UML, günümüzden yazılım projeleri denildi mi akla gelen ilk şey. 1997 yılından beri kullanımda olan UML’in popülaritesini arttırmasındaki en büyük etkenlerden biri her türlü senaryoya dayalı 13 farklı türünün bulunmasıdır. Ayrıca bazı eklentiler vasıtasıyla kodlar direkt UML’e veya UML direkt koda dönüştürülebilmektedir.

UML tasarımları müşteri ile geliştirici arasında bir köprü görevi üstlenmektedir. Öyle ki, müşteri isteklerini belirtir. Tasarımcı, koda girmeden UML tasarımı ile istekleri görselleştirir. Böylelikle, geliştiricinin kodlaması daha kolay hale gelecektir.

Peki UML tasarımlarının bir bitiş tarihi var mıdır? Yoksa bir projenin başından sonuna kadar tasarım yapılabilir mi?

Yapılan bir araştırmaya göre UML tasarlayanların %52,5’i, bitiş tarihinin tasarımın tamamlanma tarihi olarak belirtmişlerdir. Lakin tamamlanma konusunda herhangi bir kriter söz konusu değildir. Tamamen kişiye ve geliştiriciye kalmış bir durumdur. Katılanların %32,8’i teslim tarihinin bitiş tarihi olduğunu vurgulamışlardır.

Başka bir araştırmaya göre UML tasarımlarının vakti gelmeden durmasının sebebi olarak büyük bir çoğunluk, “iletişimsizlik” olarak belirtmiştir. İletişimsizlik, büyük projelerde vakit ilerledikçe projenin karmaşıklığından doğan sorunların paydaşlarla iletişim halinde olup çözülememesi ve projenin işin içinden çıkılamaz bir hal alması durumu. Bu durum sadece UML tasarımlarını değil, aynı zamanda projeyi de kötü etkilemektedir.

Aslında bu araştırma da gösteriyor ki UML dünyada çokça kullanılan bir tasarım kalıbı olsa da farklı kullanıcılar tarafından farklı yorumlanabilmektedir.

UML tasarımında ortaya çıkan sorunları şöyle sıralacayak olursak;

* Dağınık Bilgi: Bir projeye başladığınızda vakit geçtikçe işin dallanıp budaklandığını, işin karmaşıklaştığını göreceksiniz. Bu açında bilgilerin arasındaki bağlantıyı görmek ve ilişkili olanları birleştirmek dağınıklığı ortadan kaldıracaktır.
* Eksik Bilgi: Her proje zaman geçtikçe paydaşların isteklerine göre şekillenmektedir lakin bu istekler projenin en başında belirtilmez ise projenin beklenmedik bir ürüne dönüşebilir.
* Tutarsız Olmak: Her büyük proje bölümlere ayrılıp farklı ekipler tarafından geliştirikmektedir. Bu durum ekiplerin düşünce tarzlarına ve kodlama biçimlerine göre karmaşıklığa sebep olmaktadır.

Bu 3 durum bir yazılım projesinde kodlamadan daha fazla sıkıntıya neden olmaktadır. Çünkü kodlama kısmı dünyaca kabul gören kriterler, kurallara sahiptir. İstesenizde bunun dışına çıkmanız mümkün değildir. Lakin UML’de belli bir tasarım kriteri olmadığı için bir tasarımcını tasarladığı UML, farklı bir tasarımcı tarafından anlaşılamamaktadır. Buna karşın, her UML diyagramı farklı bir amaca hizmet etmektedir. O amaç doğrultusunda her UML’in küçük de olsa bir kriteri mevcuttur. Bu kriterlere rağmen tasarımlarda kusurlar, eksiklikler mevcuttur. Bu kusurlar, tasarımın farklı mimarlar tarafından farklı algılanmasına hatta projenin farklı kodlanmasına bile sebep olmaktadır. Özetle, bu tür kusurlar, eksiklikler projeyi kötü etkileyeceğinden her aşamada test edilerek belirlenmeli ve önlenmeye çalışılmalıdır.