**Consequences of Dependency**

* Dependency causes complexity.
* Dependency necessiates understanding and changing together.
* Dependency prevents understanding and changing dependent object in isolation.
* That’s because of the fact that the semantics of the dependent object is incomplete without the depended object.
* And a change to a depended object may cause changes to depending ones.

Dependency birlikte anlama, birlikte degisme problemi ortaya koyar. Yani clienti anlamak icin service anlamak zorundadır. Clientı service olmadan anlayamayız. Tersine service degisirse client’ında degisme ihtimali var, degisimi de izole edemiyoruz. Anlamayı ve degisimi izole edemediğimiz noktada yazılımın en temel problemi olarak ortaya cıkıyor.(**dependency**)

**Dependency is Unavoidable**

* On the other hand dependency is a must; to be part of a system objects must depend on each other in numerous ways:
* Some objects must be part of other objects,
* Some objects must create others,
* Objects must call methods on each other by passing some other objects,
* Some objects must throw exceptions while others must catch them.
* etc.

**Dependency kacınılmaz dolayısıyla bunu nasıl kuracağımız önemli bir faktör.**

**Mesele dependencyleri sağlıklı ve düzgün bir sekilde kurmak ve yönetmek. Design patterns lar dependecy’nin bilimsel tarafı diyebiliriz**.