

# Yazılım Projelerinin Temel Adımları ve Metodoloji İlişkisi

Yazılım projelerinin temel adımları aşağıdaki gibidir:

1. **Planlama:** Yazılım yaşam döngüsünün başlangıç evresidir. Projeyi sipariş eden veya projenin hedeflediği kullanıcıların istekleri ve ihtiyaçları dikkate alınarak proje ihtiyaçları tespit edilir. Proje boyunca takip edilecek olan yol, projede kullanılacak olan donanımların maliyeti, proje sürecinde personeller arasında iş bölümünün nasıl olacağı ve proje adımlarının zamana nasıl yayılacağı gibi detaylar değerlendirilir.
2. **Analiz:** Geliştirilecek olan üründen tam olarak ne istendiği net bir şekilde kağıda dökülür. Proje somutlaşmaya başlamıştır.
3. **Tasarım:** Yazılım sisteminin temel yapısı bu aşamada oluşturulur. Yazılım ürününün mantıksal ve fiziksel tasarımı üzerine çalışmalar yapılır.
4. **Geliştirme:** Temelleri oluşturulmuş olan ürünün tam olarak kodlanması, kurulması ve test edilmesi bu aşamada gerçekleştirilir.
5. **Teslim ve Bakım:** Test aşamalarının tamamlanması; ürünün hazır olduğu ve müşteriye teslim edilebileceği anlamına gelir. Müşterinin yazılımı aktif olarak kullanması ve kullanım sonucu ihtiyaç duyulan bakımların yapılması bu aşamada gerçekleşir.

Başarılı bir yazılım geliştirme süreci geçirmek istiyorsanız, yazılım projenizin her aşamasını detaylı olarak planlamalı ve bu planı sağlıklı bir şekilde uygulamalısınız. Peki projeniz için mükemmel planı nasıl oluşturacaksınız? İşte yazılım metodolojileri tam da bu noktada devreye giriyor ve bir yazılım projesini oluşturan temel adımları nasıl planlamanız ve yönetmeniz gerektiği konusunda hazır şablonlar sunuyor.

## Waterfall Metodolojisi

Waterfall yazılım geliştirme modeli, Türkçe kaynaklarda şelale veya çağlayan modeli olarak da karşınıza çıkabilir. Waterfall metodolojisi, klasik yazılım geliştirme yaklaşımı olarak nitelendirilir. Waterfall yazılım metodolojisinin temel felsefesi, proje sürecindeki bir aşamayı tamamlamadan diğer aşamaya geçmemek üzerine kuruludur. Örneğin; analiz aşaması kesin olarak tamamlanmadan tasarım veya geliştirme aşamasına geçilmemelidir vb gibi...

Bu yaklaşımda; bir aşamanın hiçbir eksik olmadan tamamlanması ve bir sonraki aşamaya geçilmesi, daha sonra ise hiçbir şekilde önceki aşamalara geçilmemesi tavsiye edilir. Yani tamamlanmış olan her aşamanın mükemmel yakın performans göstermesi beklenir ki çoğu yazılım projesinde böyle bir ihtimalden söz etmek mümkün değildir.

Waterfall metodolojisinin en büyük avantajı; bu metodolojiyi anlama, uygulama ve yönetmenin çok basit olmasıdır. Proje safhaları arasında sadece ileri yönlü geçiş olacağı için çalışanlar arasındaki iş bölümü projenin en başından itibaren net sınırlarla bellidir. Diğer bir yandan bu metodolojinin yüksek riskli, yüksek maliyetli ve düşük verimli olduğunu söylemek mümkün.

## Çevik Yazılım Geliştirme Metodolojisi

Çevik çalışma modeli veya agile çalışma olarak da bilinen çevik yazılım geliştirme metodolojisi, waterfall metodolojisinin eksiklerini gidermek ve proje sürecine yeni bir yaklaşım kazandırmak için geliştirilmiştir. 2001 yılında Amerika'da yayınlanan agile manifestosu, bu metodolojinin prensiplerini ortaya koymuştur.

Bu yaklaşımın temel felsefesinde; denetim, adaptasyon ve gelişim gibi kavramlar büyük rol oynar. Agile metodolojisinde ürünü bir kere yapıp bitirmekten ziyade, ürünü en basit haliyle kullanıcıya sunmak ve kullanıcıdan gelecek feedbackler doğrultusunda ürünü sürekli olarak geliştirme amacı vardır. Örneğin Microsoft Word benzeri bir yazılım yapacağımızı düşünelim. Agile metodolojisi, öncelikle beyaz bir ekrana klavyeden girilen harflerin yazılmasını sağlayan bir yazılım üretmemizi tembihler. Ürünün ilk halini kullanıcıya sunduktan sonra; kullanıcının ihtiyaçları doğrultusunda harfleri renklendirme, başlık oluşturma, sayfaya resim ekleme, listeleme şablonları kullanabilme vb gibi yeni özellikler kodlanarak ürün adım adım tamamlanmış olur.

Çevik metodolojinin en büyük avantajı, geliştirme süreçleri kısa olduğu için ekip motivasyonunun daima üst seviyede kalmasıdır. Proje ekibinde bıkkınlık ve tükenmişlik sendromu gözlenmeyecek, kullanıcıyla olan etkili iletişim ekibi canlı tutacaktır. Bu metodolojinin en büyük dezavantajı ise sürekli değişen kullanıcı ihtiyaçları dolayısıyla aşırı çalışma olarak gösterilebilir.

## Scrum Metodolojisi

Scrum metodolojisi, çevik metodolojinin bir alt başlığıdır. Şelale modeline göre çok daha az risk içermesiyle dikkat çeken scrum metodolojisi, ürünün test edilmiş olan birbirinden bağımsız en küçük özelliklerine odaklanır. Böylelikle herhangi bir özelliğin geliştirme aşamasında ortaya çıkan problemler, diğer özellikleri etkilemeyecek ve geliştirme çalışmaları hız kesmeden devam edecektir.

## Lean Metodolojisi

Yalın model olarak da bilinen lean metodolojisi, çevik metodolojiler ailesinin bir diğer üyesidir. Scrum modelde olduğu gibi değer yaratan küçük özelliklere ayrı ayrı odaklanmayı

gerektiren lean metodolojisi, zaman ve emek israfını engellemek adına geliştirilmiştir. Lean modelinde elzem olmayan hiçbir özellik geliştirilmez.