**COMMON CLOSURE PRENSİPLERİ**

Yazılımcı olarak bizlerden bir değişiklik talep edildiğinde ilk olarak hangi bileşenleri etkilediğini düşünmeye çalışırız. Değişiklik yapmamız gereken bileşen sayısının minimum olması ise her zaman işimizi kolaylaştırır. Tek bir bileşende değişiklik yapmamız gerekiyorsa işimiz bir anda kolaylaşır. Çünkü tek bir bileşeni değiştirmek, test etmek ve yaygınlaştırmak her zaman daha kolaydır. İşte yazılımcılardaki değişimin dar bir alanda kalması isteğinin prensibe dönüşmüş hali Common Closure Prensipleri’dir. Dolayısı ile daha bileşenler tasarlanırken değişimler öngörülerek aynı sebeplerle değişecek sınıfların aynı bileşenlere yerleştirilmesi işlerimizi ciddi anlamda kolaylaştıracaktır.

Özetlemek gerekirse bir değişiklik minimum sayıda bileşeni etkilemelidir. Dolayısıyla sınıflarımızın bileşenlere dağılımları değişiklik talepleri de göz önünde bulundurularak yapılmalıdır.

SCRUM

[Yazılım Mühendisliği](https://tr.wikipedia.org/wiki/Yaz%C4%B1l%C4%B1m_M%C3%BChendisli%C4%9Fi)'nde bir uygulama geliştirme yöntemidir. Bu geliştirme yönteminin temel özelliği gözlemci, geliştirmeci ve [tekrara dayalı](https://tr.wikipedia.org/wiki/%C4%B0terasyon) olmasıdır. Birçok modern yazılım projesinin oldukça karmaşık olduğu ve en baştan tümünü planlamanın zor olacağı şeklindeki bir varsayımdan hareket eder. Bu karmaşıklığı üç ilke ile azaltmaya çalışır.

Şeffaflık: Projedeki ilerlemeler ve sorunlar günlük olarak tutulur ve herkes tarafından izlenebilir olması sağlanır.

Denetleme: Ürünün parçaları ya da fonksiyonları düzenli aralıklarla teslim edilir ve değerlendirilir.

Uyarlama: Ürün için gereksinimler en baştan bir defalığına belirlenmez, bilakis her teslimat tekrar değerlendirilir ve duruma göre uyarlamalar yapılır.

Amaç başlangıçta hayal edilen bir ürünün, hızlı, ucuz ve kaliteli şekilde üretilmesidir. Tasarlanan ürünün gerçekleştirilmesi, müşteri tarafından mümkün olduğunca detaylı şekilde hazırlanmış bir [talepler listesinin](https://tr.wikipedia.org/w/index.php?title=Talepler_listesi&action=edit&redlink=1) aşama aşama gerçekleştirilmesi biçiminde yapılmaz. Bunun yerine müşteri tarafından istenilen ve tanımlanan işlevler, iki ya da dört haftalık "[Sprint](https://tr.wikipedia.org/w/index.php?title=Sprint_(yaz%C4%B1l%C4%B1m&action=edit&redlink=1)" adı verilen dönemler içerisinde geliştirilir ve yeniden gözden geçirilir. Bu kullanıcı bazlı gereksinim tanımı [Kullanıcı Hikayesi](https://tr.wikipedia.org/w/index.php?title=User_Story&action=edit&redlink=1) olarak nitelenir ve özellikler defterinde yer alır. Her Sprint sonunda yazılımın fonksiyonel bir parçası bitmiş ve müşteriye teslim edilebilir bir durumda olur. Scrum [Çevik yazılım geliştirme](https://tr.wikipedia.org/wiki/Atik_yaz%C4%B1l%C4%B1m_geli%C5%9Ftirme) prensiplerini hayata geçiren bir yöntemdir.