Yazılım Mühendisliğinde Uygulama Geliştirme Yöntemleri

1-[Test Güdümlü Yazılım Geliştirme(Test Driven Development)](http://yunusemregl.blogspot.com/2012/10/test-gudumlu-yazlm-gelistirmetest.html)

Test güdümlü yazılım geliştirme Extreme Programming süreçlerinde sık olarak kullanılan bir metod dur.

**Süreç Nasıl İşler  ?**

1-)Yapılacak işlem senaryoları belirle

2-)Her senaryo için bir test kodu ekle

 - Burada direk senaryo gereği talep edeceğiniz veriyi, olmasını istediğiniz anlamlı isim ile çağıracak kodu yazıp gerekli parametreleri vermeniz gerekiyor.(Parametreleri farklı testinizi başarıya uğratabilecek alternatifler ile değiştirmeniz fayda sağlayacaktır)

3-)Testi çalıştır.Hatayı al

4-)Sadece hatayı ortadan kaldıracak kadar kodu en yalın haliyle yaz.

5-)Testi çalıştır.Hata yok ise tekrar kodu gözden geçirip en sade hale getir.

**2.  Özellik Güdümlü Geliştirme**(Feature-Driven Programming),

Jeff De Luca ve Peter Coad tarafından 1997’de geliştirilen ve özellik tabanlı gerçekleştirimi esas alan bir modeldir.Süreç beş ana basamaktan oluşur; her bir basamak için çıkış Sartları tanımlanır ve bunlara uyulur. Bunlar:

* Genel sistem modelinin geliştirilmesi,
* Özellik listesinin oluşturulması,
* Özellik güdümlü bir planlama yapılması,
* Özellik güdümlü tasarımın oluşturulması,
* Özellik güdümlü geliştirmenin yapılması.

Gereksinimlerden oluşan özellikler güdümlü çalışır. Sisteme yeni bir özellik kazandırılmadan önce, detaylı bir tasarım çalışması yapılarak bu özelliği kapsayan mimari yapı oluşturulur. Diğer çevik yöntemlerin aksine, kritik sistemler için kullanımı uygundur.

**3- Uçdeğer Programlama (“Extreme Programming – XP” )**

Uçdeğer programlama, Kent Beck tarafından 1999 yılında bir yazılım geliştirme disiplini olarak ortaya çıkarılmıştır. 4 prensip etrafında toplanır ve bunlar Basitlik, İletişim, Geri bildirim, Cesarettir. Bu prensipleri uyguladığımızda kazançlarımız neler olabileceğini aşağıda belirtelim.

* Basitlik prensibinin Katkıları:
  + Basit bir tasarım üzerinde odaklanmak, müşterinin onaylamayacağı ve uzun zaman alacak bir tasarımın yapılması riskini azaltır.
  + Kodu basit tutmak, gereksinimlerdeki değişikliklerin daha kolay yapılmasını sağlar. Doğru ve basit kodu yazan yazılımcı daha sonra iyileştirmeye zaman ayırır. Bu sayede değişecek müşteri isteklerine de zaman ayırır.
* İletişim Prensibinin Katkıları:
  + Yazılım ekibi ile yazılımı kullanacaklar arasında iyi bir iletişimin olması, sorunların erken fark edilmesini ve hızı kesilmeden projenin devam etmesini sağlar.
  + İletişimin gereksinimlerin toplanmasından başlayarak sürecin devam eden tüm adımlarında üst seviyede tutulması müşteri memnuniyetinin sağlanmasında çok önemlidir.
* Geri Bildirim Prensibinin Katkıları:
  + Yazılım geliştirme süreçlerinde en önemli konulardan biri kalitedir. XP projelerinde kalite geribildirim üzerinden sağlanır.
  + Geribildirim, sorular sormak ve cevaplardan öğrenmek demektir. Müşterinin gerçekten ne istediğini anlamak için ona soru sormak gerekir. Ne kadar erken geri bildirim alınırsa, o kadar çabuk çözüm üretilir.
* Cesaret Prensibinin Katkıları:

Gerektiği zaman zor kararlar verebilmeyi savunur. Yazılımcı yeteri kadar cesur olmazsa hata yapma korkusu ile yazılımı müşteriye götürmekten çekinebilir; bu ise geribildirimin geç alınmasına, yapılan hatanın geç anlaşılmasına ve maliyetin artmasına neden olabilir. Ayrıca yazılımcı yaptığı hatayı kabul etme cesaretini gösterebilmelidir.

Kaynakça : <http://www.yazilimheryerde.com/2014/09/agile-metodoloji-scrum.html>

Hazırlayan: Merve Dönmez