerekli olan şeyleri yapmaktır. Örneğin; sistemin tasarımını mümkün olduğunca basit tutarsanız bakımı, desteği ve değişikliği daha kolay olur.
- Basitlik yalnızca bildiğiniz gereksinimleri ele almak anlamına gelir; geleceği tahmin etmeye çalışılmamalıdır.

Basitlik

- Gereken ve isteneni yapılır, daha fazlasını değil.
- Bu, bugüne kadar yapılan yatırım için yaratılan değeri en üst düzeye çıkaracaktır.
- Hedefe doğru küçük ve basit adımlar atılır ve başarısızlıklar meydana geldikçe azaltılır.
- Gurur duyulacak bir şey yaratılır ve makul maliyetlerle uzun vadede korunur.

İletişim

- Yazılım geliştirme, bilgiyi bir takım üyesinden takımdaki diğer herkese aktarmak için iletişime dayanan, doğası gereği bir takım sporudur.
- XP, uygun iletişim türünün önemini vurgular - bir beyaz tahta veya başka bir çizim mekanizmasının yardımıyla yüz yüze görüşme.
- Herkes ekibin bir parçasıdır ve her gün yüz yüze iletişim kurulur.
- Gereksinimlerden kodlara kadar her şey üzerinde birlikte çalışılır.
- Sorunlara birlikte yaratabilecek en iyi çözümü yaratılır.

Geri bildirim

- XP modelinin en önemli yönlerinden biri, müşterinin tam ihtiyaçlarını anlamak için geri bildirim almaktır.
- Müşteriyle sık sık temas kurmak, geliştirmeyi etkili hale getirir.
- Önceki çabaları hakkında sürekli geri bildirim yoluyla, ekipler iyileştirme alanlarını belirleyebilir ve uygulamalarını gözden geçirebilir.
- Yazılımı erken ve sık sık gösterip ardından dikkatlice dinleyip gereken değişiklikleri yapılır.
- Geri bildirim ayrıca basit tasarımı da destekler.
- Ekibiniz bir şeyler inşa eder, tasarımınız ve uygulamanız hakkında geri bildirim toplar ve ardından ürününüzü ileriye dönük olarak ayarlar.

Saygı

- Ekip üyelerinin birbirleriyle iletişim kurabilmeleri, geri bildirimler sağlayıp kabul edebilmeleri ve basit tasarımlar ve çözümler belirlemek için birlikte çalışabilmeleri için birbirlerine saygı duymaları gerekir.
- Herkes değerli bir ekip üyesi olarak hak ettiği saygıyı gösterir ve hisseder.
- Herkes, sadece coşku olsa bile, değer katar.
- Geliştiriciler müşterilerin uzmanlığına saygı duyar ve tam tersi de geçerlidir.
- Yönetim, sorumluluk alma ve kendi işimiz üzerinde yetki alma hakkına saygı duyar.

Cesaret

- Kent Beck cesareti "korku karşısında etkili eylem" olarak tanımlamıştır.
- Bu tanım, sonuçların takıma zarar vermemesi için diğer ilkelere dayalı eyleme yönelik bir tercihi gösterir.
- Takımınızın etkinliğini azaltan organizasyonel sorunları gündeme getirmek için cesarete ihtiyacınız vardır. İşe yaramayan bir şeyi yapmayı bırakıp başka bir şey denemek için cesarete ihtiyacınız vardır. Kabul etmesi zor olsa bile geri bildirimi kabul edip ona göre hareket etmek için cesarete ihtiyacınız vardır.

XP İlkeleri

- Değerlerin aksine, ilkeler daha sağlam temellere dayanan, gerçekçi ve somut fikirlerdir.
- İlkeler, ekiplerin özellikle ne yapması gerektiğini tanımlar ve ekip üyelerinin birbirlerini XP'nin değerlerine karşı sorumlu tutmaları için bir araç sağlar.
- Beş ilke şunlardır:
 - Hızlı geri bildirim
 - Varsayılan basitlik
 - Artımlı değişim
 - Değişimi benimseme
 - Kaliteli çalışma

XP ilkeleri

- **Hızlı geri bildirim:** Bu ilke, geri bildirim hızı bir şekilde almak ve buna hızlı bir şekilde yanıt vermek ve ertelememek anlamına gelir.
- **Varsayılan basitlik:** Ekip üyeleri enerjilerini en yüksek önceliğe sahip olan göreve yönlendirmeli ve gereksiz veya tekrarlayan işlerden kaçınmalıdır.
- **Kademeli değişiklikler:** Küçük değişiklikleri adım adım yapmak, hepsini biriktirip aynı anda halletmelerine izin vermekten daha iyidir.
- **Değişimi benimseyin:** Müşteri ürünü değiştirmek isterse, programcılar fikri desteklemeli ve yeni değişiklikleri nasıl dahil edeceklerini planlamalıdır.
- **Kaliteli iş üretin:** Birlikte iyi çalışan bir ekip kaçınılmaz olarak üstün bir ürün yaratacak ve sonuçtan gurur duyacaktır.

XP model



Kullanıcı Hikayeleri (User Stories)

- Büyük bir gereksinim belgesi yerine kullanılırlar.
- Kullanıcı hikayeleri, müşteriler tarafından sistemin onlar için yapması gereken şeyler olarak yazılır.
- Sürüm planlama toplantısı için zaman tahminleri oluşturmak için kullanılırlar.
- Müşterinin, teknik sözdizimi olmadan müşterinin terminolojisinde yazdığı yaklaşık üç cümlelik metin biçimindedirler.

Sürüm Planlama (Release Planning)

- Bir sürüm planlama toplantısı, genel projeyi ortaya koyan bir sürüm planı oluşturmak için kullanılır.
- Sürüm planı daha sonra her bir yineleme için yineleme planları oluşturmada kullanılır.
- Teknik kişilerin teknik kararları, iş kişilerinin ise iş kararlarını alması önemlidir.
- Sürüm planlaması, projeye dahil olan herkesin kendi kararlarını almasını sağlayan bir dizi kurala sahiptir.
- Kurallar, herkesin taahhüt edebileceği bir zaman çizelgesi üzerinde pazarlık yapmak için bir yöntem tanımlar.

Kabul Testleri (Acceptance Tests)

- Kabul testleri kullanıcı hikayelerinden oluşturulur.
- Bir yineleme sırasında yineleme planlama toplantısı sırasında seçilen kullanıcı hikayeleri kabul testlerine çevrilecektir.
- Müşteri, bir kullanıcı hikayesi doğru şekilde uygulandığında test edilecek senaryoları belirtir.
- Bir hikaye, işlevselliğin çalışmasını sağlamak için ne gerekiyorsa, bir veya daha fazla kabul testine sahip olabilir.

XP Pratikleri

- Planlama oyunu (Planning game)
- Küçük yayımlar (Small releases)
- Sistem metaforu (Metaphor)
- Basit tasarım (Simple design)
- Test (Testing)
- Yeniden düzenleme (Refactoring)
- Eşli programlama (Pair programming)
- Ortak kod sahipliği (Collective Code Ownership)
- Sürekli entegrasyon (Continuous Integration)
- 40 saatlik çalışma haftası (40-hour workweek)
- Ekip üyesi müşteri (On-site customer)
- Kodlama standartları (Coding Standards)

Planlama oyunu

- Planlama oyununa müşteriler ve geliştiriciler katılır.
- Bir sonraki iterasyonda yapılması gereken işler belirlenir.
- Müşteriler tarafından oluşturulan kullanıcı hikayeleri önceliklendirilir.
- Teknik personeller kullanıcı hikayeleri için gerekli zamanı, maliyeti vb. belirler.
- Bir sonraki sürüm ve iterasyon planları oluşturulur.

Küçük sürümler (Small releases)

- Küçük ve kısa aralıklı müşteriye sunulan artırımlardır.
- Yeni eklenen veya değişikliğe uğrayan bileşenler yeni sürüm olarak müşteri ve kullanıcılara sunulur.
- Müşteriler hem çalışır durumdaki üründen faydalanır hem de belirlenen gereksinimlerin karşılanıp karşılanmadığı konusunda geri bildirim verir.
- Eğer yeni sürüm müşteri isteklerini karşılamıyorsa gereksinimlerde değişiklik yapılabilir.
- Planlama oyununu destekler.

Sistem metaforu (Metaphor)

- Ortaya çıkarılacak ürün için bir veya birden fazla, ürünün nasıl bir işlevi olacağını ekibin gözünde canlandırmalarını sağlayacak mecazi isim, öge ya da resimler kullanılır.
- Bunlar proje çalışanlarının ortak bir payda da buluşarak, ne yapılması gerektiği hakkında bir fikir sahibi olmalarını kolaylaştırır.
- Sistemi açıklamak için hızlı ve kolay bir yoldur.
- Nesnelere ne ad verildiği, sistemin genel tasarımını ve kodun yeniden kullanımını anlamak için çok önemlidir.

Basit tasarım (Simple design)

- Geliştiriciler ortaya çıkaracakları ürünü en basit haliyle tasarlar.
- Basit tasarımlar yazılım sisteminin karmaşıklığını önler.
- Daha kolay ve daha hızlı bir şekilde geliştirilebilir.
- İleride yazılım ürününün değiştirilebilir ve genişletilebilir olmasını kolaylaştırır.
- Basit tasarlanmış yazılımları anlamak ve anlatmak daha kolaydır.
- Yeniden düzenleme ve ortak kod sahipliği pratiklerinin uygulanmasını kolaylaştırır.

Test (Testing)

- Oluşturulan yazılım ürünlerinin kalite kontrolünü sağlamak için yazılımın gerçekleştirim aşamasında birtakım testler yapılır.
- Bileşenlerin işlevlerini kontrol etmek için birim testleri, bileşenlerin diğer bileşenler ile entegrasyonunu test etmek için entegrasyon testleri ve müşteri gereksinimlerini test etmek için kabul testleri yapılır.

Yeniden düzenleme (Refactoring)

- Yeniden düzenleme, mevcut bir yazılım parçasının yapısını veya tasarımını, işlevsellik eklemeyi veya değiştirmeyi kolaylaştıracak şekilde revize etmeye olanak tanır.
- Yazılım tasarımını bu şekilde revize edebilmek önemlidir, çünkü aksi takdirde yalnızca orijinal tasarımla bir şekilde tutarlı olan eklemeler veya değişiklikler yapılabilir.
- Bireysel yeniden düzenleme adımları küçük olmalı ve hatalardan kaçınmak için her biri kontrol edilmelidir.
- Yeniden düzenleme yönteminin temel bir özelliği, her yeniden düzenleme adımından sonra, yazılımın yeniden düzenlemeden önce olduğu gibi çalışmaya devam ettiğinden emin olmak için değiştirilmiş yazılımı test etmektir.

Eşli Programlama (Pair programming)

- İki geliştiricinin aynı iş istasyonunda birlikte çalıştığı eşli programlamayı teşvik eder.
- Bu sayede programcıların kısa bir zaman içinde aynı seviyeye gelmesi sağlanır.
- Bu yaklaşım bilgi paylaşımına yardımcı olur, hataları azaltır ve kod kalitesini artırır.

Ortak kod sahipliği (Collective Code Ownership)

- Geliştiriciler ortak sorumluluk taşırlar. Bu her kod parçasının herhangi bir programcı tarafından gerekli durumlarda değiştirilebileceği anlamına gelir.
- Böylece yapılması gereken işler aksamaz, çünkü belli kod bölümlerinden belli programcılar sorumlu değildir.
- Aksine her programcı programın her bölümü üzerinde çalışma hakkına sahiptir.
- Bir programcının işe gelmemesi durumunda, başka bir programcı kolaylıkla onun görevlerini üstlenebilir.

Sürekli entegrasyon (Continuous Integration)

- Sistem değişiklikleri ve yeni bileşenler hemen sisteme entegre edilerek test edilir.
- Sürekli entegrasyon sayesinde yapılan tüm değişiklikler her programcının sistem üzerinde yapılan değişiklikleri görmesini sağlar.
- Ayrıca sistem entegrasyonu için gerekli zaman azaltılır, çünkü oluşabilecek hatalar erken teşhis edilerek, ortadan kaldırılır.

40 saatlik çalışma haftası (40-hour workweek)

- Geliştiriciler, değişim stresinin her zaman mevcut olduğu bir ortamda çalışmaktadır.
- XP projeleri üzerinde çalışmak, projenin ömrü boyunca kalite ve performansı tutarlı bir şekilde artırmak anlamına gelir.
- Fazla mesai yapan ekiplerle kusur oranları artmaya başlar, sinirler gerilir, iletişim bozulur ve kalite düşer.
- XP'nin temel ilkesinin iş kalitesinin asla tehlikeye atılamayacağı olduğunu hatırlayarak, geliştiricilerin ne kadar çalışmaları gerektiği belirlenerek yazılımda yüksek standartları korumaları kontrol edilir.
- 40 saatlik haftanın fikri, ekip üyelerinin kaliteyi sürdürebilecekleri saatlerde çalışmaları gerektiğidir.
- Sürdürülebilir iş yükü 40 saat veya buna yakın olabilir.

Ekip üyesi müşteri (On-site customer)

- XP'ye göre son müşteri geliştirmeye tam olarak katılmalıdır.
- Müşteri, ekip sorularını yanıtlamak, öncelikleri belirlemek ve gerekirse anlaşmazlıkları çözmek için her zaman hazır bulunmalıdır.
- Ayrıntılı işlevlerle karşılaştırıldığında, kullanıcı hikayeleri genel açıklamalardır, bu da ayrıntıların geliştirici ile müşteri arasındaki iletişimle desteklenmesi gerektiği anlamına gelir.

Kodlama standartları (Coding Standards)

- Bir ekibin, kod yazımında aynı formatları ve stilleri kullanarak ortak kodlama uygulamaları olması gerekir.
- Standartların uygulanması, tüm ekip üyelerinin kodu kolayca okumasını, paylaşmasını ve yeniden düzenlemesini, belirli kod parçaları üzerinde kimin çalıştığını izlemesini ve diğer programcılar için öğrenmeyi daha hızlı hale getirmesini sağlar.
- Aynı kurallara göre yazılan kod, ortak kod sahipliğini teşvik eder.

XP roller

- XP projesinin çalışması için belirli roller gereklidir ve bu rolleri üstlenen kişiler, karşılık gelen sorumlulukları üstlenir ve bu rollere yaptıkları katkılardan sorumludur.
- İnsanları bu rollere uydurmaya çalışmak yerine, rollere doğru kişileri atamanız önerilir.
- İnsanları sistemin merkezine koyar ve iletişim, işbirliği, yanıt verme ve geri bildirim gibi sosyal becerilerin değerini ve önemini vurgular.
- XP roller:
 - Müşteri (Customer)
 - Programcı (Programmer)
 - Koç (Coach)
 - Takipçi (Tracker)
 - Testçi (Tester)
 - Yönetici (Manager)

XP roller

- **Müşteri (Customer)**

- Kullanıcı hikayeleri yazar.
- Fonksiyonel testleri belirler.
- Öncelikleri belirler, hikayeleri açıklar.
- Son kullanıcı olabilir veya olmayabilir.
- Hikayelerle ilgili sorulara karar verme yetkisine sahiptir.

- **Programcı (Programmer)**

- Hikayeleri tahmin eder.
- Hikayelerden görevleri tanımlar.
- Hikayeleri ve birim testlerini uygular.

XP roller

- **Koç (Coach)**

- Her şeyi izler.
- Projenin yolunda gitmesini sağlar.
- Her konuda yardımcı olur.

- **Takipçi (Tracker)**

- Programcıların ilerlemesini izler, işler yolunda gitmiyor gibi görünüyorsa harekete geçer.
- Eylemler arasında müşteri ile bir toplantı ayarlar.
- Gerektiğinde koç veya başka bir programcıya yardım eder.

XP roller

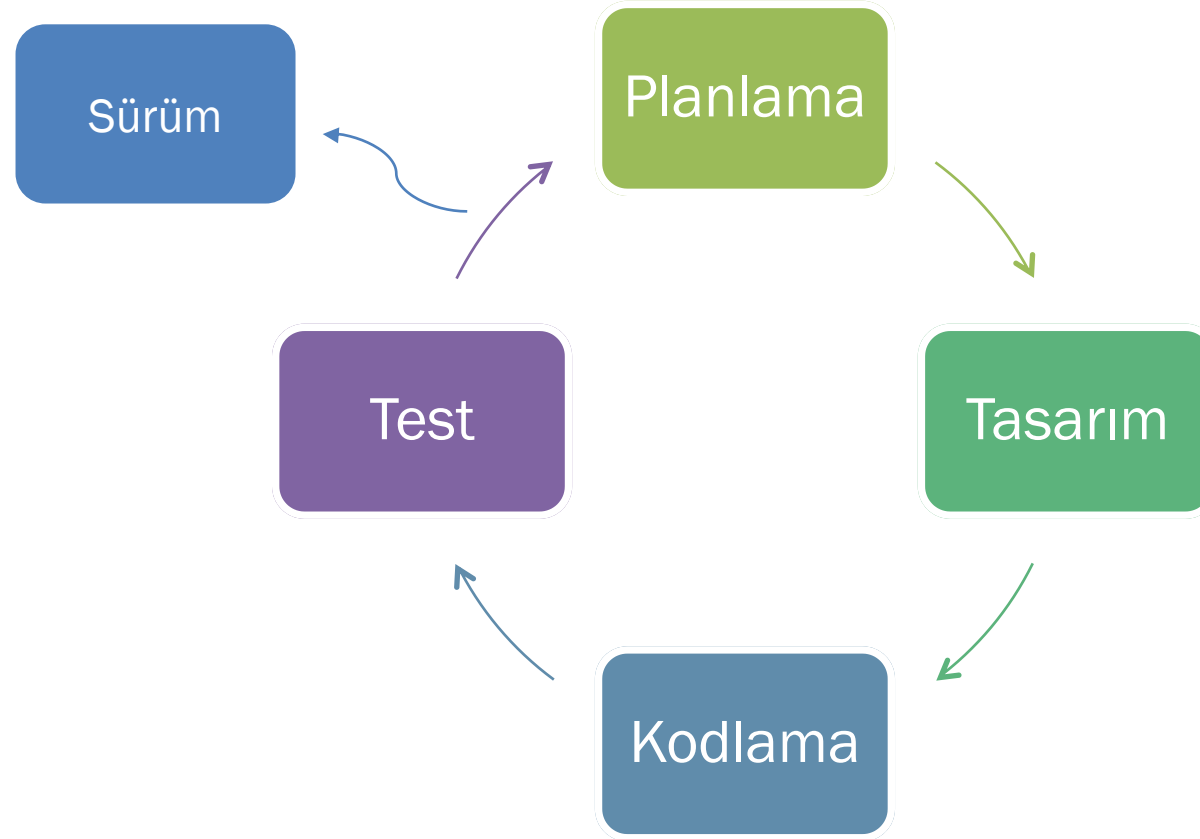
- **Testçi (Tester)**

- Fonksiyonel testleri uygular ve çalıştırır (Birim Testleri değil!).
- Sonuçları grafiksel olarak gösterir.
- Test sonuçları düştüğünde insanların bilmesini sağlar.

- **Yönetici (Manager)**

- Toplantıları planlar (örneğin, iterasyon planı, sürüm planı).
- Toplantı sürecinin takip edildiğinden emin olur.
- Gelecekteki raporlama için toplantı sonuçlarını kaydeder ve izleyiciye iletir.

XP Yaşam Döngüsü



XP Yaşam Döngüsü

■ Planlama

Müşterinin geliştirme ekibiyle buluştuğu ve istenen sonucu tanımlamak için kullanıcı hikayeleri biçiminde gereksinimleri sunduğu zamandır.

Daha sonra ekip hikayeleri tahmin eder ve gerekli işlevselliği parça parça kapsayacak şekilde gereken yinelemelere ayrılmış bir sürüm planı oluşturur.

Hikayelerden bir veya daha fazlası tahmin edilemiyorsa, bu daha fazla araştırmanın gerekli olduğu anlamına gelir.

■ Tasarım

Aslında planlama sürecinin bir parçasıdır, ancak önemini vurgulamak için ayrı tutulabilir.

İyi bir tasarım sisteme mantık ve yapı getirir ve gereksiz karmaşıklıklardan ve yedekliklerden kaçınmayı sağlar.

XP Yaşam Döngüsü

■ Kodlama

Kodlama standartları, eşli programlama, sürekli entegrasyon ve ortak kod sahipliği gibi belirli XP pratiklerinin uygulanmasıyla gerçek kodun oluşturulduğu aşamadır.

■ Test

XP programlamanın özüdür.

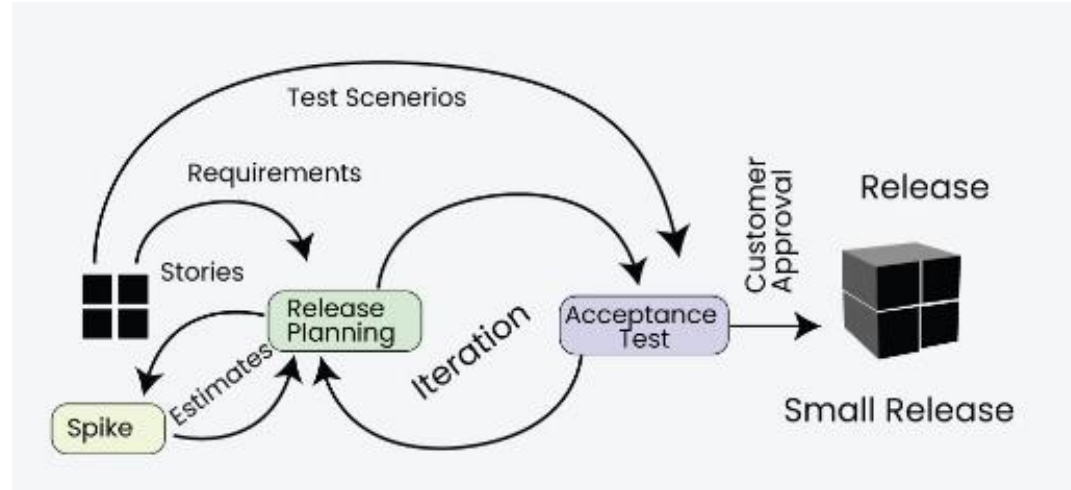
Hem birim testlerini (geliştirilen özelliğin düzgün çalışıp çalışmadığını belirlemek için otomatik test) hem de kabul testlerini (genel sistemin ilk gereksinimlere göre oluşturulduğunu doğrulamak için müşteri testi) içeren düzenli bir aktivitedir.

XP Süreci

- Programlama süreci ekibin yazılımı sürekli olarak geliştirmesine ve özelleştirmesine olanak tanıyan birkaç adımdan oluşur.
- İlk olarak, ekip müşterinin ihtiyaçlarını analiz eder ve yazılım için bir gereksinim belirtimi geliştirir.
- Daha sonra, ekip bir sonraki yinelemede neyi başarmak istediğini planlar ve hedeflere ulaşmak için hangi işlevselliğin gerekli olacağını belirler.
- Ardından, ekip yazılımın yapısını tasarlar ve onu uygulamak için gereken araçları ve teknolojileri belirler.
- Sonuç olarak, kodu uygulamak, birim testleriyle test etmek, diğer yazılım parçalarıyla entegre etmek, sürekli yeniden düzenleme, sistem ve kabul testlerini gerçekleştirmek ve yazılımı müşteriye teslim etmek gerçekleşir.

XP Süreci

- Son olarak, ekip hangi şeylerin iyi gittiğini ve hangilerinin gitmediğini ve gelecekte nelerin iyileştirilebileceğini değerlendirmek için bir retrospektif gerçekleştirir.
- Tüm bu adımlar, ekibin müşterinin ihtiyaçlarına sürekli olarak uyum sağlamasını ve değişen koşullara hızlı bir şekilde yanıt vermesini sağlayan küçük yinelemelerde gerçekleştirilir.



XP Avantaj ve Dezavantajları

Avantajlar

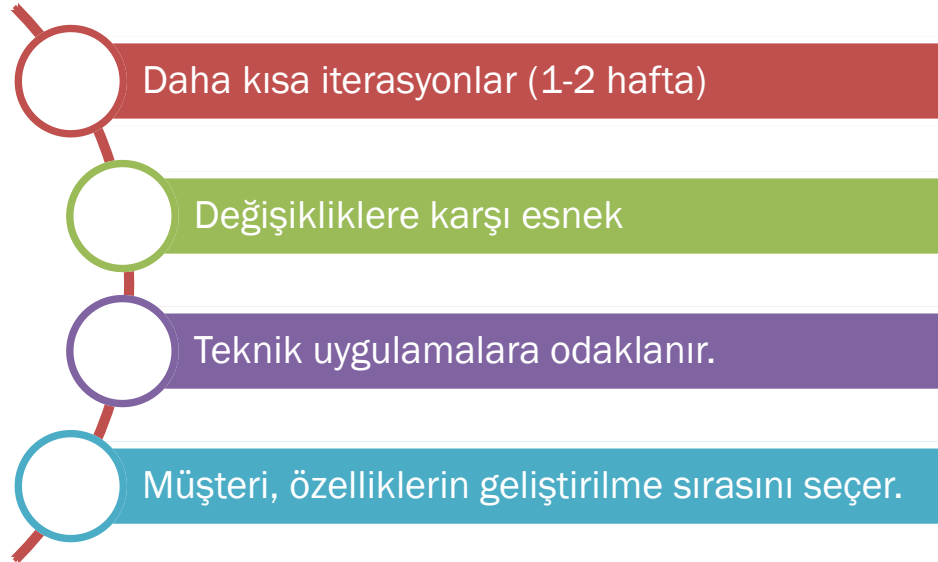


Dezavantajlar



XP – Scrum Karşılaştırma

XP



SCRUM

