Extreme Programming

Giriş:

**Extreme Programming nedir ?**

-Extreme programming yada XP değişen müşteri ihtiyaçlarına yazılım kalitesi ve yanıt için tasarlanmış bir yazılım geliştirme metodolojisidir, Agile yazılım geliştirme türü olarak bilinir.

Yazılımı bütün olarak parçalama ve geliştirilen yazılımı müşteri gereksinimlerine adapte etme ve verimi yükseltme matığını savunur.

Xp-nin diğer elementlerinden : çift halinde programlama yada diğer değişle Geniş ve kapsamlı kod inceleme , unit testing (test aşamaları), kullanışlı ve kullanıcı için basit bir yönetim yapısı kurma , kodta sadelik ve berraklık sağlama vb. Elementleri kapsayan bir metodolojidir.

**Tarihçe** :

Extreme programming Chrysler Comprehensive Compensation Sysem (C3) Bordro projesinden Kent Beck tarafından Mart 1996-da ortaya cıkarılmıştır ve Ekim 1999-da “Extreme Programming explained” yani Extreme Programming açıklanış kitabını çıkarmıştır.

Amacı : Extreme programming explained bize insanların nasıl daha verimli ve daha kaliteli yazılımlar üretmek için organize yazılım geliştirme disiplinleri ve tekniklerinden bahsetmektedir.

**Xp 4 temel faaliyetlerden oluşmaktadır** :

-Coding (Kodlama)

-Testing (Test)

-Listening (Müşteri isteklerini dinleme)

-Designing(Tasarlama)

**Kodlama** : XP savunucuları , bir sistem geliştirme sürecinin en önemli faktörü kod olduğunu savunmakta , “kodsuz çalışan ürün olmaz” yönünde.

**Testing** : XP yaklaşımı şunu savunur , Eğer az yapılmış bir test birkaç kusuru ortadan kaldırabilirse eğer , br sürü test daha birçok kusuru ortadan kaldırabilir.

-Unit testleri : Kodlanan belirli bir özellik çalışıp çalışmadığını belirler . Bir programcı , kodu bozabileceğini düşündüğü tehditlere karşı bir sürü test uygular , eğer yazılan kod bütün yapılan testlerden gecerse ozaman kod tamamlanmış sayılır. Yazılan kodun her parçası bir sonraki özelliğe geçmeden önce test edilir.

-Acceptance test (Kabul testi) : Kabul testleri programcılar tarafından anlaşıldığı gibi gereksinimler müşterinin gerçek ihtiyaçlarını karşılamaktan emin olur.

**Litening** – Programcılar Müşterinin sistemden ne istediğini dinlemeleri ve anlamalıdırlar. Onlar , müşteriye iyi bir feedback yada geri bildirim vermek için yani sistemdeki bir problemin nasıl düzelebileceği yada düzelemeyeceğini anlatmak için Müşterinin isteklerini çok iyi bir şekilde dinlemeleri gerekmektedir.

-Designing (Tasarım): Basitlik açısından bakıldığında bir sistem kodlama, test , ve dinlemeden daha fazla ihtiyaç duymaz diyebiliriz. Eğer bu aktiviteler doğru bir şekilde uygulanırsa bize herzaman çalışan bir yazılım verir. Uygulamada , bu işimize yaramaz . Uygulamamız uzun bir süre tasarımsız ilerleyebilir , ama sistemimiz yada uygulamamız gittikçe karmaşık hale gelmeye başladığında , başımız ağrımaya başlayabilir . Örnek olarak tasarlanmamış bir mobil uygulama düşünelim , ve bu uygulamayı halka sunduğumuzu . Uygulamayı kullanmak isteyen kullanıcılar , uygulamızın farklı dilleri desteklemeği için kullanmayı redd etmekteler. Baştan iyi bir tasarımımız olmadığı için ve farklı dil desteği vermediğimiz için kullanıcı ve zaman kaybına neden oluyoruz .

Iyı bir tasarım sistem içinde bir çok bağımlılıkları önleyecektir . Bu , sistemin bir parçasını değiştirerek sistemin diğer bölümlerini etkilememesi anlamına gelir .

**Values (Değerler):** Xp içinde birkaç değer barındırmakta bunlar :

-Communication (İletişim)

-Simplicity (Basitlik)

-Feedback (Geri Bildirim)

-Courage (Cesaret)