https://www.sonarqube.org/

​iterative incremental

unified process

------------------**İsimlendirme Kuralları**-----------------------

Değişken adını açık açık yazmalısınız

Google Araması:C# Naming Conventions

ingilizce kullanılmalı çünkü programlara dilleri ingilizcedir

------------------Fonksiyonlar-----------------------

Single Responsibility Principle(SOLID),Refactoring Tools

Google Araması: Single Responsibility Principle,Code Refactoring

Parametreler!!!!!!!!!!

->Encapsulation:

Olabildiğince tek bir nesneye indirgenmeli,

Model,Dto,Complex type gibi ortak nesnelere konulmalıdır

------------------Yorumlar-Comments-----------------------

Kurumsal hafıza için yapılıyor ama yanlış yapılıyor

Software Development Methodology(Agile),ALM(Application Lifecycle Management)

Google Araması:SCRUM with TFS

Fonksiyonun içine Comments Yazılmaz

Fonksiyonun üzerine yazılmalı

------------------Soyutlama-----------------------

Çıplak class kalmasın

Class interface abstract veya baseden yararlanmıyorsa sıkıntı büyük

------------------SOLID-----------------------

Single Responsibility:Her sınıfın veya metodun tek bir sorumluluğu olmalı,Bir tek şeyi yap ve onu en iyi yap

Open Closed:Sınıflar değişikliğe kapalı ancak gelişime açık olmalıdır

Liskov ‘s Substitution:Alt sınıflar miras aldığı üst sınıfın bütün özelliklerini kullanmalı, alt sınıflarda oluşturulan nesneler üst sınıfların nesneleriyle yer değiştirdiklerinde aynı davranışı göstermeli ve herhangi bir kullanılmayan özellik olmamalı.

Interface Segregation:xx

Dependency Inversion:Katmanlı mimarilerde üst seviye modüller alt seviyedeki modüllere doğruda bağımlı olmamalıdır.

------------------Aspect Oriented Programming-----------------------

IoC Container

Mümkün olduğunca business classlarında az if kullan ve bunları AOP ile yap daha büyük avantaj ve daha temiz kod