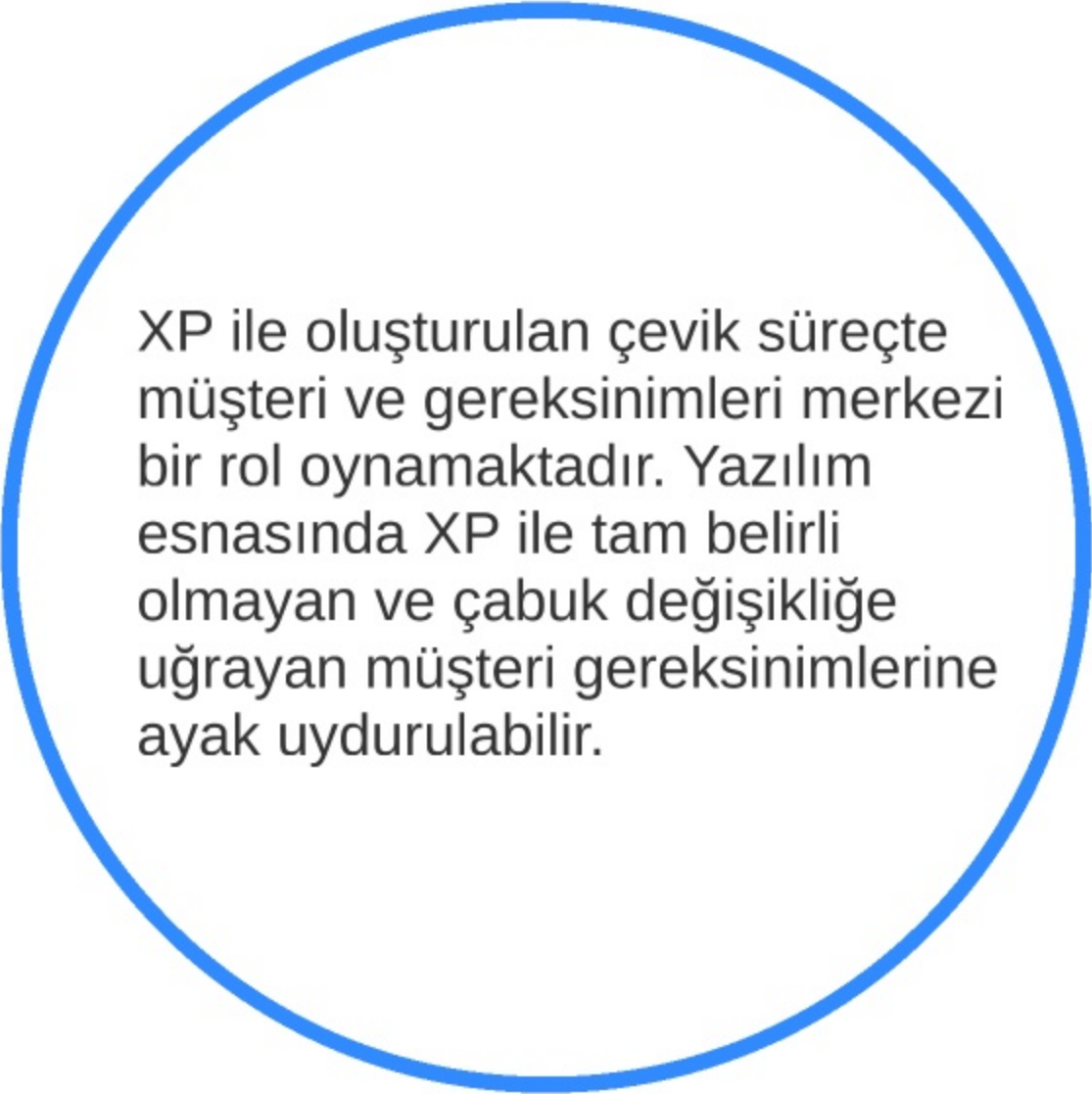


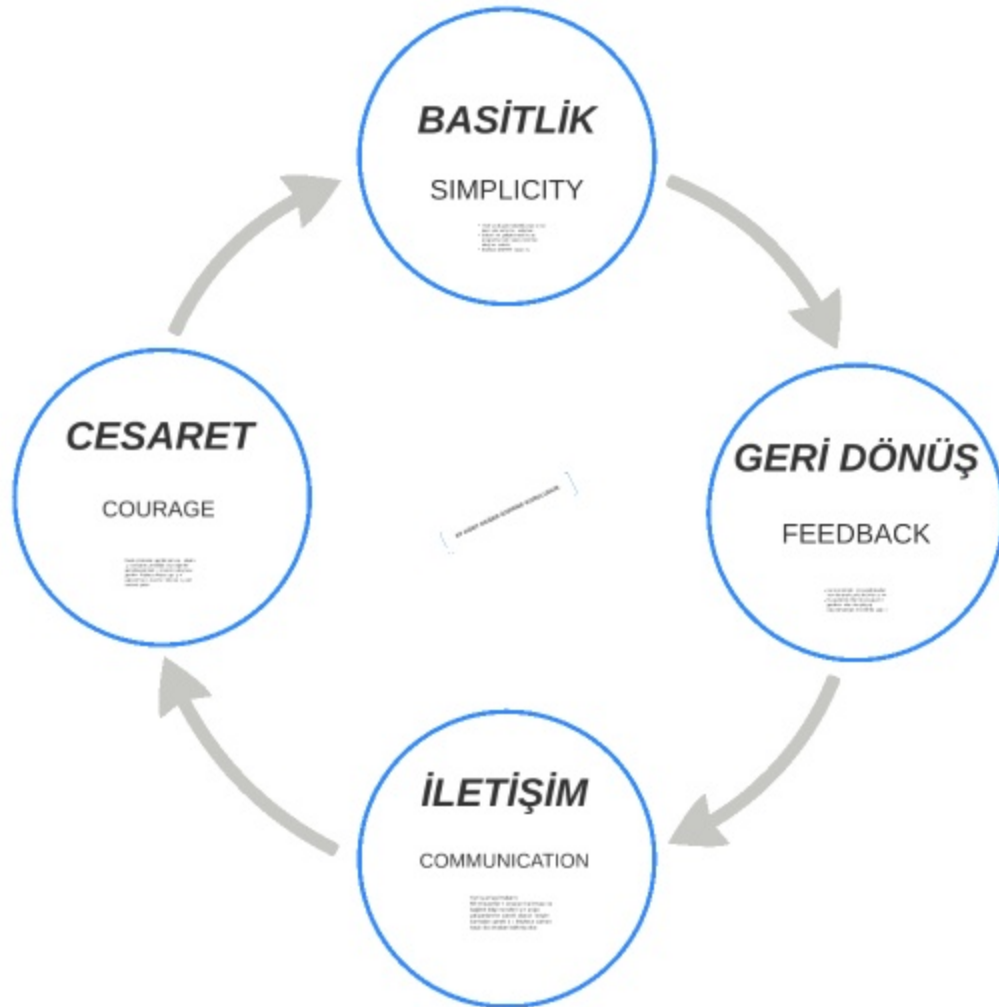
## ***EXTREME PROGRAMMING NEDİR?***

Kent Beck ve arkadaşları tarafından 1996 yılında Chysler firmasında yapılan bir proje bünyesinde oluşan XP, ihtiva ettiği basit ama 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.

**XP DÖRT DEĞER ÜZERİNE KURULUDUR.**





# ***BASİTLİK***

## SIMPLICITY

- Hızlı ve düşük maliyetli projeler için basit yöntemler kullanılırken,
- Bakımı ve geliştirilmesi kolay programlar için basit çözümler oluşturulmalıdır.
- Böylece ZAMAN kazanılır.

- Hızlı ve düşük maliyetli projeler için basit yöntemler kullanılırken,
- Bakımı ve geliştirilmesi kolay programlar için basit çözümler oluşturulmalıdır.
- Böylece ZAMAN kazanılır.

# ***GERİ DÖNÜŞ***

## **FEEDBACK**

- Kalite kontrolü için çeşitli testler hazırlanarak geridönümler alınır.
- Bu geridönümler ile programın gereksinimleri karşılayıp karşılamadığının kontrolü yapılır.

- Kalite kontrolü için çeşitli testler hazırlanarak geridönümler alınır.
- Bu geridönümler ile programın gereksinimleri karşılayıp karşılamadığının kontrolü yapılır.



# ***İLETİŞİM***

## COMMUNICATION

- Yanlış anlaşılımların,
- Bilinmeyenlerin ortadan kalkması ve
- Sağlıklı bilgi transferi için proje çalışanlarının sürekli olarak iletişim kurmaları gereklidir. Böylece zaman kaybı da ortadan kalkmış olur.

- Yanlış anlaşılmanın,
- Bilinmeyenlerin ortadan kalkması ve
- Sağlıklı bilgi transferi için proje çalışanlarının sürekli olarak iletişim kurmaları gereklidir. Böylece zaman kaybı da ortadan kalkmış olur.



# ***CESARET***

## COURAGE

Basit çözümler, geridönüm ve iletişim için cesaret gereklidir. Bu değerleri gerçekleştirmek için takım çalışması gerekir. Böylece kişisel gelişim sağlanmış olur, temelinde ise kişisel cesaret yatar.

Basit çözümler, geridönüm ve iletişim için cesaret gereklidir. Bu değerleri gerçekleştirmek için takım çalışması gerekir. Böylece kişisel gelişim sağlanmış olur,temelinde ise kişisel cesaret yatar.

- XP projeleri müşteri odaklı ilerler.
- Bu nedenle müşteri ve sistem kullanıcılarının da projeye dahil edilmeleri gereklidir.
- Böylece hatalar azaltılır ve zaman kaybı önlenmiş olur.

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

XP projeleri iteratif (tekrar ederek) ve artarak yol alır. Bir sonraki iterasyonda yapılması gereken işler 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.

XP projelerinde yeni implemente edilen ve deęişikliğe uğrayan unsurlar yeni sürümler oluşturularak müşteri ve kullanıcının beęenisine sunulur. Bu sayede hem müşteriler çalışır durumda olan programdan faydalanabilir hem de yeni sürümü inceleyerek, gereksinimleri ile örtüşüp, örtüşmediğini kontrol edebilirler.

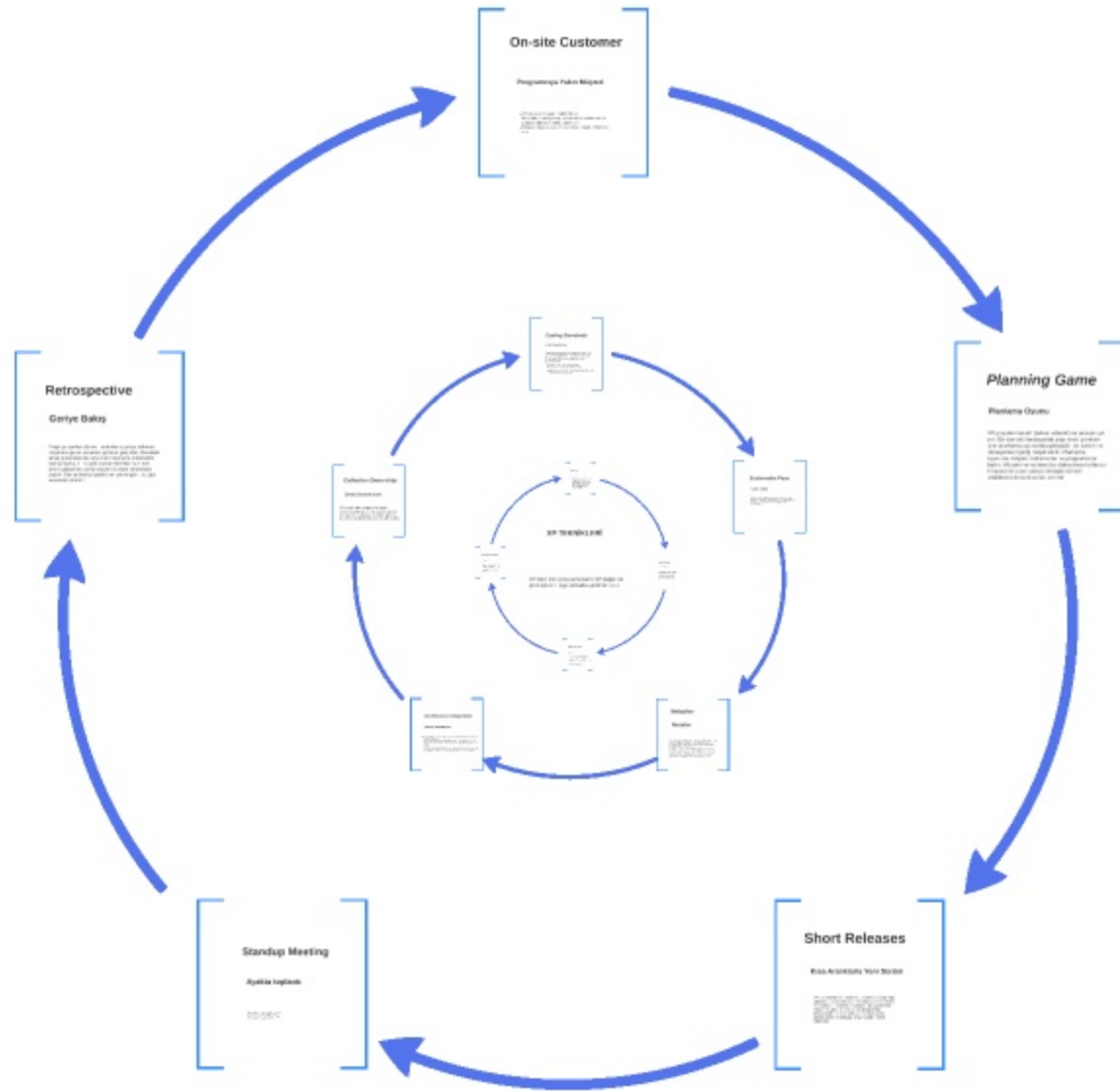


Proje çalışanları düzenli aralıklarla geriye bakarak, meydana gelen sorunları gözden geçirirler. Buradaki amaç gelecekte bu sorunların tekrarını önlemektir. Geriye bakış 1 - 6 aylık zaman birimleri için tüm proje çalışanları ya da seçilen bireyler tarafından yapılır. Geriye bakış toplantıları yarım gün - üç gün arasında sürebilir.

- XP projelerinde hazırlanan program için bir veya birden fazla, programın nasıl bir işlevi olacağını ekibin gözünde canlandırmalarını sağlayacak mecazi isim, öge yada resimler kullanılır.
- Örneğin bir shop sistemi yazılımı yapılacaktır. Burada metafor olarak alışveriş sepeti kullanılabilir. Alışveriş sepetini duyan her programcının aklında, bir shop sisteminin programlanması gerektiği fikri doğar.

# Ortak Sorumluluk

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



# On-site Customer

## Programcıya Yakın Müşteri

- XP projeleri müşteri odaklı ilerler.
- Bu nedenle müşteri ve sistem kullanıcılarının da projeye dahil edilmeleri gereklidir.
- Böylece hatalar azaltılır ve zaman kaybı önlenmiş olur.

# ***Planning Game***

## **Planlama Oyunu**

XP projeleri iteratif (tekrar ederek) ve artarak yol alır. Bir sonraki iterasyonda yapılması gereken işler 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.

# Short Releases

## Kısa Aralıklarla Yeni Sürüm

XP projelerinde yeni implemente edilen ve değişikliğe uğrayan unsurlar yeni sürümler oluşturularak müşteri ve kullanıcının beğenisine sunulur. Bu sayede hem müşteriler çalışır durumda olan programdan faydalanabilir hem de yeni sürümü inceleyerek, gereksinimleri ile örtüşüp, örtüşmediğini kontrol edebilirler.

# Standup Meeting

## Ayakta toplantı

Proje çalışanları her gün 15 dakikayı aşmayan ve ayakta yapılan toplantılarda bir araya gelirler. Bu toplantıların amacı, projenin gelişim hakkındaki bilgi akışını hızlandırmaktır.





# Retrospective

## Geriye Bakış

Proje çalışanları düzenli aralıklarla geriye bakarak, meydana gelen sorunları gözden geçirirler. Buradaki amaç gelecekte bu sorunların tekrarını önlemektir. Geriye bakış 1 - 6 aylık zaman birimleri için tüm proje çalışanları ya da seçilen bireyler tarafından yapılır. Geriye bakış toplantıları yarım gün - üç gün arasında sürebilir.

okunarak,  
Bununaki  
mekaniz-  
tama-  
ndan  
üç gün




P  
P  
XP  
alın  
işle  
her  
öyl  
kat  
hik  
iste



# ***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.

# Metaphor

## Metafor

- XP projelerinde hazırlanan program için bir veya birden fazla, programın nasıl bir işlevi olacağını ekibin gözünde canlandırmalarını sağlayacak mecazi isim, öge yada resimler kullanılır.
- Örneğin bir shop sistemi yazılımı yapılacak. Burada metafor olarak alışveriş sepeti kullanılabilir. Alışveriş sepetini duyan her programcının aklında, bir shop sisteminin programlanması gerektiği fikri doğar.

# ***Continuous Integration***

## **Sürekli entegrasyon**

- Sistem değişiklikleri ve yeni unsurlar 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.

# ***Collective Ownership***

## **Ortak Sorumluluk**

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

hip

sinin  
rekli  
a gelir.

Sustain

Kalıcı Tempo

XP projelerinde p  
sasiyeri amsadlar.  
yotulan bir program  
beklenmez.

## XP TEKNİKLERİ

XP teknikleri programcılarının XP değer ve prensiplerini uygulamada yardımcı olur.

### Testing

#### Test Ekleme

Test ekleme, geliştirme sürecinde yazılan kodun doğru çalıştığını ve beklenen sonuçları ürettiğini doğrulamaktır. Test ekleme, yazılımın kalitesini artırır ve hataları erken aşamada tespit etmeye yardımcı olur.

### Pair Programming

#### Çift Programlama

• XP metodolojisi, bir programcı ile bir denetçiye dayanır.  
• Bu şekilde programlama, kodun daha kaliteli ve doğru yazılmasını sağlar.  
• Her iki kod yazar da programın işlevini bilir.

### Refactoring

#### Yeniden Yapılandırma

Refactoring, kodun işlevini koruyarak yapısal olarak iyileştirilmesini ifade eder. Bu, kodun daha okunabilir ve sürdürülebilir olmasını sağlar. Refactoring, kodun kalitesini artırır ve hataları erken aşamada tespit etmeye yardımcı olur.

### Simple Design

#### Kolay Tasarım


• Programcıların, bir sorunu çözmede en basit ve en hızlı çözüme ulaşmaya çalışmasıdır.  
• Basit bir tasarım, daha kolay değiştirilebilir ve daha az hata içerir.  
• Bu, kodun kalitesini artırır ve hataları erken aşamada tespit etmeye yardımcı olur.





# ***Testing***

## **Test Etmek**

- Oluşturulan programların kalite kontrolünden geçmesi gerekmektedir.
  - Bu yazılım esnasında oluşturulan testlerle yapılır.
  - Programcılar unsurlar için unit testleri hazırlar.
  - Sınıf bazında yapılan bu testlerle unsurların işlevleri kontrol edilir.
  - Müşteri gereksinimlerini test etmek için akseptans (onay/kabul) testleri hazırlanır.
  - Unsurların entegrasyonunu test etmek için entegrasyon testleri hazırlanır.
- 

# ***Refactoring***

## **Yeniden Yapılandırma**

Tasarım hataları yazılım sisteminin daha ilerde tamir edilemeyecek bir hale dönüşmesine sebep olabilir. Bu yüzden bu hatalar hemen giderilir. Bu yeniden yapılandırma işlemine refactoring ismi verilir.



## ***Simple Design***

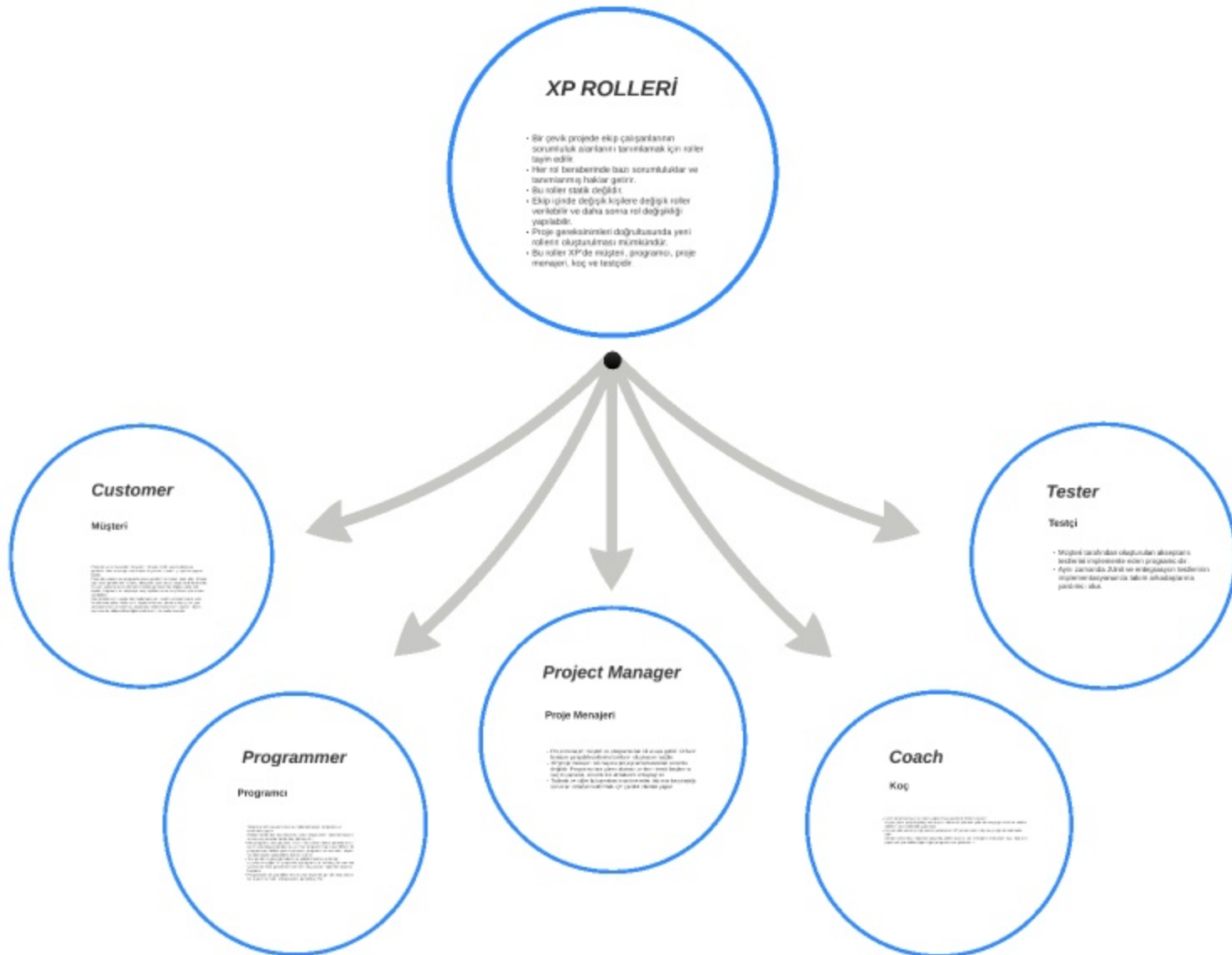
### **Sade Tasarım**

- Programın basit bir yapıda kalması ve ileride değiştirilebilir, geliştirilebilir olması gereklidir.
- Sade bir tasarım daha kolay ve daha hızlı gerçekleştirilebilir. Bu da anlamayı ve anlatmayı daha kolaylaştırır.

# ***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 ROLLERİ***

- 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.
- Bu roller XP'de müşteri, programcı, proje menajeri, koç ve testçidir.

# ***Customer***

## **Müşteri**

- Projenin var olma sebebi müşteridir. Müşterinin ihtiyaç duyduğu ve gereksinimlerine cevap verebilecek bir yazılım sistemi için yatırım yapan kişidir.
- Proje bünyesinde ne programlanması gerektiğini müşteri tayin eder. Müşteri yapılması gerekenleri kullanıcı hikayeleri (user story) oluşturarak ifade eder.
- Müşteri çalışma alanı (domain knowledge) hakkında bilgiye sahip olan kişidir. Programcılar müşteriye karşılaştıkları sorunları çözmek için sorular sorabilirler.
- Hangi kullanıcı hikayelerinin implemente edileceğine müşteri karar verir.
- Implemente edilen kullanıcı hikayelerini kontrol etmek amacıyla müşteri akseptans testleri tanımlar. Akseptans testleri kullanıcı hikayesini doğru implemente edilip edilmediğini kontrol edici bir mekanizmadır.



# ***Programmer***

## **Programcı**

- Sistem analizi, tasarım, test ve implementasyon programcılar tarafından yapılır.
- Müşteri tarafından hazırlanan kullanıcı hikayelerinin implementasyon süresi programcılar tarafından tahmin edilir.
- Her programcı test güdümlü (TDD – Test Driven Development) ve bir takım arkadaşıyla beraber çalışır. Pair programming olarak bilinen, iki programcının birlikte yazılım yapması, 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 her gün bir veya birden fazla şekilde modül entegrasyonu gerçekleştirirler.





# ***Project Manager***

## **Proje Menajeri**

- Proje menajeri müşteri ve programcıları 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.

# ***Coach***

## **Koç**

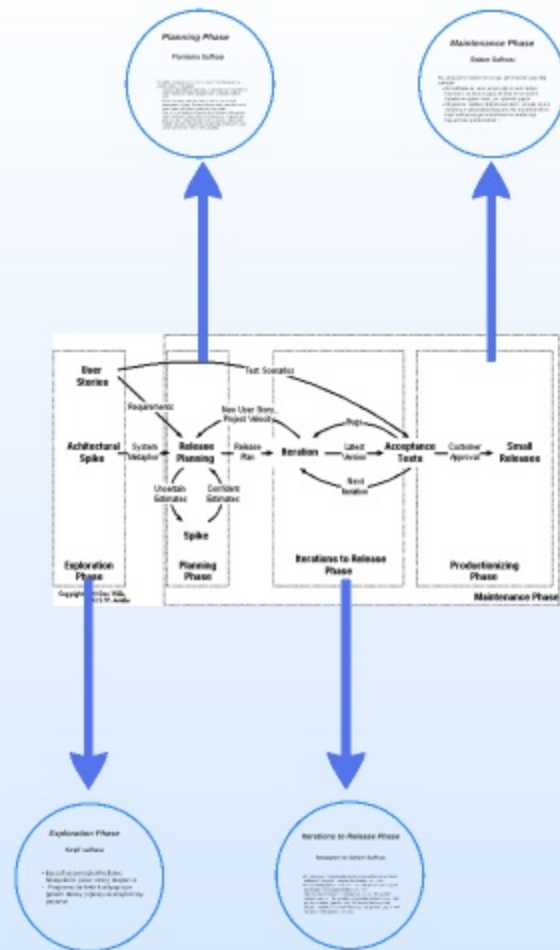
- Çevik süreci tanıyan ve nasıl uygulanması gerektiğini bilen experdir.
- Koçun görevi proje başlangıcında çevik takımı oluşturmak yada bir araya getirmek ve onlara belirli bir süre rehberlik yapmaktır.
- Koç projede sorun çıktığı zaman yada takım XP yöntemlerinin dışına çıktığında müdahale eder.
- Zaman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



# ***Tester***

## **Testçi**

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



## ***Exploration Phase***

### **Keşif safhası**

- Bu safhada müşteri kullanıcı hikayelerini (user story) oluşturur.
- Programcılar teknik altyapı için gerekli deney (spike) ve araştırmayı yaparlar.

# ***Planning Phase***

## **Planlama Safhası**

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 edilemezse, programcılar spike solution olarak bilinen basit bir çözüm implemente ederek, kullanıcı hikayesinin gerçek implementasyonu için gerekli zamanı tahmin etmeye çalışırlar.

## ***Iterations to Release Phase***

### **İterasyon ve Sürüm Safhası**

- Bir iterasyon bünyesinde implemente edilmesi gereken kullanıcı hikayeleri müşteri tarafından belirlenir.
- Implementasyonunun işlevini kontrol etmek için müşteri tarafından akseptans testleri belirlenir.
- Her iterasyon sonunda müşteriye, çalışır bir yazılım sistemi sunulur. Bu şekilde müşterinin sistem hakkındaki görüşleri alınır (geridönüm). Bir önceki iterasyonda oluşan hatalar bir sonraki iterasyonda gözden geçirilmek ve giderilmek üzere planlanır.

# ***Maintenance Phase***

## **Bakım Safhası**

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.