



Extreme Programming

Özcan Acar
Bilgisayar Mühendisi

<http://www.KurumsalJava.com>

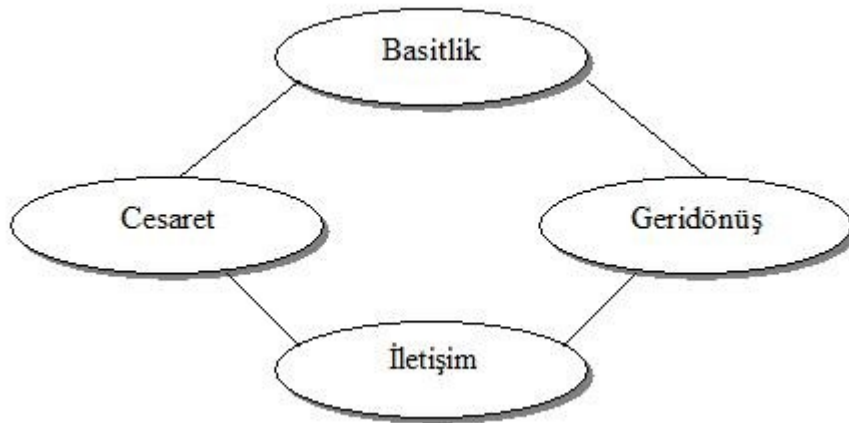
XP Değerleri.....	3
XP Prensipleri.....	4
XP Teknikleri.....	7
XP Roller.....	10
Haklar ve Sorumluluklar.....	12
Süreç İşleyişi.....	13
XP Proje Safhaları.....	13

En popüler çevik süreçlerden (Agile Process) birisi XP olarak bilinen Extreme Programming'dir. Kent Beck ve arkadaşları tarafından 1996 yılında Chrysler firmasında yapılan bir proje bünyesinde oluşan XP, ihtiva ettiği basit ama bir o kadar etkili yöntemlerle yazılım sektöründe yeni bir rüzgarın esmesini sağlamıştır.

XP ile oluşturulan çevik süreçte müşteri ve gereksinimleri merkezi bir rol oynamaktadır. Yazılım esnasında XP ile tam belirli olmayan ve çabuk değişikliğe uğrayan müşteri gereksinimlerine ayak uydurulabilir. Bu konvensiyonel yazılım metodlarda mümkün değildir, çünkü proje öncesi müşteri gereksinimleri en son detayına kadar kağıda dokülmüştür. Oluşan bir dokumentasyon baz alınarak, yazılım gerçekleştirilir. Proje ilerledikçe müşteri tarafından yapılması istenen değişikliklerin maliyeti çok yüksek olacaktır, çünkü mevcut yapı (tasarım – design) istenilen değişikliklerin yapılmasını engelleyebilir yada yeni bir yapılanmaya gidilmesi gerekebilir. XP, kullanıldığı projelerde formalite ve bürokrasinin mümkün en az seviyeye çekilmesine önem verir. Çevik olabilmek için az yükü yola çıkılması gerekmektedir. Bu yüzden proje öncesi geniş çapta tasarım ve dokumentasyon oluşturulması izin verilmez. Bu kesinlikle dokümantasyon ve tasarım oluşturulmadan çalışıldığı anlamına gelmez!

XP Değerleri

XP dört değer üzerine kuruludur: Basitlik (Simplicity), İletişim (Communication), Geridönüm (Feedback) ve Cesaret (Courage).



XP'nin özü bu dört değer içinde yatmaktadır. Bu değerler yaşandığı takdirde XP öğrenimi ve kullanımı kolaylaşır. Bu değerlerin geçerlilik bulmadığı ortamlarda XP'nin uygulandığı söylenemez. XP ile verimli bir çevik süreç oluşturabilmek için bu değerlerin hepsinin kabul görmesi ve uygulanması gerekmektedir.

Bu değerlerin ne anlama geldiğini yakından inceliyelim. XP basit yöntemler aracılığıyla sonuca ulaşmak ister, çünkü sadece bu şekilde hızlı ve düşük maliyetli projeler gerçekleştirilebilir. Bunun yanı sıra basit çözümlerle oluşturulan programın bakımı ve geliştirilmesi kolaydır. Basit çözümler kolay anlatılır ve adapte edilir. Bu zaman kazanılması anlamına gelmektedir.

Yazılımda en önemli konulardan birisi kalite kontrolüdür. XP projelerinde kalite kontrolü geridönüm (feedback) üzerinden sağlanır. Programcılar yazdıkları testlerden geridönüm

arak kaliteyi sağlarlar. Kısa zamanlarla yeni sürüm oluşturularak müşteri ve kullanıcılardan geridönüm aracılığıyla programın gereksinimleri tatmin edip, etmediği kontrol edilir. Yazılım esnasında sürekli entegrasyon yapılarak, programın en son durumu hakkında geridönüm sağlanır. XP'nin uygulanabilmesi için değişik katmanlarda geridönüm mekanizmalarının oluşturulması gerekmektedir. İnsanlar için su ne ise, XP için geridönüm odur.

Tüm proje çalışanlarının sürekli olarak aralarında iletişim kurmaları gerekmektedir. Bireyler arası yüz yüze görüşmeler büyük önem teşkil etmektedir. Sadece bu sayede sağlıklı bilgi transferi gerçekleşebilir. Böylece yanlış anlaşılımlar ve bilinmeyenler ortadan kaldırılır. Eğer takım içinde iletişim ve interaksyon güçlü ise, dokümentasyon oluşturma ve kullanma gereksiz hale gelebilir. Bu zaman kaybını önler. Dokümentasyon yazılımı ve kullanımı başka sebeplerden dolayı gerekli olabilir, ama projenin başarısı için dokümentasyon öncelikli rol oynamamaktadır.

Basit çözümler, geridönüm ve iletişim için cesaret gereklidir. Bu saydığımız değerler bireyler arası interaksyonu ve iletişimi artıracığı için, bireylerin kendi iç dünyalarını terk edip, takımın bir parçası olmalarını kolaylaştırır. Bu kişisel gelişmeyi sağlar ve temelinde kişisel cesaret yatar.

XP Prensipleri

XP değerlerinden yola çıkarak onbeş XP prensibi oluşturulmuştur. Bunlar:

Rapid Feedback

Hızlı geridönüm

Sık ve hızlı geridönüm edinmek, projenin gidişatını olumlu etkiler. Geridönüm sayesinde yanlış anlaşılımlar ve hatalar ortadan kaldırılır.

Assume Simplicity

Basitliği tercih etmek

Basit çözümler kolay implemente edilir ve kısa zamanda oluşturulur. Bu geridönümün de hızlı bir şekilde gerçekleşmesini sağlar. Basit çözümlerin kavranması ve anlatılması daha kolaydır. XP programcılardan o anki gereksinimi tatmin etmek için basit çözümü bekler. Programcı gelecekte oluşabilecek eklemeleri ve değişiklikleri düşünmemeli, sadece ve sadece kendisinden o an için bekleneni en basit haliyle implement etmelidir.

Incremental Change

İnkrementel değişiklik

Basit çözümler uygulasa bile, yazılım sistemleri zaman içinde karmaşık bir yapıya dönüşebilir. Yapılan en ufak bir değişiklik bile, sistemin düşünmediğimiz bir bölümü üzerinde hata oluşmasına sebep verebilir. Oluşabilecek bu hataları kontrol altında tutabilmek için değişikliklerin ufak çapta olması gerekmektedir. Büyük değişiklikler beraberinde büyük sorunları getirebilir. Bu sebepten dolayı değişikliklerin ufak çapta ve sıklıkla yapılması gerekmektedir.

Embracing Change

Değişimi istemek

İlerliyelemek için kendimize bir yön tayin etmemiz ve yeniliklere açık olmamız gerekiyor. Yeniliklere açık olmak cesaret gerektirir. Bilinmeyenle uğraşmak, rahatsız edici olabilir, ama başarıyı elde edebilmek için değişimi istemek gerekir.

Quality Work

Kaliteli iş

XP projelerinde kaliteli işin ortaya konabileceği bir ortamın oluşturulması gerekmektedir. Hiçbir programcı hatalı program yazmak istemez. Çalışma ortamında etkisiyle yüksek kalitede yazılım yapmak hem programcının özgüvenini artırır, hemde müşteriyi tatmin edici ürünlerin ortaya konmasını sağlar.

Teach Learning

Öğrenmeyi öğret

XP programcı takımlarında tertipcilik ve kıdem farkı yoktur. Tecrübeli programcılar bilgilerini daha az tecrübeli programcılarla paylaşarak, hem bilginin çoğalmasını sağlarlar, hemde takım arkadaşları ile teknik olarak aynı seviyeye gelirler. Programcılara komutlar vererek iş yaptırmak yerine, kendiliğinden bazı şeyleri öğrenerek, görevlerini yerine getirmeleri sağlanmalıdır.

Small Initial Investment

Az başlangıç yatırımı

XP en modern ve pahalı araç gereçlerle projeye başlanmasını beklemez. Başlangıç giderleri ne kadar düşük tutulabilirse, projenin iptali durumunda kayıplar o oranda az olacaktır. Başlangıçta tüm takımın dar bir finansman korsasını giymesi sağlanarak, proje için daha önemli görevlere odaklanmaları sağlanır. Bunu bir örnekle açıklayabiliriz. Proje başlangıcında en son Oracle bilgibankası ve kullanıcı araçları (Toad gibi bir client program) satın alınarak, tüm ekibe bu bilgibankasını ve araçları nasıl kullanacaklarına dair seminer verilebilir. Bu çok masraflı ve bir o kadarda gereksiz bir şeydir. Projeye open source ve ücretsiz olan HSQL yada PostgreSQL bilgibankası ile başlanabilir. Programcı ekip böylece ne bir hafta tanımadıkları bir ürün için harcamış olurlar, ne de büyük masraflar yapılarak şu an için gerek olmayan bir altyapı komponenti satın alınmış olur. Gerekli araçlar zamanı geldiğinde edinilmelidir.

Play to win

Kazanmak için oyna

XP takımları kazanmak için oynar. Her zaman gözlerinin önünde nihayi sonuç vardır: programı tamamlamak ve müşteriye teslim etmek. XP programcı takıma tünelin sonundaki ışığı görmek için gerekli tüm imkanları sunar.

Concrete Experiments

Somut denemeler

Verdiğimiz kararların sonuçlarını kontrol edebilmek için denemeler yaparız, çünkü alınan kararlar her zaman doğru olmayabilir. Bir kontrol mekanizmasına ihtiyacımız olduğu belli. Bu da somut denemeler aracılığıyla nerede olduğumuzu testpit etmektir. Bu somut denemeler yazılım sistemleri içinde geçerlidir. Örneğin testler hazırlıyarak, oluşturduğumuz mimari ve tasarımı kontrol ederiz.

Open, honest Communication

Açık ve samimi iletişim

Projenin başarılı olabilmesi için bireyler arasında açık ve samimi türde bir iletişimin olması gerekmektedir. Birçok projede bu böyle değildir. Çogu zaman bireylerin korkuları, deneyimsiz olmaları yada kendilerini çok beğenmeleri ve diğerlerini kendilerinden alt safhada görmeleri, açık ve samimi bir iletişim ortamının oluşmasını engeller.

Work with people's instincts, not against them

Takımın içgüdülerini kullan, onlara karşı koyma

Bireysel içgüdü yanı sıra, bireylerin oluşturduğu takımların da içgüdü vardır. Eger takım bir şeylerin doğru gitmediği hissine sahipse ve bunu dile getiriyorsa, o zaman bir şeyler yolunda gitmiyor demektir. Takımın içgüdüne kulak verilmelidir. Bunun göz ardı edilmesi, proje için olumsuz sonuçlar doğurabilir.

Accepted Responsibility

Sorumluluk üstlenmek

Sorumluluk birilerine verilmemeli, bireyler kendileri sorumluluk üstlenmelidir. Eğer bir bireye ya da bir takıma yapılması zor bir projenin sorumluluğu yüklenirse, bu birey ya da takım için motivasyonun düşmesini ve kaybetme korkusunun pekişmesini hızlandırır. Eğer bireyler ya da takımlar kendi sorumluluklarını kendileri seçerlerse, hem yaptıkları işte kendilerini iyi hissederler, hem de yüksek motivasyon ile üstlendikleri işi başarıyla tamamlarlar.

Local Adaptations

Sürecin ortam şartlarına adapte edilmesi

Büyük bir ihtimalle her takımın XP'yi Kent Beck'in anlattığı tarzda harfiyen uygulaması mümkün değildir. Atalarımızda dediği gibi her yiğidin yogurt yeyiş tarzı başkadır. Amaç XP'yi harfiyen uygulamak değildir, amaç kısa bir zamanda projeyi başarılı bir sonuca ulaştırmaktır. Eğer proje XP'de yapılacak modifikasyonlarla başarıya ulaşacaksa, o zaman süreç üzerinde bu modifikasyonlar yapılmalı ve uygulanmalıdır. Bunda bir sakınca yoktur.

Travel light

Az yükle yolculuk yapmak

Projede hızlı ilerleyebilmek için fazla bir yükle yola çıkılmaması gerekmektedir. Beraber çalışmayı kolaylaştırmak için kullanımı kolay araç ve gereçler seçilmelidir. Formalitelerden uzak durulmalıdır.

Honest Measurement

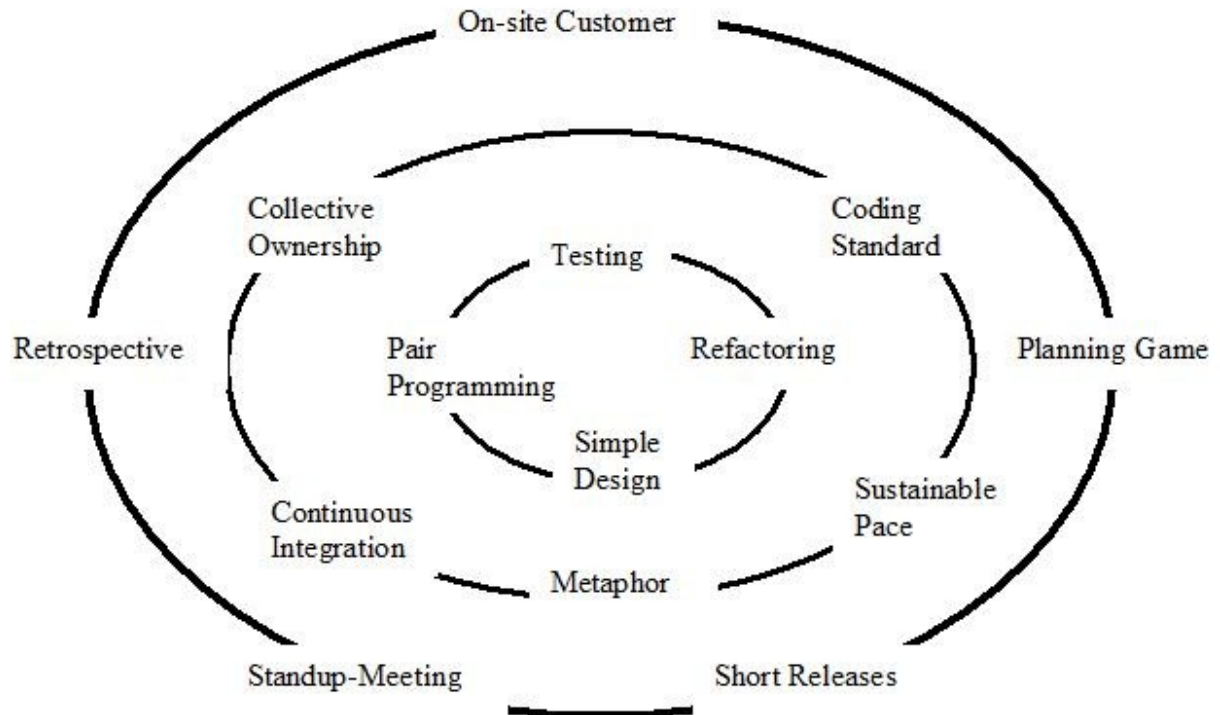
Doğru ölçüm

Proje gidişatını kontrol edebilmek için değişik türde ölçümlerin yapılması gerekmektedir. Örneğin hazırlanan unit testleri ile sınıfların işlevleri kontrol edilir. Yapılan ölçümler doğru ve samimi yapıldığı takdirde kontrol mekanizması olarak kullanılan ölçümlerin bir anlamı vardır. Programcılar tarafından samimi ve doğru yapılmayan ölçümler projenin gidişatını olumsuz yönde etkiler.

XP Teknikleri

Dört XP değer ve onbeş XP prensibi ondört XP tekniği ile desteklenmektedir. XP teknikleri programcıların XP değer ve prensiplerini uygulamada yardımcı olur. Kent Beck tarafından hazırlanan ilk XP versiyonunda oniki teknik yer almaktaydı. Diğer çevik süreçlerin de etkisiyle Standup-Meeting'ler ve retrospektif toplantılar XP teknikleri arasına katıldı.

Bunlar:



On-site Customer

Programcıya yakın müşteri olarak tercüme edilebilir.

XP projeleri müşteri gereksinimlerine odaklı ilerler. Bu yüzden müşteri ve sistem kullanıcılarının projeye dahil edilmeleri gerekmektedir. Müşteri gereksinimlerini ekibe bildirir. Programacıların implementasyonu gerçekleştirebilmesi için müşteri tarafından dile getirilen gereksinimleri anlamaları gerekmektedir. Yanlış anlaşılmalara ve hatalara giderek için programcıların müşteri ve sistem kullanıcıları ile diyalog halinde olabilmesi gerekmektedir. Bu sebepten dolayı müşteri veya sistem kullanıcılarının programcıların erişebileceği bir uzaklıkta olmaları gerekir. Tipik XP projelerinde müşteri ve programcılar aynı odada beraber çalışırlar. Müşteri ekibin sorularını zaman kaybı olmadan cevaplar ve projenin ilerlemesine katkıda bulunur.

Standup-Meeting

Ayakta toplantı

Proje çalışanları her gün 15 dakikayı aşmayan ve ayakta yapılan toplantılarda biraraya gelirler. Bu toplantının amacı, projenin gidişatı hakkında bilgi alışverişinde bulunmaktır.

Planning Game

Planlama oyunu

XP projeleri iteratif ve inkrementel yol alır. Bir sonraki iterasyonda yapılması gereken işleri planlama oyununda görüşülür ve sürüm ve iterasyonun içeriği tespit edilir. Planlama oyununa müşteri, kullanıcılar ve programcılar katılır. Müşteri ve kullanıcılar daha önce kullanıcı hikayesine (user story) dönüştürdükleri isteklerine öncelik sırası verirler. Programcılar her kullanıcı hikayesi için gerekli zamanı tahmin ederler. Kullanıcı hikayelerinin öncelik sırası bu tahmine bağımlı olarak değişebilir. Planlama oyunlarında sürüm ve iterasyon planları oluşur.

Short Releases

Kısa aralıklarla yeni sürüm

XP projelerinde yeni implemente edilen ve değişikliğe uğrayan komponentler yeni sürümler oluşturularak müşteri ve kullanıcının beğenisine sunulur. Bu sayede hem müşteriler çalışır durumda olan programdan faydalanabilir hem de yeni sürümü inceliyerek, gereksinimleri ile örtüşüp, örtüşmediğini kontrol edebilirler. Eğer yeni sürüm müşteriyi tatmin edecek durumda değilse, gereksinimler değişikliğe uğrayabilir. Bu değişiklikler bir sonraki iterasyonda göz önünde bulundurularak, müşteri istekleri ile yüksek derecede örtüşen bir programın oluşturulması sağlanır.

Retrospective

Geriye bakış

Proje çalışanları düzenli aralıklarla geriye bakarak, meydana gelen sorunları gözden geçirirler. Buradaki amaç gelecekte bu sorunların tekrarını önlemektir. Geriye bakış bir ila altıaylık

zaman birimleri için tüm proje çalışanları yada seçilen bireyler tarafından yapılır. Geriye bakış toplantıları yarım gün ila üç gün arasında sürebilir.

Metaphor

Mecaz

XP projelerinde hazırlanan program için bir veya birden fazla, programın nasıl bir işlevi olacağını ekibin gözünde canlandırmalarını sağlayacak mecazi isim, öge yada resimler kullanılır. Bunlar proje çalışanlarının ortak bir payda da buluşarak, ne yapılması gerektiği hakkında bir fikir sahibi olmalarını kolaylaştırır. Örneğin bir shop sistemi yazılımı yapılacak. Burada metaphor olarak alışveriş sepeti kullanılabilir. Alışveriş sepetini duyan her programcının aklında, bir shop sisteminin programlanması gerektiği fikri doğar.

Collective Ownership

Ortak sorumluluk

XP projelerinde programcılar ortak sorumluluk taşırlar. Bu her kod parçasının herhangi bir programcı tarafından gerekli durumlarla 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.

Continuous Integration

Sürekli entegrasyon

Sistem değişiklikleri ve yeni komponentler 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.

Coding Standards

Kod standartları

Programcılar tarafından aynı kalitede kod yazılımı yapılabilmesi için, kod yazarken kullanılacak kuralların oluşturulması gerekmektedir. Kodun nasıl formatlanacağı, sınıfların, metot isimlerinin ve değişkenlerin nasıl isimlendirileceği kod standartlarında yer alır.

Sustainable Pace

Kalıcı tempo

XP projelerinde programcılar haftalık belirli mesai saatlerini aşmazlar. Gereğinden fazla çalıştırılan ve yorulan bir programcıdan verimli iş yapması beklenemez. Programcılarının motivasyonun ve çalışma enerjilerinin yüksek olması için günde sekiz saatten fazla çalışmalarına izin verilmemelidir. Bazen fazla mesai saatlerine ihtiyaç olabilir. Eğer durum devamlı böyle ise, bu proje gidişatında bazı olumsuzlukların göstergesi olabilir.

Testing

Test etmek

Oluşturulan programların kalite kontrolünden geçmesi gerekmektedir. Bu yazılım esnasında oluşturulan testlerle yapılır. Programcılar komponentler için unit testleri hazırlar. Sınıf bazında yapılan bu testlerle komponentlerin işlevleri kontrol edilir. Müşteri gereksinimlerini test etmek için akseptans testleri hazırlanır. Komponentlerin entegrasyonunu test etmek için entegrasyon testleri hazırlanır.

Simple Design

Sade tasarım

Programcılar üstlendikleri görevleri (task) en basit haliyle implemente ederler. Bu programın basit bir yapıda kalmasını ve ilerde değiştirilebilir ve genişletilebilir olmasını sağlar. Sade bir tasarım yazılım sisteminin kompleks bir yapıda olmasını önler. Bunun yanı sıra basit tasarımlar daha kolay ve daha hızlı implemente edilebilir. Basit bir implementasyonu anlamak ve anlatmak daha kolaydır.

Refactoring

Yeniden yapılandırma

Tasarım hataları yazılım sisteminin daha ilerde tamir edilemeyecek bir hale dönüşmesine sebep verebilir. Bu yüzden bu hatalar hemen giderilir. Bu yeniden yapılandırma işlemine refactoring ismi verilir. Hazırlanan unit testleri ile yapılan değişikliklerin yan etkileri kontrol edilir. Bu açıdan bakıldığında unit testi olmayan bir sistem üzerinde yeniden yapılandırma işlemi hemen hemen mümkün değildir, çünkü değişikliklerin doğurduğu yan etkileri tespit etme mekanizması bulunmamaktadır.

Pair Programming

Eşli programlama

XP projelerinde iki programcı aynı bilgisayarda çalışır. Bu sayede programcıların kısa bir zaman içinde aynı seviyeye gelmesi sağlanır. Ayrıca bu kalitenin yükselmesini sağlar.

XP Roller

Bir çevik projede ekip çalışanlarının sorumluluk alanlarını tanımlamak için roller tayin edilir. Her rol beraberinde bazı sorumluluklar ve tanımlanmış haklar getirir. Bu roller statik değildir. Ekip içinde değişik kişilere değişik roller verilebilir ve daha sonra rol değişikliği yapılabilir. Proje gereksinimleri doğrultusunda yeni rollerin oluşturulması mümkündür.

Müşteri (Customer)

Projenin var olma sebebi müşteridir. Müşteri ihtiyaç duyduğu ve gereksimlerine cevap verebilecek bir yazılım sistemi için yatırım yapan kişidir. XP projelerinde şu atasözümüz geçerlidir: “parayı veren düdüğü çalar!”.

Proje bünyesinde ne programlanması gerektiğini müşteri tayin eder. Müşteri yapılması gerekenleri kullanıcı hikayeleri (user story) oluşturarak ifade eder. Programcılar müşteriye bu süreçte yardımcı olurlar. Ama kullanıcı hikayelerinin oluşturulma sorumluluğu büyük ölçüde müşteriye aittir. Her kullanıcı hikayesi yazılım sisteminin bir özelliğini tanımlar. Programcılarının implementasyonu gerçekleştirebilmeleri için kullanıcı hikayesini anlayabilmeleri gerekmektedir. Sadece bu durumda implementasyon süreci için bir tahminde bulunabilirler. Ayrıca oluşturulan kullanıcı hikayelerinin test edilebilir yapıda olması gerekmektedir.

Müşteri çalışma alanı (domain knowledge) hakkında bilgiye sahip olan kişidir. Programcılar müşteriye karşılaştıkları sorunları çözmek için sorular sorabilirler. Bu soruların cevabını en iyi verebilecek şahıs müşteridir.

Hangi kullanıcı hikayelerinin implemente edileceğine müşteri karar verir. Bu konuda müşteriye herhangi bir sınırlama getirilmez. Müşteri seçer ve programcılar implemente eder.

İmplemente edilen kullanıcı hikayelerini kontrol etmek amacıyla müşteri akseptans testleri tanımlar. Bu testler programcı yada testçi tarafından implemente edilir. Akseptans testleri kullanıcı hikayesini doğru implemente edilip, edilmediğini kontrol edici bir mekanizmadır.

Programcı

Sistem analizi, tasarım, test ve implementasyon programcılar tarafından yapılır.

Müşteri tarafından hazırlanan kullanıcı hikayelerinin implementasyon süresi programcılar tarafından tahmin edilir. Bu programcıların XP projelerinde proje planlama sürecine dahil edildikleri anlamına gelmektedir. Geleneksel projelerde bu tahminleri teknik bilgiye sahip olmayan yöneticiler yapmak zorundadır. Bu yüzden bu tahminler genelde gerçekleri yansıtmaz.

Her programcı test güdümlü (TDD - Test Driven Development) ve bir takım arkadaşıyla beraber çalışır. Pair programming olarak bilinen, iki programcının birlikte yazılım yapması, kısa zamanda kod hakkındaki bilginin tüm programcılar tarafından paylaşılmasını kolaylaştırır. Ayrıca pair programming programcılar arasında iletişimi ve takım içinde çalışabilme özelliğini artırır. Test güdümlü çalışmak bakımı ve geliştirilmesi kolay kodun oluşmasını sağlar. XP projelerinde programcılar herhangi bir satır kod yazmadan önce gerekli test sınıflarını oluşturarak implementasyona başlarlar.

Programcılar oluşturdukları testler yardımıyla hergün bir veya birden fazla şekilde modül entegrasyonu gerçekleştirirler. Sürekli entegrasyon programcılar tarafından oluşturulan modülleri entegre eden bir süreçtir. Her programcı bu süreçten geridönüm sağlayarak, kendi yaptıklarının ne derecede sisteme entegre olduğunu ölçebilir.

Proje Menajeri

Proje menajeri müşteri ve programcılarını bir araya getirir. Onların beraber çalışabilecekleri ortamların oluşmasını sağlar. XP proje menajeri tek başına proje planlamasından sorumlu

değildir. Programcılara görev atamaz, onların kendi başlarına seçim yaparak, sorumluluk almalarını kolaylaştırır. Toplantı ve diğer buluşmaları koordine eder, takımın karşılaştığı sorunları ortadan kaldırmak için gerekli olanları yapar.

Koç

Çevik süreci tanıyan ve nasıl uygulanması gerektiğini bilen experdir. Koçun görevi proje başlangıcında çevik takımı oluşturmak yada bir araya getirmek ve onlara belirli bir süre rehberlik yapmaktır. Koç projede sorun çıktığı zaman yada takım XP yöntemlerinin dışına çıktığında müdahale eder. Zaman zaman koç implementasyonda aktif olarak rol alır. Örneğin unit testlerin nasıl doğru bir yapıda oluşturulabileceğini diğer programcılara gösterebilir.

Testçi

Müşteri tarafından oluşturulan akseptans testlerini implemente eden programcıdır. Aynı zamanda JUnit ve entegrasyon testlerinin implementasyonunda takım arkadaşlarına yardımcı olur.

Haklar ve Sorumluluklar

Proje çalışanları çoğu zaman içgüdüsel korku hissedebilirler. Bunun en büyük sebebi belirsizliktir. Ne ve nasıl yapılması gerektiği bilinmediği taktirde ekip içinde huzursuzluk doğar. Bu projenin başarısını negatif etkiler.

Projenin başarılı olabilmesi için çalışanların hissettikleri korkunun azaltılması yada yok edilmesi gerekmektedir. XP bu konuda proje çalışanlarına bir takım hakların tanınması gerektiğini dikte eder. Hangi rollerin hangi haklara sahip olduğunu yakından inceliyelim.

Müşteri Hakları

Müşterinin sahip olduğu haklar şu şekilde özetlenebilir:

- Müşteri bütçe ve zaman planlaması yapabilmek için neyin yapılabilir olduğunu ve hangi zaman biriminde yapılabileceğini bilme hakkına sahiptir.
- Müşteri fikir değiştirerek, gereksinimler üzerinde değişiklik yapma ve yeni gereksinimlerin implementasyonunu talep etme hakkına sahiptir.
- Müşteri programcılar tarafından sağlanabilecek en yüksek verimi ve değeri elde etme hakkına sahiptir.
- Müşteri projede somut ilerlemeyi görme hakkına sahiptir. Bu kısa aralıklarla oluşturulan yeni sürüm ve akseptans testlerine olumlu cevap veren yeni implementasyonlarla sağlanır.
- Müşteri bir sonraki sürümde implemente edilecek kullanıcı hikayelerini seçme hakkına sahiptir.

Programcı Hakları

Programcının sahip olduğu haklar şu şekilde özetlenebilir:

- Programcı takım arkadaşlarına soru sorma ve cevap alma hakkına sahiptir.

- Programcı kullanıcı hikayeleri için implementasyon zaman tahmini yapma hakkına sahiptir. Programcı tahminler üzerinde değişiklik yapma hakkına sahiptir.
- Programcı, kendisine görev verilmesi yerine sorumluluk alma hakkına sahiptir.
- Programcının her zaman yüksek kalitede iş çıkarma hakkı vardır.
- Programcının hangi öncelik sırasına göre neyi yapması gerektiğini bilme hakkı vardır.

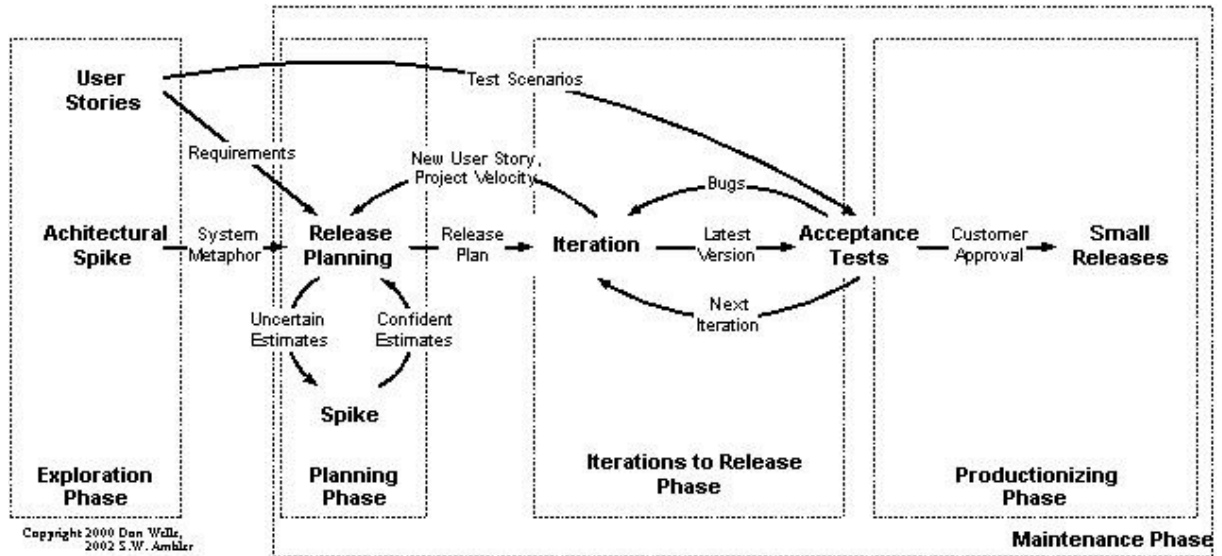
Süreç İşleyişi

XP projelerinde projenin gidişatını genel olarak şu şekilde özetliyebiliriz:

- Müşteri gereksinimleri ihtiva eden kullanıcı hikayelerini oluşturur. Bunlara öncelik sırası atar. Programcılar her kullanıcı hikayesi için implementasyon zamanını tahmin ederler.
- Müşteri programcılarla beraber iterasyon(1-2 hafta) ve sürüm (1-2 ay) planını hazırlar. Her sürüm birden fazla iterasyon ihtiva eder ve müşteri tarafından kullanılacak özellikte çalışır bir sistemdir.
- Müşteri ilk iterasyon (1 hafta) için gerekli kullanıcı hikayelerini tahminleride dikkate alarak seçer.
- Programcılar iterasyon için seçilen kullanıcı hikayelerini implemente ederler. Kafalarda oluşan sorular müşteriye danışılarak cevaplandırılır.
- İterasyon sonunda programcılar müşteriye çalışır bir sistem sunarlar. Müşteri sistemi değerlendirerek programcılara geridönüm sağlar.
- Edinilen tecrübeler ışığında bir sonraki iterasyon planlanır. Eğer müşteri yeni gereksinimlerin implemente edilmesini isterse, bunlar için tekrar kullanıcı hikayeleri oluşturulur ve tahminler yapılır. Eğer yeni kullanıcı hikayeleri yoksa, mevcut kullanıcı hikaye listesinden en yüksek öncelik sırasına sahip olanlar seçilir ve bir sonraki iterasyondan implemente edilir.
- İlk sürüm sonunda oluşan yazılım sistemi müşteri tarafından üretken olarak kullanıma alınır. Başka sürümler planlandysa bir sonraki iterasyonla devam edilir.

XP Proje Safhaları

Bir XP projesi değişik safhalardan oluşur. Her sahfa, bünyesinde kendine has aktiviteler ihtiva eder. Bir sonraki resimde bir XP projesinde olması gereken safhalar yer almaktadır.



XP projesi şu safhalardan oluşur:

- **Keşif safhası (Exploration Phase):** Projenin başlangıcında keşif safhasını oluşturan aktiviteler yer alır. Bu safhada müşteri kullanıcı hikayelerini (user story) oluşturur. Programcılar teknik altyapı için gerekli deney (spike) ve araştırmayı yaparlar.
- **Planlama Safhası (Planning Phase):** Keşif safhasını planlama safhası takip eder. Bu safhada müşteri programcılar yardımıyla iterasyon ve sürüm planlarını oluşturur. İterasyon planlaması için oluşturulan kullanıcı hikayelerinin implementasyon süresi programcılar tarafından tahmin edilir. Müşteri kullanıcı hikayelerine öncelik sırası vererek, iterasyonlarda hangi kullanıcı hikayelerinin öncelikli olarak implemente edilmeleri gerektiğini tespit eder. Programcılar tarafından herhangi bir kullanıcı hikayesinin implementasyon süresi tahmin edilemese, programcılar spike solution olarak bilinen basit bir çözüm implemente ederek, kullanıcı hikayesinin gerçek implementasyonu için gerekli zamanı tahmin etmeye çalışırlar.
- **İterasyon ve Sürüm Safhası (Iterations to Release Phase):** Kullanıcı hikayelerinin implementasyonu iterasyon ve sürüm safhasında gerçekleşir. Bir iterasyon bünyesinde implemente edilmesi gereken kullanıcı hikayeleri müşteri tarafından belirlenir. Implementasyonunun işlevini kontrol etmek için müşteri tarafından akseptans testleri belirlenir. Bu testler programcılar yada testçiler tarafından implemente edilir. Her iterasyon sonunda müşteriye, çalışan bir yazılım sistemi sunulur. Bu şekilde müşterinin sistem hakkındaki görüşleri alınır (geridönüm). İterasyon son bulduktan sonra çalışma hızını tahmin etmek için bir önceki iterasyonunda elde edilen tecrübeler kullanılır ve iterasyon planı bu değerler doğrultusunda gözden geçirilir. Bir önceki iterasyonda oluşan hatalar bir sonraki iterasyonda gözden geçirilmek ve giderilmek üzere planlanır.
- **Bakım Safhası (Maintenance Phase):** Bu programın bakımının ve geliştirilmesinin yapıldığı safhadır. Bu safhada kullanıcılar için eğitim seminerleri hazırlanır ve küçük çapta eklemeler ve sistem hatalarının giderilmesi için işlemler yapılır. Müşterinin istekleri doğrultusunda bir sonraki büyük sürüm için çalışmalara başlanır. Bu durumda tekrar keşif safhasına geri dönülmesi ve oradan işe başlanması gerekmektedir.