ÇIKABİLECEK SORULAR:

Aşağıdaki boşlukları terimler ile doldurunuz?

Aşağıdakilerden hangisi kötü tasarım ve iyi tasarım farkıdır.

Çevik yazılımın mottosu nedir?

Çevik yazılım neyi neye tercih eder?

Çevik yazılım şemşiyesinde neler vardır?

Sprint kaç hafta?

Backlog

Feedback

XP Programming

Çevik ve çağlayan farkları nelerdir?

UML hangi grup tarafından standartlaştırıldı?

Class diyagramı verilen bir programın kodu nasıl yazılır?

Kodu verilen bir programın UML diyagramı nasıl çizilir?

Class diagram associotion türleri bilinsin?

**Hafta-2.1:**

TASARIM: Herhangi bir mühendislik ürünü geliştirme sürecindeki ilk adım.

YAZILIM TASARIMI: Bir sorunun yazılım çözümü için sorun giderme ve planlama şamasıdır.

YAZILIM TASARIM HEDEFLERİ:

* Gereksinimleri karşılamalıdır.
* Ürünün geniş çeşitliliğini tahmin etmek.

İyi bir yazılım mimari ve tasarımcısı uygulama genel bir düzeyde nasıl çalışacağını gösteren net bir zihinsel model oluşturur , daha sonra bu zihinsel modelle uygun bir ayrışma geliştirir.

İYİ TASARIM:

* Kullanılabilirlik
* Güvenilirlik
* Genişletebilirlik
* Bakım Kolaylığı
* Uyumluluk
* Yeniden Kullanılabilirlik
* Pazarlanabilirlik
* Test edilebilirlik
* Taşınabilirlik

KÖTÜ TASARIM:

* Esnemezler, sistem değişikliğe izin vermez.
* Kırılgandır, değişiklik yapıldığında çöker. (Katasrofi)
* Yeniden kullanılamazlar, taşınabilir değillerdir.

UML(Unified Modeling Language)

* Programlama dili değildir
* Evrensel modelleme biçimi ve dilidir.

FAYDALARI:

* Tasarım ve analiz yapılırsa kodlama kek kalır.
* Hatalar minimalize edilir.
* Tekrar kullanılabilir kod sayısı epey artar.
* Bellek kullanımı daha verimli olur
* Programın kararlılığı artar()- neyi nasıl yapacağın belli (Senayolar)
* Takım çalışması için harika yardımcıdır. (Herkes genel şablon biliyor)

**Hafta-2.2:**

ÇEVİK YAZILIM ŞEMSİYESİ:

* Scrum
* Kanban
* XP
* DSDM
* FDD
* Crystal

TEMELL DEĞER:

* Süreç ve araç ► Bireyler ve etkileşim
* Kapsamlı dokümantasyon ► Çalışan yazılım
* Sözleşmeli pazarlıklardan ► Müşteri ile iş birliği
* Plana bağlı kalmak ► Değişime karşılık vermek

FEEDBACK: Geri dönüşler.

GELENEKSEL VS ÇEVİK:

* Müşteri ne istediğini bilir ► Keşfeder
* Geliştiriciler ne yapacağını bilir ► keşfeder
* Değişim olmaz ► ayak uydurulur

SPRİNT: 3-5 Hafta

**Hafta-3:**

UML AVANTAJLARI:

* Biçimsel bir dildir
* Kısa bir dildir
* Anlaşılır bir dildir
* Ölçeklenebilir bir dildir
* Tecrübelerden oluşmuştur
* Bir standarttır (Object Management Group)

SINIF DİYAGRAMI: (CLASS DIAGRAM):

* Nesne, öznitelik, ilişkiler ve işlemleri göstermek üzere kullanılır.
* Public + , private - , protected #, package ~
* Constraints alt sekmesi de bulunabilir.
* Sınıf Adı, Öznitelikler, Methodlar ve Kısıtlar bulunur.
* Öznitelik standardı : -name:string , +yariCapi: Double , +salary: int=300
* Metod Standardi : +yüktasirMi(yük:int):bool

ASSOCİATİON:

* Dört tip;1….n, n…….n,1……1
* Direk uzun çizgi ile gösterilir. Tek yönlü ise sadece bir sınıflık durum vardır.
* -------------------------------------
* Sınıflar arasında HAS-A ilişkisi kurulur.

DEPENDENCY(Bağımlılık):

* Eğer bir sınıf diğer sınıf metotlarından en az birinde parametre olarak kullanılıyorsa iki sınıf arasında bağımlılık ilişkisi vardır.
* Kesikli ok sonu açık, associtioton kesikli hali.

Flyweight Design pattern (Nesne sayısını azaltmak)   
Adapter Design pattern (Uyumsuz arayüzleri uyumlu kılmak)   
Composite Design pattern (Bütün parça ilişkisi kurgulamak)  
Proxy Design pattern (Bir nesneye erişimi kontrol etmek)   
Decorator Design pattern (Nesneye dinamik olarak yeni özellikler kazandırmak)   
Bridge Design pattern (Soyutlama ile gerçekleştirerek resmi birbirinden ayırmak)