

5.5. Proje Bölünleşme (Entegrasyon) Planı

Geliştirilen projelerde bilgilerin, farklı planların, kaynakların, değişen hedef ve alternatiflerin bölünleşik olarak yönetilmesi gereklidir (PMBOK-2008). Bölünleşme planında; kullanılacak yöntem, ara yüzler, sistemler arası veri alışverişi ve konunun sorunluları bulunmaktadır. *Bölünleşme* işleme plan yaparken ihtiyaç duyulanları veya bir yazılımın alt modülleri arasında entegrasyon sağlarken ihtiyaç duyulabilir:

- **Plan türlerini bölünleşik yürütme:** Kapsam, maliyet ve risk gibi farklı planların projede birlikte uygulanmasında nasıl hareket edileceği bir tür bölünleşme planıdır (PMBOK-2008).
- **Sistemler arası bölünleşik çalışma:** Güncümüzde kurulumlar temel yazılım ve donanım alt yapılarını oluşturmuş durumdadır. Yeni yazılacak bir projenin var olan projelerle birlikte çalışması gerekir.
- **Modüller arası bölünleşme:** Eğer çok büyük bir sistem tasarlanıyorsa, bu sistemin alt sistemlere bölünerek tasarlanması gerekir. Bunlar arasındaki bölünleşme ve iletişim de planlanmalıdır.

5.6. Görevlerin Belirlenmesi: İş Bölünleşme Yapısı

Kapsam belirlendikten sonra yapılacak görevlerin ayrıntılı belirlenmesi aşamasına geçilir. Yazılım geliştirme karmaşık bir işlemdir. Orta ölçekli bir yazılımda dahi birçok aşamaya dağılmış yüzlerce ilişkili görev vardır. Bu görevlerin tutarlılık bozulmadan, müşteri talep ve teknik gereksinimleri karşılayacak şekilde gerçekleştirilmeleri gerekir. Yapılan araştırmalar yazılımın karmaşıklığı tekrar tekrar ortaya koymuştur.

Yapılan araştırmalar yazılımın karmaşıklığı tekrar tekrar ortaya koymuştur.

Hic kimse modern bir bilgisayar programını tamamıyla bir defada anlayabilecek bir zekaya sahip değildir. [Edsger Dijkstra 1972]

Yapılması gereken zekâmızı bir defada tüm karmaşayı yönetecek şekilde geliştirmeye çalışmak değildir. Zira bu zaten mümkün değil. [McConnell-2004.11]

Karmaşıklığı azaltmanın en önemli yöntemi, sistemi kendi içerisinde anlamlı parçalara bölmektir. Bir yazılımı oluşturan yapılar özeten ayrıntıya doğru model, paket, nesne, metot ve metot içerisindeki kodlar şeklinde ifade edilebilir. Bu yapıya yazılım mimarisi veya yazılımın mimari bölünleşmesi ismi verilir. Yazılımın bölünleşmesi ayrılmaları diğer yapıların yanında görevlerin belirlenmesi ve atanmasını da kolaylaştırır. Örneğin bir nesneyle ilgili yapılacak işler, bir görevin altında toplanabilir. Yazılım proje planı, yazılımın hiyerarşik yapısına paralel olarak geliştirilmelidir. Yapılacaklar seviyelendirilerek birim işlere veya alt görevlere bölünür. Böylece yapılacakların tanımlanması ve birbirleriyle ilişkilerinin belirlenmesi gerçekleştirilir. Bu bölünleşmeye *İş Bölünleşme Yapısı* veya *İş Bölünleşme Ağacı* (*Work Breakdown Structure*) ismi verilir. İş bölünleşme yapısı planın temelidir.

İş bölünleşme yapısının amacı, yazılım projesindeki işlemleri yönetilebilir küçük parçalara bölmektir. Böylece her parçayı gerçekleştirecek kişi, gerçekleştirmeye zamanı ve tamamlanma kriteri tanımlanabilir. Görevler gruplanarak proje geliştirme sürecinin önemli noktalarında için kilometre taşı veya kontrol noktası oluşturulur. Görevler arası ilişkiler *Gantt* ve *Ağ Şeması* gibi farklı gösterim şekilleriyle analiz edilebilir.

Yazılım mimarisi ve iş bölünleşmesi birbirine paralel ancak farklı kavramlardır (Fairley-2009). Yazılım mimarisi genellikle nesneleri açıklar ve isim cümleleriyle ifade edilir. İş bölünleşme yapısı ise yapılacakları fiil cümleleriyle ifade eder. Bir görevde tek bir nesne olması tercih edilir. Ancak bir nesneyle ilgili birden fazla görev tanımlanabilir gibi bir görev tanımı da birden fazla nesneyi etkileyebilir. Aradaki eşleşme planında yer almaz. Yazılım bileşenleri arasında hiyerarşik haricinde de ilişkiler olabilir. Örneğin kod kütüphanesindeki ortak bir fonksiyonu birçok bileşen kullanacaksa, bu fonksiyon diğerlerinden önce tasarlanmalıdır.

5.6.1. Görev Tanımı

Görev proje planındaki insan veya diğer kaynağın yapacağı, manalı, bütünsel ve birim işler. *Görev* genellikle iş bölünleşme yapısının en alt seviyesinde gösterilir. Görevlerin gruplandığı üst işler de *özet görev* olarak adlandırılır. Her görevin bir kişi tarafından yürütülmesi tercih edilir. Görevin özellikleri:

- Görev tanımı
- Başlangıç ve bitiş tarihi, tahmini süre
- Gerekti kaynaklar: Kişi, araç gibi
- Girdiler ve çıktılar, üretilcek ürün
- Başarı/Tamamlanma kriterleri
- Riskler (istenirse tanımlanabilir)

5.6.2. Görevlerin Tespiti İçin Yöntemler

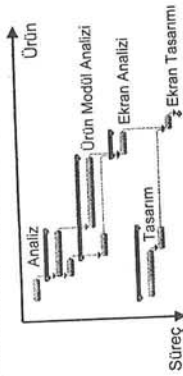
Proje planının yapısı; özeten ayrıntıya, ayrıntıdan özete, akış mantığına veya birden fazla yöntem birlikte düşünülerek inşa edilebilir. Özeten ayrıntıya incelemede önce-likle en temel görev tespiti edilir. Temel görev alt görevlere bölünür. Bölünleşme sistemin en alt seviyesine ulaşmaya kadar devam eder. Bu planın ayrıntıdan özete inceleme yapılacak bir işin eksik kalıp kalmadığı tespit etmeyi sağlar. Bir projede farklı yöntemlerden faydalanılabilir ve hatta faydalanılması da gereklidir. En önemli şey nereden başladığının unutulmamasıdır [Nizam 2011].

Akış-Süreç temelli analiz, işlerin birbirine bağlantısını incelemek ve sırasını belirlemek için kullanılabilecek bir yöntemdir. İşlerin birbiriyle olan bağlantısı hiyerarşik yapıdan daha karmaşık olabilir. Bu yüzden son yıllarda proje plan oluşturma araçlarında hiyerarşik *Gantt Şeması* yerine *Ağ Şeması* almaktadır.

Model tekniğini belirlerken proje yapısı önemlidir. Temel bir amaç etrafında geliştirilen bir proje, özeten ayrıntıya daha kolay planlanır. Temel amaç belirsizse ayrıntıdan özete bir modelleme tekniği kullanmak daha uygundur. [McConnell-2004.11]

İş bölümleri, ürün ve yazılım süreci uyumu

İşlerin bölünmesi ürün ve süreç şeklinde iki eksenle düşünülebilir; bkz. Şekil 5.5. Yazılım mimarisi işlerin hiyerarşik ayrıştırılması, yazılım süreci de zamana bağlı sırasını belirlemeye yardımcı olur. Örneğin bir muhasebe sisteminde geliştirilecek her bir ekran için *geliştirme aşamasının* en alt seviyesinde bir görev tanımlanabilir. Bu ekranların analizi planın analiz aşamasında, tasarımı planın tasarım aşamasında yer alır. Farklı tür yazılım süreçlerinde farklı eksen yapıları kullanılabilir.



Şekil 5.5. Görev yapısında ürün ve süreç eksenli

Görevler, tercih edilen yazılım süreç yönetim yöntemine göre proje plana yerleştirilir. Örneğin şelale modeli için sıra analiz, tasarım, kodlama şeklindedir. Artımlı geliştirmede ise her çevrim plana dâhil edilmelidir.

Analizde istenenlerin ana faaliyet kabul edilmesi

Genellikle analizdeki bir ihtiyaç için tasarım birden fazla bileşen içerir. Bu yüzden hiyerarşik yapıda analizdeki ihtiyaçların üst kademelere yazılması önerilir (Fairley-2009). Böylece yapılan işlerle istekler arasında net bir bağlantı kurularak isteklerin ne kadarının karşılandığı görülebilir. Müşterinin de planı izlemesi kolaylaşır.

5.6.3. Görevler Arası İlişki Türleri

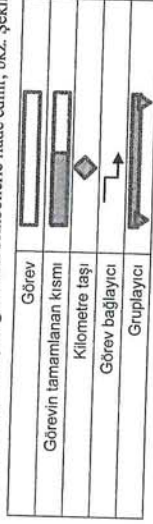
Proje planı yapılan işler arasında gerçek hayatta var olan ilişkileri yansıtmalıdır. Proje süreci ve yapılacakların sırasının belirlenmesinde bu ilişkilerin çok büyük önemi vardır. İlişkiler en çok kullandılardan başlayarak aşağıdaki tabloda sıralanmıştır.

İlişki Türü	Açıklama	Örnek
Bitiş-Başlangıç	Bir görevin bitiş tarihi diğerinin başlangıç tarihini belirler	Tasarım bitmeden kodlama başlayamaz veya tasarım bitince kodlama başlar
Başlangıç-Başlangıç	İki görevin aynı anda başlaması	Yazılım geliştirme ve donanım temini birlikte başlayabilir.
Bitiş-Bitiş	Bir görevin bitiş diğerinin de tamamlanmasına sebep olabilir.	Test tamamlanmadıkça donanımlarda temin edilmiş olmalıdır ki proje devreye alınabilsin
Başlangıç-Bitiş	Bir görevin başlangıç tarihinin diğerinin bitiş tarihini belirlemesi	Tasarım konusunda dışarıdan alınacak ve tarihi belli bir uzman desteği başlamadan önce ihtiyaç analizinin bitirilmesi

Gerçek hayatta her zaman, bir görevin diğerinin tam bittiği tarihte başlaması gerekmez. Burada gecikme süresi (*lag time*) kavramı devreye girer. Örneğin bitiş başlangıç ilişkisinde, bir görev diğeri tamamlandıktan 3 gün sonra başlayabilir.

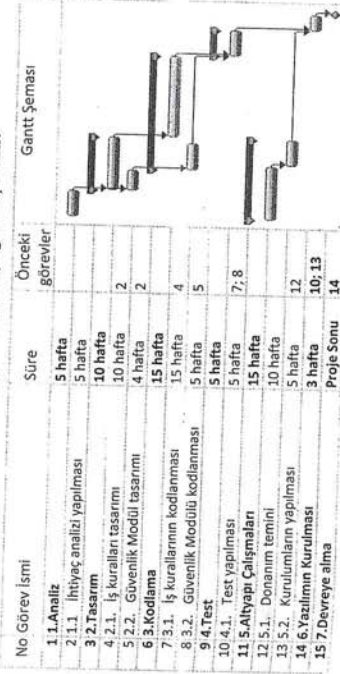
5.6.4. Gantt Şeması

Gantt Şeması, Grant H. tarafından önerilen, proje planını görsel olarak yansıtan ve planlama araçlarında çok kullanılan bir çizelgedir. Görev takvimi, görevler arası bağlantılar ve kilometre taşları bu çizelgede özel sembollerle ifade edilir; bkz. Şekil 5.6.



Şekil 5.6 Gantt Şeması şekilleri

Örnek bir projenin *Gantt Şeması* Şekil 5.7'de gösterilmiştir. Bu çizelgede genellikle görevlerin isimleri yer almaz. Görevlerin hiyerarşik yapısı ve çizelge, paralel olarak çizilir. Bu sayede hangi görevin hangi şekle karşı düştüğü anlaşılabilir.



Şekil 5.7. Örnek proje planı

5.7. Kaynakların Planlanması

Kaynak planlama aşamasında öncelikle projede hangi uzmanlıklara hangi tarih aralığında ihtiyaç duyulduğu belirlenir. Sonrasında kurumun bu uzmanlıkları nasıl sağlayacağı, kendi imkânlarıyla mı yoksa dış kaynaklardan mı temin edeceği ayrıntılı şekilde belirlenerek plana eklenir.

Kaynak planı hazırlanması, zaman planıyla birlikte yürütülmesi gereken bir çalışmadır. Görevin süresini, istihdam edilen kişinin üretkenliği ve projeye katılım oranı belirler. Bunu yaparken kaynak takvimleri ayrıntılı incelenir. İzmiter, hastalık ihtimali ve resmi tatiller de dikkate alınmalıdır. Kaynak planlanmasında yapılanlar:

- **Kaynakların belirlenmesi:** Hangi tür kaynaklara ihtiyaç duyulduğun tespiti (kişiler için rol), kaynak temin önerilerinin değerlendirilmesi ve seçimi, kaynak takvimleri oluşturma
- **Kaynak maliyetlerinin belirlenmesi:** Normal çalışma, fazla mesai ve diğerler için gider ve maliyet değişim tablosu tanımlama
- **Bütçeleme:** Ayrıntı maliyetleri toplayarak proje bütçesini oluşturma
- **Görevlere kaynak atma:** Kaynak görev eşleştirilmesi, kaynağın görevde ne kadar yer alacağını belirlenmesi, kaynak atamalarının kaynak ve faaliyete göre kontrol edilmesi
- **Çıkışma düzenleme:** Kaynak çıkışmalarını gözden geçirme ve düzeltme

Görev tanımlarının netleşmesi, projedeki kaynak ihtiyacının da tam olarak ortaya çıkmasını sağlar. Görevlerdeki çıkışmalar ve aşırı yüklenen kaynaklar tespit edilir. Görevler, çıkışmalar en aza indirilecek veya çözümlenecek şekilde tekrar zamanlanır. Görevlerin kaydırılmasında özellikle *boluluk zamanları* dikkate alınır.

5.7.1. Kaynak Türleri

Projedeki hedefleri gerçekleştirmek ve görevleri yerine getirmek için kullanılan her şey projenin kaynağıdır. Bunlar görev yapan kişi, işlerin yapılmasında kullanılan malzeme, makine ve diğer mali değerler olarak sayılabilir. Kaynaklar yapılarına göre farklı şekillerde sınıflandırılır. Her kaynak türü kendi içerisinde farklı özellikler gösterir. *Kaynak türü* kaynağın nasıl sağlanacağını, kullanım şeklini ve maliyetini belirleyen yardımcı olur. Kaynak türleri:

- **İnsan:** İşlerin yerine getirilmesi için gerekli uzmanlık ve kişileri ifade eder. İnsan her projenin kaynağı ve hedefidir. Genellikle diğer kaynaklardan ayrı değerlendirilir. Öncelikle ihtiyaç duyulan uzmanlık, sonrasında buna sahip kişilerin tespiti önemlenmektedir.
- **Malzeme:** Projenin tamamlanması için gerekli malzemelerdir. Kırtasiye, bilgisayar, yazılım ve donanım şeklinde sayılabilir. Malzemelerin satın alma ve bakım maliyetleri dikkate alınmalıdır.
- **Mali:** Doğrudan mali değer olarak ifade edilen kaynaklardır. Yapılan seyahat ve otel masrafları gibi maliyetler örnek olarak verilebilir.
- **Diğer:** Projenin yapılması için gerekli altyapı, kurumsal standartlar, önceki projelerden elde edilen terimler ve kod kütüphaneleri gibi kurumsal kaynaklardır.

5.7.2. Kaynak Takvimlerinin Belirlenmesi

Kaynağın projede aktif görev alabileceği süre, katılım yüzdesi ve takvimi belirlenmelidir. Görevlerin gerçek çalışma zamanlarına göre planlanması için kaynak takvimlerinin proje başında oluşturulması uygun olur. Kişi ve diğer kaynakların proje çalışmalarını etkileyecek tüm bilgileri sisteme girilmelidir. Örneğin izin ve seyahatler takvime işlenmelidir. Girilmesi unutulmuş her süre, projeye olumsuz yansır ve riski artırır. Hastalık gibi beklenmeyen durumlar düşülerek belli bir gevşek zaman bırakılmalıdır.

5.7.3. Kaynakların Göreve Atanması

Kaynak atama, görev ile ilgili kişi ve diğer kaynakların eşleştirilmesidir. Kişi görevi atandığında kişinin uzmanlığı, proje takvimi ve kişinin takvimi karşılaştırılarak görevin alacağı toplam süre bulunur. Bu eşleştirmede kişinin görevi ne kadar zaman ayracağı belirlenir. Görev için harcanacak emek ve süre aşağıdaki değişkenlerle modellenir [Quiring-2012]:

- **Süre:** Görev için planlanan toplam süre. Gün cinsinden ölçülür.
- **Birim:** Bir kaynağın toplam zamanının ne kadarını görev için ayrabileceğidir. Yüzde cinsinden ölçülür.
- **Çalışma:** Görevi yapmak için fiilen ne kadar zaman gerektiğidir. Saat cinsinden ölçülür.

Bir kişi 2 gün süren bir görevi tamamlamak için günlük 8 saatlik mesainin 4 saatini harcamışsa, *süre* 2 gün, *birim* %50 ve *çalışma* 8 saat olarak bulunur. Görevin süresi, harcanacak emek ve kaynak ilişkisinde aşağıdaki denklem kullanılır.

$$\text{Süre} \times \text{Birim} = \text{Çalışma}$$

Yukarıdaki denklemde değişkenlerden birisi sabit kabul edilirse kalan iki değişken birbirini etkiler. Burada bir görev için temelde üç durum söz konusudur. Bu durumlar görev türleri olarak standartlaşmıştır:

- **Sabit Süre:** Görevin alacağı toplam (gün) süresi sabittir. Süre ilave kaynaklarla değişmez. Örneğin "donanım temini" görevi ne kadar kaynak ilave edilirse edilsin, sağlayıcı firmanın teslimat süresine bağlıdır.
- **Sabit Birim:** Çalışan kişinin görevi ayracağı birim zaman sabittir. Bu birim zaman görev süresi veya çalışma değişiminden etkilenmez. İlave kaynak atanırsa süreye kısalır. Bir programcı tek görevdeki birbine bağlı iki ekranı 4 günde yazıyorsa ikinci bir programcı ilavesi süreyi iki güne düşürür. Dikkat edilirse programcı çalışma yüzdesi değil, sayısı artırılmıştır.
- **Sabit Çalışma:** Toplam fiili çalışma süresi (saat) sabittir. Fiili çalışma süresi görevin süresindeki veya birim çalışmadaki değişimden etkilenmez. Kişilerin çalışma yüzdesi artarsa görev süresi kısalır. Kişinin zamanının % 50'ini harcayarak 40 saatlik çalışmayla ve 10 günde geliştirdiği ekran eğer zamanının %100'ünü harcaysa yine 40 saatte ancak bu defa 5 günde tamamlanır. Çalışma birimi %25'e düşerse görev 20 gün sürer.

Yukarıda sıralanan durumlar, her zaman geçerlidir denemez. Örneğin *sabit birim* türünden 10 gün süren bir görev, 9 kişi ilave etmek süreyi 1 güne düşürmez. Proje yönetim yazılımının bu konuda yaptıkları otomatik hesaplar ancak kontrol edildikten sonra plana uygulanmalıdır.

Kaynağın görevle atanmasıyla görevin gerçek süre ve maliyet tahmini ortaya çıkar. Kaynağın görevle atanmasıyla görevin gerçek süre ve maliyet tahmini ortaya çıkar. Plan gerçek durumu daha ayrıntılı yansıtmaya başlar. Atanan kaynağın becerisi (üretkenliği, *productivity*) ve işe ayırdığı verimli zaman (*utilization*) da görev süresi üzerinde önemli ölçüde etkilidir. %70 üretken çalışan bir kişi zamanının %60'lık kısmında verimli çalışıyorsa 10 günlük bir iş $(10 / 0.7) / 0.8 = 17$ gün sürecektir. Daha kalifiye bir personelin atanması, görevin daha kısa sürede bitmesini sağlayabilir.

5.7.4. Kaynak Dengeleme

Kaynakların işlere atanması sonrasında bazı kaynaklar üzerinde birden fazla iş aynı zamanda çalışabilir. Bu durumda kişi veya diğer kaynak aşırı yüklenmiş olur. *Kaynak dengeleme*, aşırı yükli bir kaynağın iş yükünün yapılacak düzenlemeyle azaltılmasıdır. Aşırı yüklemeleri düzenlemek için öneriler (Happy-2010):

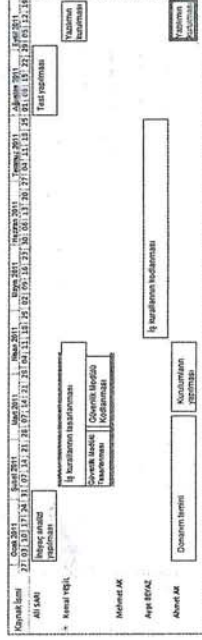
- Fazla yüklemeye yapılmış kaynak tanımlı ve bu kaynakların belirlenmesi
- Görev önceliklerinin belirlenmesi; Kişinin üzerindeki yüksek öncelikli görevler ilk olarak yerleştirilir. Düşük öncelikli görevlerse kaydırılabilir.
- Kısıtlara (*constraints*) sahip görevler önce yerleştirilir.
- Kaynak üzerindeki görevleri kısaltacak düzenlemeler yapılabilir.
- Bolluk süresi fazla görev önce kaydırılır (Kısım 5.8.3'de açıklanacaktır).
- Erken başlangıç tarihine sahip görev, geç başlangıça göre önce kaydırılır.
- Kaynağın başka projelere ayırdığı zaman azaltılabilir.

Proje planı üzerinden kaynakların yoğunluğunu gösteren raporlar da alınabilir. Bunu gösterebilmek için ilk örnekteki plana kaynaklar eklenmiştir.

No	Görev İsmi	Süre	Önceki görevler	Kaynak İsimleri
1	1. Analiz	5 hafta		Ali SARI
2	1.1. İhtiyaç analizi yapılması	5 hafta		Kemal YEŞİL
3	2. Tasarım	10 hafta	2	Kemal YEŞİL
4	2.1. İş kuraları tasarımı	4 hafta	2	Ayşe BEYAZ
5	2.2. Güvenlik Modülü tasarımı	15 hafta	4	Kemal YEŞİL
6	3. Kodlama	15 hafta	5	Ali SARI
7	3.1. İş kuralının kodlanması	5 hafta	5	Ahmet AK
8	3.2. Güvenlik Modülü Kodlanması	5 hafta	5	Ahmet AK
9	4. Test	5 hafta	7; 8	Kemal YEŞİL; Ahmet AK
10	4.1. Test yapılması	15 hafta	12	Kemal YEŞİL; Ahmet AK
11	5. Altyapı Çalışmaları	10 hafta	10; 13	Kemal YEŞİL; Ahmet AK
12	5.1. Donanım temini	5 hafta	14	
13	5.2. Kurulumların yapılması	3 hafta		
14	6. Yazılımın Kurulması	Proje Sonu		
15	7. Devreye alma			

Papatya Yayıncılık Eğitim

Kişinin iş yoğunluğu ve çıkışmalar, proje planlama araçları yardımıyla bir takvim üzerinde gösterilir; bkz. Şekil 5.8. Bu şekil *Microsoft® Project* türünden alınarak düzenlenmiştir. "Kemal Yeşil" üzerinde aynı anda birden fazla iş olduğu şemadaki farklı şekil ve renkleri gösterimden kolayca anlaşılır. Birden fazla kişiye atanan görevler de "Yazılımın Kurulması" görevinde olduğu gibi farklı şekilde gösterilir.



Şekil 5.8. Kişilerin üzerindeki iş yükleri

Bazı küçük üst üste binmeler ihmal edilebilir. Proje plan yazılımları aşırı yüklemeleri çözmek için proje, kaynak veya görev seviyesinde çeşitli otomatik çözüm araçlarına sahiptir. Bu işlem *kaynak seviyelenme (leveling)* olarak adlandırılır. Ancak bu tür çözümler bir benzetim kabul edilmeli ve kontrol edilmeden uygulanmalıdır.

5.8. Proje Zaman Planı

Proje zaman planının hedefi, görev takvimlerini projeyi zamanında tamamlayacak şekilde oluşturmaktır. Projenin ilk aşamalarında başlangıç ve bitiş tarihlerinden en azından birisi kabaca belli olmaktadır. Görev süreleri de ilk aşamada aşağı yukarı tahmin edilir. Görevlere kaynakların atanmasından sonra görev süreleri daha net hale gelir. Görev süreleri kritik yol hesabıyla toplanarak, toplam proje süresi ortaya çıkar. Plan müşterinin veya üst yönetimin belirleyeceği kısıtlara göre tekrar incelenip düzenlenir ve gerekirse süre kısıtlamaya çalışılır.

5.8.1. Zaman Çizelgesi Oluşturma Yöntemleri

Zaman çizelgesi aşağıdaki yöntemleri kullanarak görevler ve görevler arasındaki bağımlılıkların farklı açılardan analiziyle hazırlanır (PMBOK-2008).

1. Ağ analizi: İşlemler arasındaki ilişkilerin, iş akış mantığı ve ağ şeması kullanarak analizidir.
2. Kritik yol: Proje süresinin uzunluğunu belirleyen artarda işlemleri tespit yöntemi. Kritik yol ileri ve geri doğru incelenerek bir görevin önce veya sonra başlatılabileceği süreler tespit edilir.
3. Kritik zincir: Kaynak sınırlarının zaman çizelgesine etkisini incelemekte kullanılır. Kaynak kullanılabilirlikleri girdikten sonra kritik yol genellikle değişir.

4. **Kaynak dengeleme:** Kaynak kullanımının bir süre aralığında sıkışmasının önüne geçilmesi ve düzenli kullanım sağlamak için planın tekrar düzenlenmesidir. Kritik yol bu durumda da değişebilir.
5. **Senaryo analizleri:** Varsayılan bir özel durumunun gerçekleşmesi halinde planı nasıl etkileyeceği incelenir. Olumsuz durumların önlemek veya etkilerini azaltmak hedefiyle yapılan bir çalışmadır.
6. **Önden gitme ve bekleme:** Planın önce başlama veya gecikme durumlarına göre tekrar incelenmesi ve gerekirse düzenlenmesidir.
7. **Sıkıştırma:** Kaynak ilavesi ve işlerin paralel hale getirilmesiyle kapsam değişiminden plan süresinin kısaltılmasına yönelik çalışmalardır.

5.8.2. Proje Takvimi Oluşturulması

Proje takvimi çeşitli şekillerde belirlenir. Genellikle projenin bitiş tarihi veya süresi belli olur. Bazı projelerin de başlangıç tarihleri belli olabilir. Olusabilecek durumlar:

- **Süresi belirli projeler:** Bu tür projelerde projenin süresi belirli ancak ilk anda başlangıç tarihi belirsizdir. Özellikle inaleyle alınan projelerde bu durum ortaya çıkar. Alıcı firma şartnamede süreyi belirtir. İhale sürecinin ne kadar süreceği net belli olmadığından ilk anda başlangıç ve bitiş tarihleri net değildir. İhale sonucunda iş başlangıç tarihi belirlendiğinde son tarih de belli olmuş olur. Bu durumda projenin en uzun yolu müşteri tarafından belirlenir. Yapılacak tüm işler bu süreye sığdırılmaktadır.
- **Bitiş tarihi belirli projeler:** Projenin bitiş tarihi kesin belirlidir. Şirketlerde mali işler, stok gibi dönem esaslı çalışan birimler için geliştirilen projeler bu türdür. Proje başlangıcı gecikirse süre kısalmır. Örneğin 2013 başında devreye girecek bir proje 2012 Nisan'da başlarsa 9, 2012 Ekim'de başlarsa 4 ay sürer. Yapılacak son işin bitiş tarihi, proje bitiş tarihine eşitlenerek toplam süre belirlenir. İşler bu süreye sığdırılır. Proje yöneticisi, bu tür projeleri mümkün olan en kısa sürede başlatmaya çalışmalıdır.
- **Başlangıç tarihi belirli projeler:** Bazı projelerin başlangıç tarihi belirlidir. Bitiş tarihini belirlemek de proje ekibine bırakılmıştır. Proje yöneticisi ve ekip, plandaki görev akışına göre toplam süreyi ve bitiş tarihini önerir. Bu durum ideal olmakla birlikte, bu tür projeler çok nadirdir.

Proje takvimi normalde günlük 8 saatlik bir çalışmaya göre belirlenir. Proje takviminde ayrıca resmi tatil günleri gibi projede çalışan tüm kişileri etkileyecek bilgiler bulunmalıdır. Projenin kritikliğine göre çift vardiyalı çalışma gibi farklı çalışmalar da takvime eklenir. Ancak planın ilk anda fazla mesai yapılacağı gibi düzenlenmesi, sonraki aşamalar için yedek kaynak bırakmayacağından, *önerilmez!*

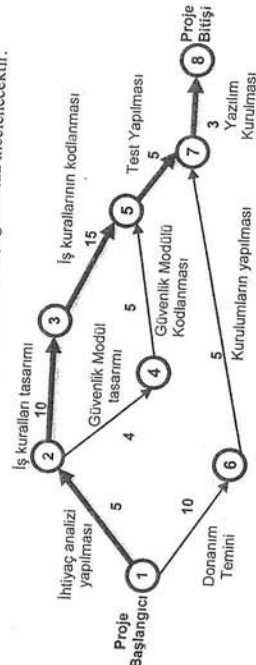
5.8.3. Kritik Yol Yöntemi

Projenin bitmesi için izlenebilecek alternatif yolların, toplam süre olarak en uzunlu, kritik yol olarak isimlendirilir (Kerlley-1959). Toplam süre, projenin başlangıcından bitişine kadar birbiri takip eden görevlerin süreleri toplamıdır.

Kritik yol en uzun süreli yol olduğu için projenin toplam süresini gösterir. Kritik yoldaki görevlerin herhangi birinin gecikmesi proje süresini uzatır. Farklı bir bakışla kritik yol proje süresini uzatabilecek kritik görevleri bulmayı sağlar denebilir. Kritik yola paralel diğer yollar kaynaklarda düşük kritiklikte veya *kritik olmayan şekilde* isimlendirilir. Ancak *kritik olmayan* ifadesi, bazı görevlerin öneminin, olduğundan daha düşük değerlendirilmesine yol açmamalıdır. Çünkü bu görevlerin de aşırı uzaması durumunda kritik yol yine değişecektir.

Görev No	Görev İsmi	Tahmini Ortalama Süre	En Kısa Süre	En Uzun Süre	Önceki görevler
1	1. Analiz	5 hafta	4	7	
2	1.1. İhtiyaç analizi yapılması	5 hafta	4	7	
3	2. Tasarım	10 hafta	8	11	
4	2.1. İş kuralları tasarımı	10 hafta	8	11	
5	2.2. Güvenlik Modülü tasarımı	4 hafta	4	5	2
6	3. Kodlama	15 hafta	14	20	
7	3.1. İş kurallarının kodlanması	15 hafta	14	20	
8	3.2. Güvenlik Modülü kodlanması	5 hafta	5	7	5
9	4. Test	5 hafta	4	8	
10	4.1. Test yapılması	5 hafta	4	8	
11	5. Altyapı Çalışmaları	15 hafta	9	20	
12	5.1. Donanım temini	10 hafta	5	12	
13	5.2. Kurulumların yapılması	5 hafta	4	8	
14	6. Yazılımın Kurulması	3 hafta	2	4	12
15	7. Devreye alma	Proje Sonu	-	-	10, 13

Yukarıdaki proje planındaki görevler arasındaki bağlantılar ağ şeması üzerinde de gösterilebilir; bkz. Şekil 5.10. Görevlerin akışını gösteren birçok farklı gösterim şekli olmakla birlikte bu çalışmada hepisi *ağ şeması* başlığı altında incelenecektir.



Şekil 5.9. Proje ağ şeması

Örnekte kritik yol hesabının nasıl yapıldığı grafik üzerinde gösterilmiştir. Koyu renkli okların toplam süresi, projenin toplam süresidir; bkz. Şekil 5.9. Bu süre 1-2-3-4-5-7-8 yolundaki görevlerin toplamından 38 hafta olarak hesaplanmıştır.

Ağ şemasında görevlerin birleştiği ve ayrıldığı kesişim noktalarına *kilometre taşı* ismi verilir. *Kilometre taşları*, şekilde daire içerisinde numaralarla gösterilmiştir. Uygulamada ise sadece önemli noktalar, kilometre taşı veya kontrol noktası olarak belirlenir ve projenin bu noktadaki performansına göre genel başarı değerlendirilir.

Görevlerin bazılarının uzaması kritik yolu değiştirebilir. Bu durumda kritik yol haricindeki görevlerin başlangıç ve bitiş tarihlerinin ne kadar değişebileceği önem kazanır. Bu amaçla öncelikle bir görevin en erken-en geç başlayabileceği ve bitebileceği noktaların hesaplanması gerekir. Böylece görevin zaman içerisinde ne kadar kaydırılabileceği tespit edilmis olur. Aşağıdaki hesaplamalar görevler arasında *bitiş başlangıç* ilişki olduğu düşünülerek yapılmıştır. (İfadeler basitleştirilmesine rağmen belli ölçüde karışık olduğundan örneklerin incelenmesi özellikle tavsiye edilir.)

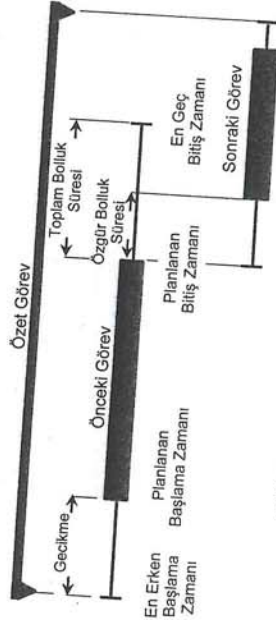
- **En erken başlama zamanı:** Bir görevin *en erken başlama zamanı*, öncesindeki görevin en erken başlama zamanı ile yine öncesindeki görevin süresi toplamıdır. Öncesinde birden fazla görev olan bir görevin (bir görevin başlangıç noktasındaki kilometre taşına birden fazla paralel yolla ulaşıyorsa) en erken başlama zamanı önceki görevler içinden erken başlama zamanıyla süresi toplamı en büyük olan seçilerek belirlenir.

Şekildeki 2 noktalı kilometre taşından sonraki görevlerin en erken başlama zamanı "*İhtiyaç analizi yapılması*" görevi için başlangıçtan itibaren 5 haftadır. "Test yapılması" görevinin öncesinde 5'olu kilometre taşı mevcuttur. Bu noktaya $1 \rightarrow 2 \rightarrow 3 \rightarrow 5$ ve $1 \rightarrow 2 \rightarrow 4 \rightarrow 5$ yollarından ulaşılabilir. $1 \rightarrow 2 \rightarrow 3 \rightarrow 5$ yolu için en erken başlama zamanı $5+10+15=30$. $1 \rightarrow 2 \rightarrow 4 \rightarrow 5$ yolu için en erken başlama zamanı $5+4+5=14$ haftadır. Bu durumda 5 noktasındaki görevler için erken başlama zamanı 30 olarak seçilir. Çünkü her iki yoldan bu noktaya ulaşılmadan 5 noktasına bağlı bir görev başlayamaz.

- **En erken bitiş zamanı:** Görevin en erken başlama zamanı ile süresi toplamıdır.
- **En geç başlama zamanı:** Bir görevin proje süresini uzatmayacak en geç başlama zamanıdır. Bu sürenin hesabı, proje bitiş noktasından başlanarak geriye dönük yapılır. Proje bitişinin en geç başlama zamanı ile en erken başlama zamanı eşit kabul edilir. *Önceki bir görevin en geç başlama zamanı* bu görevin süresi sonraki görevin en geç başlama zamanından çıkarılarak bulunur. Kendisini takip eden kilometre taşından birden fazla görev başlayan bir görevin en geç başlama zamanı, en geç başlama zamanı en küçük olan sonraki görev üzerinden belirlenir. Şekildeki 8 noktalı kilometre taşına bağlı "*Yazılım Kurulması*" görevinin en geç başlama zamanı $38-3=35$ haftadır. 5 noktasının en geç başlama zamanı $35-5=30$ haftadır. 3 noktasının $30-15=15$ ve 4 noktasının $30-5=25$ hafta şeklinde hesaplanır. 2 noktasında plan iki yola ayrılmaktadır. Bu yollardan $7 \rightarrow 5 \rightarrow 3 \rightarrow 2$ sırası izlendiğinde en geç başlama zamanı 5, $7 \rightarrow 5 \rightarrow 3 \rightarrow 2$ izlendiğinde en geç başlama zamanı 21 olarak bulunur. Bu durumda 2 noktasında sonlanan görev için 5 değeri en geç başlama zamanı olarak seçilir.

- **En geç bitiş zamanı:** Bir görevin proje süresini geciktirmeden en son bitebileceği zamandır. En geç başlama zamanı ile görev süresi toplamıdır.

Kritik yol haricindeki bir görevin bitişinde, kritik yolun süresini aşmayacak kadar, gecikme olması projenin gecikmesine yol açmaz. Bir görevin proje süresini uzatmayacak kadar geç başlayabileceği süreye, *toplam bolluk-geçerlik süresi* (*total float-slack time*) denir; bkz. Şekil 5.10. Bir görevin diğer hiçbir görevi geciktirmeyecek kadar gelebileceği süreye; *özgür bolluk-geçerlik süresi* (*free float-slack time*) olarak isimlendirilir. Kritik yolda bolluk süresi yoktur.



Şekil 5.10. Görev süresi, bağımlılıklar ve bolluk süresi

Herhangi bir görev için *toplam bolluk zamanı*, en geç başlama zamanından en erken başlama zamanı çıkartılarak bulunur. Örneğin 4 noktası için en erken başlama zamanı $1 \rightarrow 2 \rightarrow 4$ yolu izlenerek 9, en geç başlama zamanı $8 \rightarrow 7 \rightarrow 5 \rightarrow 4$ yolu izlenerek 25 olarak bulunur. Bu durumda $25-9=16$ haftalık bir bolluk süresi oluşur.

Herhangi bir görev için *özgür bolluk süresi*, görevlerin birbirine olan etkileri *toplam bolluk süresi* da dikkat alınarak incelenerek bulunur. Örnekteki donanım temini ve kurulumlar üzerinden giden yol 18 hafta ummaktadır. Kritik yol 38 hafta olduğuna göre donanım temini ile çıkılan yoldaki görevler için 20 haftalık bir *toplam bolluk süresi* mevcuttur. Yollar üzerindeki bolluk süresi görevler üzerine dağıtıldığında asıl tadr. "*Donanım Temini*" için ise özgür bolluk süresi sıfırdır. Çünkü "*Donanım Temini*" gecikirse "*Kurulumların yapılması*" da gecikir.

Proje kısmen incelenirken sadece projenin başlangıç ve bitişine göre değil aradaki paralel yolların da birbirine göre bolluk süreleri bulunabilir. Örneğin $2 \rightarrow 3 \rightarrow 5$ yolu ile $2 \rightarrow 4 \rightarrow 5$ yolu birbiriyle paraleldir. $2 \rightarrow 3 \rightarrow 5$ yolu 25 gün $2 \rightarrow 4 \rightarrow 5$ yolu ise 9 gün süreye sahiptir. Bu durumda ikinci yol için $25-9=16$ gün bir bolluk süresi vardır.

Projenin birden fazla bitiş olması kritik yolu belirlemeyi zorlaştırabilir. Bir projede birden fazla bitiş varsa, birden fazla kritik yol varmış gibi görünür. Uygulamada ise projenin tek bir bitiş mevcuttur. Bu durumda birleştirici yapay görevler tanımlanarak proje planında birden fazla bitiş noktası olması önlenir. [Fairley-2009]

136 Yazılım Proje Yönetimi

Kritik yolun bulunması sonrasında projeye ilgili analiz edilecek birçok yeni konu açığa çıkar. Özellikle kritik yol kritik görevleri bulmayı sağlar. Böylece proje ilerlerken proje yöneticisinin süreyi uzatabilecek işlemleri takip etmesi mümkün olur. Kritik yolun kısıtlaması için yapılabilecek çalışmalar ve kaynak ilaveleri belirlenir.

Kritik yolun bulunması ve süre hesaplanmasında çeşitli matematiksel algoritmalar kullanılmaktadır. Oldukça karışık bu işlemler için proje plan oluşturma yazılımları içerisinde hazır çözümler sunulmaktadır. Bu bölümde kritik yol hesaplamasına örnek olarak ortalamaya süreleri kullanılmıştır. Tahmin kısmında kritik yolun gerçekleştirme ihtimali farklı açılardan incelenecektir.

5.9. Proje Maliyet Planı

Maliyet planı yapılacak işler için gerekli maliyetlerin kullanılacak kaynaklara bağlı olarak plana eklenmesi, bütçeleme, harcama takvimi ve harcama kontrol işlemlerinin belirlenmesinden oluşur. Yazılım geliştirme sürecinde hedeflenen üründe birçok değişim olabilir. İlk anda kaynak kullanımlarını tahmini olduğundan, maliyet de tahmini olarak belirlenir. Proje toplam bütçesi ise genellikle en başta belirlenir ve sabittir. Bu yüzden en başta mümkün olan en ayrıntılı maliyet analiz yapılarak ilerleyen aşamalar da mali sıkıntı oluşması önlenmelidir. Bu amaçla kurum içi geliştirme ve dışardan temin etme gibi alternatifler arasında seçim yapılır. Bu kısımda yazılımı geliştiren kurum için maliyet plan ve analizi ele alınacaktır. Yüklenici kullanımı kısaca belirlenmektedir. Dış kaynak kullanımındaki ayrıntılı maliyet analizi için, bkz. Bölüm 14.

5.9.1. Maliyet Kalemleri

Bir yazılım projesindeki maliyet kalemleri aşağıda sıralanmıştır (Sarıdoğan-2008):

- **Yönetim:** Proje yönetiminde görevlendirilecek kişilerin istihdamı ve kılavuzlar üretme gibi maliyetlerdir.
- **Geliştirme Ekibi:** Geliştirme görevlendirilecek kişilerin istihdamı
- **Yazılım Altyapısı:** Geliştirme, yönetim, test, veritabanı gibi yazılımlar
- **Donanım Altyapısı:** Yazılım geliştirme ve çalıştırma ortamında gerekli sunucu, istemci ve ağ altyapı maliyetleridir. Müşteriye sadece yazılım satılacaksa çalıştırma ortamı maliyeti projeye dâhil değildir.
- **Teslimat:** Yazılımı müşteri ortamına kurmakla ilgili paketlenme, yardım belgesi ve kılavuzlar üretme gibi maliyetlerdir.
- **Test:** Müşteri özel test ve sertifikasyon isteyebilir. Ayrıca müşteri kendi kurumuna test ortamı kurulumu da isteyebilir. Bu maliyetler projeye eklenir.
- **Bakım Destek Hizmetleri:** Müşteriye garanti süresince sağlanacak destek, danışmanlık ve eğitim gibi hizmetler
- **Diğer giderler:** Çalışma ortamı kurma, kurtarıcıye harcaması, seyahat vb.
- **Değişim Maliyeti:** Yazılım projesinde sürekli değişiklik istekleri görülür. Hangi tür talebin yeni *istek* dolayısıyla ilave maliyet kabul edileceği, hangisinin mevcut bütçeden karşılanacağı proje başında kararlaştırılmalıdır.

Fiyat ve maliyet farklı kavramlardır (Sarıdoğan-2008). Fiyat: sağlayıcı-satıcı tarafından kâr, sürdürülebilirlik, müşteri beklentileri gibi birçok faktör göz önüne alınarak ürün veya işgücü için takdir edilen parasal değerdir. Maliyet ise ürünü geliştirmek için harcanan emek ve çabanın parasal karşılığıdır. Yazılımdaki değişim ve riskler yüzünden genellikle projenin fiyatı, maliyetin ve kâr toplamına göre biraz fazla belirlenir.

5.9.2. Kaynak Maliyetlerinin Plana Eklenmesi

Projede kullanılacak her kaynak için maliyet oluşturan tüm işlemler ve her işlem türü için birim maliyet belirlenir. Bu sayede her görevin toplam maliyeti hesaplanır. Süre belirlerken olduğu gibi, burada da maliyet oluşturan tüm işlemler dikkate alınmalıdır. Bazı hizmetler maliyetsiz kabul edilirse, toplam proje maliyeti yanlış hesaplanır.

Maliyetin bir defalık mı yoksa sürekli mi olduğu incelenmesi gereken önemli bir konudur. Örneğin kişiye ücreti süreye bağlı veya bir defaya mahsus şekilde ödenebilir.

5.9.3. Bütçeleme

İdeal durumda projenin maliyet tahminleri proje planı analizi edilerek ortaya çıkar. Görevlerin tahmini maliyetleri toplanarak, *toplam tahmini maliyet* belirlenir. Buna göre gerekli bütçe ayrılır. Uygulamada ise genellikle projelere önceden bir bütçe ayrılır. Plan da bu bütçe sınırları içerisinde kılacak şekilde düzenlenir. Bu düzenlemede öncelikle kritik yol haricindeki işlemlere harcanan maliyeti azaltma çalışması yapılır.

Bütçeyle birlikte harcama takvimleri ve eğer varsa yükleniciye ödeme yapmak için bitirilmesi gereken iş listeleri (*hakediş*) de oluşturulur. Ödeme noktaları genellikle projelerdeki kilometre taşları olarak seçilir.

5.9.4. Maliyet Kontrol ve İzleme

Maliyet bütçesi oluşumundan sonra onaylanır. Belirlenen kilometre taşı geldiğinde yapılanlar kontrol edilir ve değerlendirilir. Yüklenici kullanılıyorsa ve tamamlanma kriterleri karşılanmıyorsa gerekli ödeme yapılır. Yazılım projelerinde ödeme şartları, ortaya çıkan ürüne göre çok net belirlenmelidir. Burada netlik olmaması en çok yazılım sağlayan yüklenici firmayı mağdur eder. Proje maliyetlerinin nasıl izleneceği proje izleme bölümünde ayrıntılı olarak incelenecektir; bkz. Bölüm 9.

5.10. Devreye Alma Planı

Yazılımın devreye alma aşamasında izlenecek yöntem, görev alacak kişiler ve tarihler planlanmalıdır. Devreye alma planı eğitim, yazılım ve teknik altyapının kurulmuş işlemlerini içerir. Yazılım kısmı bitmesine rağmen, donanım teminindeki aksaklıklardan dolayı devreye girme geciken projeler görülmektedir. Örneğin devreye alma aşamasında sunucu odasında yeni makine koymak için yer olmadığı fark edilebilir. Bu durumda yeni bir oda, ağ ve elektrik altyapısı hazırlanması aylar alabilir.

Yazılım projeleri diğer yazılım ve donanımlara bağlı olarak çalışır. Devreye alma planında bulunacak bilgiler:

- Aday yazılımlar ve kurulacak yazılım bileşenleri arası ilişki
- Projenin çalışma ortamındaki donanımlar ve aralarındaki ilişkiler
- Donanımlar üzerindeki yük paylaşım planı
- Yedekli çalışma planı
- Kurulumu yapacak kişilerin görev ve sorumlulukları
- Kuruluma oluşabilecek sorunlar ve bunların çözüm yöntemleri
- Yedekleme ve test sisteminin kurulumu
- Ağ altyapısı konfigürasyonları
- Sunucu odası elektrik vb. diğer altyapılarının planlanması

Projenin teknik altyapısı yazılımın istenilen performans ve kararlılık düzeyinde çalışmasına uygun olarak planlanmalıdır. Projenin teknik hedeflerini sayısal olarak ölçmek için kullanılacak kriterler de yapılacak plana eklenebilir. Örneğin ekran açılma süresi veya rapor çalışma süresi gibi performans hedefleri belirlenebilir.

Proje kurulumu için yapılması gerekenler genellikle yazılım proje ekibinin sınırlarını aşar. Gerekli sunucu makinelere temini, kurulumu ve işletim sistemi ayarları farklı birimler tarafından yapılır. Bu konuda kısa süreli dış uzmanlardan da yardım alınabilir. Diğer birim ve kişilerden alınacak destek en başta planlanmalı, ilgili kişi ve kurumlara mutabakat imzalanmalıdır.

5.11. Eğitim Planı

Yazılım, birçok parçadan oluşan karmaşık bir üründür. Bu yüzden yazılımı devreye almadan önce son kullanıcıya eğitim verilmelidir. Eğitim, farklı kullanım amaçlarına göre çeşitlenir. Projenin üst düzey ayarları hakkında sistem yöneticisi ve kullanıma ilgili son kullanıcı eğitim alabilir. Büyük yazılımlarda, son kullanıcıya sadece doğrudan kullanacağı alt sistem ve bunun ana sistemle ilişkisi konusunda eğitim verilebilir.

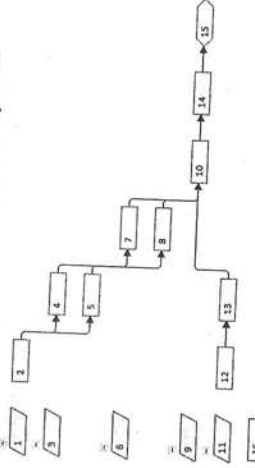
Eğitim planı eğitime katılacak kişiler, eğitimi verecek kişi, eğitim içerik, yer, tarih ve süre bilgilerini içerir. Dış kaynak kullanılan projelerde, destek sürekliliğini sağlamak için kurum içi yazılım ekibini hedefleyen, kod ve geliştirme ortamıyla ilgili eğitimler de planlanır. Bu plan, sadece proje sonunda eğitim verecek şekilde değil her aşamada ekiplerin birlikte ilerleyeceği dikkate alınarak hazırlanmalıdır.

5.12. Planın Farklı Açıdan Analizi ve Tekrar Düzenlenmesi

Proje planında ilk anda sadece görev tanımı ve tahmini süreler mevcuttur. Kaynak tanımı, kaynak takvimi ve kaynak atanması ile plan zenginleşir. Gerçek durumu daha iyi yansıtmaya başlar. Bu aşamada plan farklı açıdan incelenerek süre, maliyet ve kaynak ihtiyacı tekrar düzenlenir. Kaynak dengeleme de bir tekrar düzenlenmedir.

Plan, projeye ilgili birçok bilgi içeren bir veritabanı gibi düşünülebilir (Fairley, 2009). Plan kullanılarak işlemin birbiri ile bağlantılarının ağ şeklinde gösterilimi, toplam maliyet analizi, kişi yüklerinin gösterilmesi, aşırı yük analizi, kişi ve diğer kaynakların verimi gibi konularda standart raporlar üretilmiştir.

Proje planındaki görevler arası ilişkileri yansıtan *ağ diyagramı* planı farklı açıdan incelemeye bir örnektir; bkz. Şekil 5.11. Bu diyagramda görevlerin zamanda dağılımı ve birbirleriyle bağlantılarını akış mantığında bütünsel görmek mümkündür. Günümüzde görevlerin birbirleriyle ilişkileri çok karmaşık bir yapı gösterir. Bu sebeple görevler arası ilişkileri hiyerarşi yerine akış şeklinde modellemek daha faydalıdır.



Şekil 5.11. Proje ağ şeması (Microsoft Project ile düzenlenmiştir.)

Plan farklı açıdan analiz edildikten sonra ihtiyaca göre tekrar düzenlenebilir. Örneğin eğer proje maliyeti ayrılan toplam bütçeyi geçerse, tekrar bütçeleme yapılır. Piyasada birçok proje planlama ürünü mevcuttur. Bu ürünlerin özelliklerini inceleyerek planı değerlendirirken faydalanılabilecek farklı araçlar tespit edilebilir.

5.13. Plan Kabulü

Proje ekibinin planı kabul etmesi ve benimsemesi yalnız başına yeterli değildir. Proje planı, tamamlanma aşamasında, projenin tüm hissedarlarının onayına sunulmalıdır. Burada yapılacak mutabakat kişilerin inzivalarıyla güvenceye alınır. Proje alınması, sonrasındaki değişikliklerin makul düzeyde olmasına yardım eder.

Ortak Katılım ve Onay

Planın hazırlanması sürecinde proje ile ilgili tüm kişilerin görüş ve önerileri dikkate alınmalıdır. Sürecin sonunda mutabakat kişilerin inzivalarıyla güvenceye alınır. Proje sürecinde tüm ekibin planı kabul etmesi hesap verilebilirlik açısından çok önemlidir.

Bir yazılım projesi ilgili tüm kişiler ortak bir ekip haline gelirse başarılı olur (Nizam, 2011). Ekip olmak için de yapılacaklar üzerinde dolayısıyla planda mutabakat gerekir.

Plan hazırlama aşamasında ekibe planla ilgili fikirleri sorulmalıdır. En önemli soru "*Kişinin kendisine atanan bir görevi hangi süreçte bitirebileceği?*" sorusudur. Bu konuda proje yöneticisi de kendi fikrini söyledikten sonra ortak bir mutabakata varılmalıdır. Görevi yapacak kişiyle ilk anda mutabakat yapmak planı yürütme, izleme ve kontrolde kolaylık sağlar. Kişi, ilk başta görevinin süresini kendisi belirlediği veya süre belirlenirken etkin rol oynadığı durumlarda işi yetiştirmek için daha fazla azim gösterir. Ayrıca sürecin kısa olduğundan şikâyet etme ihtimali de azalır.

5.14. Proje Planına Yazılma Özel İhtiyaçlar

Yazılım projeleri ve planlama alanının ortak özelliklerinden yola çıkarak, örnek yazılım proje planları özetlenmektedir. Plan şablonları, kurumsal proje standartlarına dâhil edilerek, özel bir proje için plan hazırlamak kolaylaşır. Yazılıma özel plan adımları:

- Yazılımın gerçekleştirilecek işe yönelik yapılacakların ayrıntıları
- Yazılımın bileşenleri
- Her yazılımda ortak olan ihtiyaç ve modüllere referanslar
- Yazılım geliştirilmede kullanılacak modelin plana yansıtılması
- Yazılım geliştirme ortamının oluşturulması
- Yazılım kurulumu için yapılacaklar
- Yeni yazılımın mevcut yazılımlarla entegrasyon yöntemleri

Bu kapsamda hazırlanan bir plan örneği ana hatlarıyla aşağıdadır:

Görev İsmi
Analiz aşaması
İçin kapsama özel ihtiyaç analiz
Veri model analizi
Nesne model analizi
Güvenlik model analizi
Nesneler arası etkileşim veya akış modeli
Kullanıcı ara yüzü ihtiyaçları analizi
Raporlama analizi
Sistemler arası entegrasyon analizi
Tasarım aşaması
İçin kapsama özel tasarım
...
Kodlama
...
Test
...
Devreye Alma
Kullanıcı eğitimi
Donanım temini
Altyapı yazılımının temini ve kurulması
Geliştirilen yazılımın kurulması

Projedeki ihtiyaçlara göre yukarıdaki maddelere ilave veya çıkarmalar olabilir. Örnek proje planından faydalanmak, yapılması gereken işlemlerin unutulmamasını da sağlar.

Tüm şablonların hedefi en az olması gerekenleri belirtmektir. Hiçbir şablon değiştirilmeksizin doğrudan kullanılmamalıdır!

Örnek planda tasarım, kodlama ve test aşamasında yapılacaklar analize paralel olduğundan ayrıntılı yazılmamıştır. *İçin kapsama özel ihtiyaç analizi* adımı iş kuraları ve özel algoritmalar gibi ayrıntıları kapsar. Bu adım onlarca maddeye genişletilebilir.

Yukarıdaki plan şelale modeline uygun olarak hazırlanmıştır. Yinelemeli veya çevik geliştirilmenin bir çevrimine de uyulanabilir. Risk, iletişim ve değişim gibi plan türlerinde yapılacaklar eklenmemiştir. Bu tür planlar ilerleyen bölümlerde örneklenmektedir.

5.15. Özet

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 süreç 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.

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

Karşılıklı yazılım ürünlerinin geliştirilme için yürütülen süreçler, yapılacak çalışmalar alt bölümlere ayrılarak daha anlaşılır ve yönetilebilir hale gelir. Planın da bu şekilde ayrıntılı hale getirilmesinde *iş bölünme yapısı* kullanılır. Görevlerin birbiriyle ilişkilerini göstermek için *Gantt Şeması* ve *Ağ Şeması* gibi yöntemler önerilir. İşlerin bölünmesi üzerinden projenin bitiş süresini belirleyen kritik yol, bolluk stresi, 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 görevlere 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.