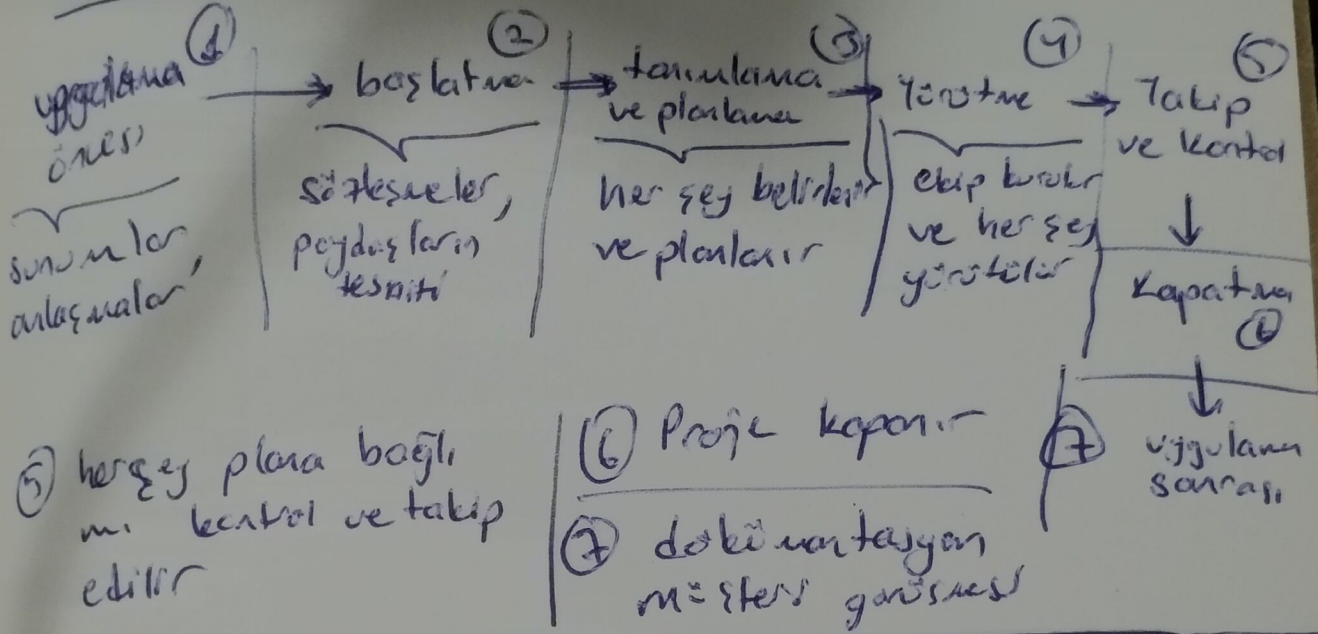


Bilgi Sistemleri Proje Bölümü

Proje Yaşam Döngüsü



Proje Yöneticisinin Sahip Olması Gereken En Önemli Özellikler

- Metakere yeteneği
- Etkileme yeteneği
- İletişim yeteneği
- Problem Çözme yeteneği

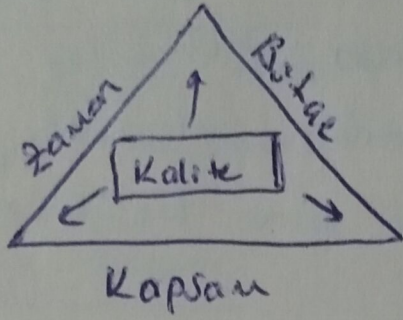
Proje Yöneticisinin Görevleri

- Kapsam tanımlama ve planlama
- Etkinliklerin planlanması
- Kaynak planlama
- Zaman çizelgesi oluşturmak
- Süre tahmini
- Maliyet tahmini
- Bütçe hazırlama
- Dokümantasyon

- Takım Liderliği
- İşleyiş raporlama
- Risk ve Sorun yönetimi
- Kalite Kontrol

Yazılım Proje Yönetim (FARKLARI)

- Ortada fiziksel ön yoktur
- Yazılım geliştirme süreci standart değildir
- Güçlü proje bir defalıktır, sonra teknoloji gelişir



Tüm Projelerin Genel Hedefi
Belirtilen kapsam, süreler, kaynaklarla, istenen süreçte ve kalitede tamamlanmak.

Yazılım Geliştirme Nedir?

Kullanıcıların istediklerini bilgisayar programı haline getirme sürecidir.

Proje Yeni bir ürün, hizmet veya sonuç elde etmek amacıyla belirli süreçte gerçekleştirilen benzersiz çalışmadır.

Yazılım Geliştirme Temel İlkeleri

- Basitlik: Basit olan şey daha kolay ve maliyetsiz yapılır.
- Tekrar kullanılabilirlik: Denge ve maliyet için önemlidir.
- Süreklilik: Sistem değişen koşullarda ayakta kalmalıdır.
- İzlenebilirlik: Proje planında kontrol noktaları olmalı.
- Güvenlik: Veri ve süreç güvenliği olarak değerlendirilir.
 - Müşteri bilgileri
 - proje detayları

1-Yazılım Projesi Başlangıç Noktası

Yazılımla giderilebilecek bir ihtiyac ortaya çıkmasıyla başlar. Kapsam belirlenir, planlama yapılır.

2-Planlama

Proje hedeflerine ulaşmak için; belirlenmiş süre ve mali kısıtlarını dikkate alarak, izlenecek yollar, yapılacaklar, kaynaklar ve süre belirlenmektedir.

3-İhtiyaç Analizi

Paydaş ihtiyaç ve beklentilerinin ayrıntılı şekilde tanımlanmasıdır.

4-Prototip Geliştirme

Prototip veya sorun sorunlarına (deneyel fonksiyon ve arayüzler içeren), sistemi açıklama ve geribildirim alma amacıyla paydaşlara, müşterilere sunulması.

5-Tasarım

İhtiyaç analizinde ihtiyaç çıkarılır, tasarımda çözüm için VUL şeması gibi yöntemlerle yazılımın üst seviye models hatırlanır.

6-Gerçekleştirme

Tasarımda tasarlanan model, yazılım geliştirme araçlarıyla yazılıma dönüştürülür. Kod yazılır, kütüphaneler oluşturulur, veritabanı kurulur vs.

7-Test

Oran: istenilenler doğru anlaşılıyor mı?

Doğrulama: Yapılanlar doğru çalışıyor mu?

8-Gözetim Geçirme

Testten farklı olarak kaliteye bakılır. İleride oluşabilecek hataları aşmaya yöneliktir.

9-Devreye Alma

Test bitince kullanıcı eğitilir, pilot kullanıma ve nihai proje kullanımı yapılarak proje devreye alınır. Sonrasında destek süreci başlar, destek anlaşmasına göre yeni işlemler gerçekleştirilir.

3. Slayt

Yazılım geliştirme planlamadan devreye almaya uzanan bir süreçtir. Kullanıcı ihtiyaçlarının anlaşılması, yapılacakların tasarlanması, kodlama, test, devreye alma ve bakım temel aşamalardır.

İhtiyaçların değişkenliği yazılım geliştirme süreciyle ilgili farklı modeller ortaya çıkmasına sebep olmuştur. Yazılım sürecini yönetmek için

- kod eksenli,
- doğrusal,
- yinelemeli,
- çevik,
- modüler
- servis tabanlı mimariler geliştirilmiştir.

Projedeki ihtiyaçlara göre bu mimariler tek başına ya da birlikte kullanılabilir.

Planlama ve analiz aşamalarının öneminin tam olarak anlaşılmaması, kodla ve düzelt veya kurlsız geliştirmeye yol açmaktadır. Günümüzün karmaşık iş ortamında bu yöntem kullanılarak orta ve büyük ölçekli yazılımlar geliştirilmesi mümkün görülmemektedir.

Şelale modeli, analizin başlangıçta tam yapıldığını kabul ederek yazılımı sürekli ileri doğru geliştirmeye dayanır. V-model, şelale modelinin test aşamaları ayrıntılı planlanmış şekli olarak kabul edilebilir. Her iki modelin de süreçte ortaya çıkan değişiklik taleplerine uyum gösterme özelliği zayıftır. Bu modeller projenin ilk aşamasında yapılacakların çok ayrıntılı bir şekilde planlanmasını gerektirir.

Yinelemeli geliştirme, yazılımın sürümler şeklinde adım adım olgunlaşarak geliştirilmesine dayanır. Artımlı, evrimsel ve sarmal model gibi alt modelleri vardır. Artımlı geliştirme, analizi tamamlanan yazılımın sürümlere bölünerek geliştirilmesidir. Evrimsel geliştirmede yazılımın analizi her sürümde tekrar düzenlenir. Sarmal model artımlı geliştirme ve şelale modelinin üstünlüklerini birleştiren risk yönetimine dayalı bir modeldir. Bir yinelemeli geliştirme modeli kullanılacaksa proje planı sürümleri ve sürümlerdeki özellikleri içerecek şekilde yapılmalıdır. Her yeni sürümde plan gözden geçirilerek daha ayrıntılı hale getirilir.

Çevik geliştirme, müşteri ihtiyaçlarının sürekli değiştiği küçük ve orta ölçekli projelere daha uygun, resmiyeti ve belgeleme ihtiyacı az bir yöntemdir. Kısa bir planlamadan sonra müşterinin de doğrudan dahil olduğu bir ekiple proje geliştirilmeye başlanır. Sürekli yeni sürümler çıkartılarak proje nihai hale getirilir.

Modüler geliştirmede, proje temel özellikler etrafından şekillenen modüllere bölünerek karmaşıklığı azaltmak hedeflenir. Modül bazında planlama yapılarak izlenir. Modüler geliştirme, diğer yöntemlerle birlikte kullanılabilir.

Servis tabanlı mimari platform bağımsız çağrılabilen web servislerine dayanan, web servisleri arası koordinasyon için standart geliştirme araçları öneren, yeni ve gelecek vaat eden bir teknolojidir. Bu teknoloji, analizden tasarıma kadar tüm aşamaların bütünleşik ve standart görsel bir ortamda modellenmesine imkan sağlar. Ancak henüz geliştirme araçlarının kullanımı oldukça karmaşıktır.

4. Slayt

Büyük projeleri yapmak için kişiler ekip olarak birlikte çalışmak zorundadır. Yönetim, ekibin çalışmalarını amaç bütünlüğü içerisinde yürütme, ekip için iletişimi sağlama, ekibi yönlendirme ve yapılan çalışmaların amaçlarına uygunluğunu değerlendirme faaliyetleridir. İnsanlık tarihi boyunca toplumsal ve ekonomik gelişmeler farklı yönetim modeller ortaya çıkarmıştır. Otoriteye dayalı, sanayi otomasyona yönelik, sistem ve kişisel performans odaklı yönetim tarzları gelişmiştir. Günümüzün sürükleyici sektörü yazılım ve bilişimdir. Yazılım insan zihni faaliyetlerinin bir ürünüdür. Bu yüzden yazılım geliştiricilerin bilgi ve becerilerini dikkate alan, yetki paylaşımına ve yeni fikirlere açık esnek bir yönetim tarzı izlenmelidir.

Örgütlenme biçimi, şirketin hedefine ulaşmak için kullandığı yönetim tarzı, görev tanımı, görevler arası ilişkiler, ekip yapısı, birimler ve birimler arası hiyerarşiden oluşur. Şirketler fonksiyon, ürün proje veya matris türü örgütlenebilir. Proje ve kurum yapısı, ortak veya dağınık mekanlarda çalışma örgütlenme yapısı için belirleyicidir. Yazılım projelerinde farklı uzmanlıklardan faydalanmak için birimler arası çalışmalar genellikle matris şeklinde yürütülür.

Proje yöneticisi kullanıcı ihtiyacı, üst yönetim beklentisi ve yazılım ekibi tecrübesi arasında köprü vazifesi yapar. Projedeki önemli kararları verir, planı oluşturur, kalite standartlarını belirler, değişimi yönetir, bilgilendirme ve iletişim faaliyetlerini yürütür, ekibin moral ve motivasyonunu sağlar. Proje yöneticisi projelerin müşteri ve son kullanıcı için yapıldığını ekibe hatırlatmalı, müşteriye bir adım yakın durmalıdır. Kurum üst yönetimi de özel durumlar hariç proje yöneticisinin kararlarını desteklemelidir. Teknik kararlar konunun uzmanı kişiler tarafından alınmalıdır.

5. Slayt

Ekip olarak çalışabilmek büyük yazılım projelerinin vazgeçilmez şartıdır. İyi bir ekip, ortak bir hedef etrafında kenetlenmiş kişilerden müteşekkildir. Başarılı bir proje için mümkün olan en tecrübeli kişiler istihdam edilerek uyumlu bir ekip oluşturulmalıdır.

Yazılım projeleri, birbiriyle ilişkili birçok farklı görevin koordineli bir şekilde yapılmasıyla gerçekleştirilir. Bu görevler belli unvanlarla standartlaşmıştır. Kodlama, analist, tasarım, test gibi alanlarda uzmanlıklar gelişmiştir. Bu uzmanlıklar yazılım geliştirme süreci içinde anlam kazanır. Ekibin her elemanı süreçteki yerini bilmelidir.

Proje yöneticisi ekibin birlikte çalışması, iletişim, koordinasyon, moral ve motivasyonunun artmasında aktif rol oynamalıdır. Ekibi tanımalı, çalışanların sürekli bölünmesini mümkün olduğunca engellemeli ve çalışanlara özel bir iş yapma duygusunu hissettirmelidir. Ekip hedefleri ve kişisel kariyer hedefleri arasında bir denge kurulmalıdır.

Kaliteli yazılım, kaliteli ve birbirine uyumlu çalışan yazılımcılarla gerçekleştirilir. Proje yöneticisi, bu konuda yapacağı önerilerle uygun bir ekibin oluşturulmasında öncülük eder. Proje için yeterli sayı ve tecrübede eleman ayrılmalıdır. Ekip sayısı ve tecrübesi, proje süresini olumlu ya da olumsuz etkiler. Sayı artışı olumludur ancak belli bir noktadan sonra iletişim ihtiyacını çok fazla arttırarak proje süresinin uzamasına sebep olabilir. Ekip oluşturduktan sonra çalışmalar sürekli değerlendirilmeli ve gerekirse ekip takviye edilmelidir.

6. Slayt

Planlama proje kısıtlarını dikkate alarak, eldeki kaynaklarla proje hedefine ulaşmak için izlenecek yöntem, kullanılacak kaynaklar, yapılacak görevler ve görev takvimlerinin belirlenmesidir. Proje planının sağlıklı olması proje başarısı için kesin şarttır.

Proje planı, projenin kapsamı belirlendikten sonra genel olarak hazırlanır. Proje ilerlerken ihtiyaç ve yapılacaklar netleştikçe plan daha ayrıntılı hale gelir. Planın tekrar gözden geçirilmesi periyodik olarak, proje aşaması veya sürüm geçişinde yapılabilir. Projenin başlarında analiz ve planlama için harcanan çaba ilerleyen aşamalarda projenin sağlıklı ilerlemesini sağlar, geri dönüş ve hataları azaltır. Küçük projelerde planlama için harcanan emek az olabilir. Ancak proje büyüdükçe bu ihtiyaç artar.

Planlanması gereken konular standartlaşarak çeşitli plan türleri ortaya çıkmıştır. Temel plan türleri kapsam, bütünleşme, risk, zaman, insan, kaynak, maliyet, teknik altyapı, gözden geçirme, test, eğitim ve kurulum şeklinde sayılabilir. Geliştirilecek yazılımının yapısına ve alanına göre farklı türde planların yapılması da gerekebilir.

Karmaşık yazılım ürünlerinin geliştirmek için yürütülen süreçler, yapılacak çalışmalar alt bölümlere ayrılarak daha anlaşılır ve yönetilebilir hale gelir. Planın da bu şekilde ayrıntılı hale getirilmesinde iş bölümlene yapısı kullanılır. Görevlerin birbiriyle ilişkilerini göstermek için Gantt Şeması ve Ağ Şeması gibi yöntemler önerilir. İşlerin bölümlenmesi üzerinden projenin bitiş süresini belirleyen kritik yol, bolluk süresi, en erken ve en geç görev bitiş tarihleri hesaplanabilir.

Kaynak, işin yapılmasını sağlayan kişi veya araçtır. Planın yürütülmesi için gerekli roller belirlenmeli ve görevlerle eşleştirilmelidir. Daha sonra role uygun kişiler istihdam edilerek göreve atanır. Bunun sonucunda bazı kaynaklar aşırı yüklenmiş olabilir. Kaynakların aşırı yükleri, kaynak dengeleme yöntemleriyle tekrar dağıtılır.

Maliyet planı projenin yürütülmesi için gerekli maliyetler, bütçeleme, harcama takvimi ve harcama kontrol işlemlerinin belirlenmesinden oluşur.

Yazılımların ve yapılacak çalışmaların ortak özelliklerinden yola çıkılarak her planda bulunacak konular önerilebilir. Bir kurum proje, birim veya yatırım alanına göre kendi standart plan şablonlarını belirlemelidir.

7. Slayt

Proje planının kalitesi, geliştirme süreci ve ortaya çıkacak ürünün kalitesi için belirleyicidir. Yüksek kalite planın anlaşılmasını, izlenmesini ve gerçekleştirilmesini kolaylaştırır. Kaliteli bir planda görevler gerçekleştirilebilir, başarı kriter ve çıktıları tanımlı, tutarlı ve tek kişinin yapabileceği seviyede bölümlenmiştir. Görev süreleri kısa ve takibi kolaydır. Plan, projeyle ilgili tüm faaliyetleri içerir. Mesai dışı çalışmalara değil, ekibin makul iş sürelerine göre yapılmıştır.

Plandaki işlemleri paralel hale getirmek, proje süresini kısaltır. Özellikle sıralı yapılması gerekmeyen işler paralel hale getirilmelidir. Fonksiyonel ve teknik ihtiyaçlarla ilgili işlemler paralel yürütülebilir. Proje modüler hale getirilirse, planlama basitleşir ve proje daha kolay takip edilebilir. Projenin alt projelere bölünmesi tasarım ve kodlama gibi birçok açıdan da ilave kolaylık sağlar. Kapalı çevrim yöntemi proje akışından bağımsız görev tanımlı yapmayı önler ve görevler arası bağlantıların doğru şekilde yapılmasına yardımcı olur. Ancak bu yöntemi uygularken çok katı davranmak planı aşırı karmaşıktırabilir.

Planlamada ortak doğrular olduğu gibi ortak hatalar da vardır. Hataları arka plandaki etkenleri dikkate alarak incelemek gerekir. Kurum ve kişiler planlamaya önem vermemek, acelecilik, işi basit görmek, tecrübe eksikliği, önceki başarıları yanlış değerlendirmek, mali şartlar, müşteri ve üst yönetim zorlaması gibi sebeplerle hata yapabilir. Hatalar, hiç plan yapmamak veya yetersiz planlamadan, her şeyin en başta planlanabileceği zannıyla planlamada aşırıya kaçmaya kadar uç noktalarda gözlenir.