

YAZILIM PROJESİ GELİŞTİRME DERS NOTLARI

Dr. Öğretim Üyesi Yüksel BAL

YAZILIM PROJELERİNDE FAALİYETLER VE DOKÜMANLAR

Dr. Öğretim Üyesi Yüksel BAL

YAZILIM PROJELERİNDE FAALİYETLER

Yazılım yaşam döngüsü, yazılım geliştirme süresince ortaya çıkan faaliyetlerin ve çıktılarının birbirini izleyen bir sıralama içerisinde belirtilmesidir.

Yazılım yaşam döngüsündeki tüm aşamalarda 2 önemli unsur vardır:

- **Çıktılar (Deliverables);** Çıktılar, yazılım projesinin herhangi bir **aşamasının sonucunda** ortaya çıkarılan; değerlendirme raporları, kaynak kod veya kullanıcı kılavuzu gibi çıktılar olabilir. Genellikle faaliyetler ile çıktılar birbirleriyle yakından ilgilidirler.
- **Milestones;** projenin durumunu anlatmak için kullanılabilen olaylardır. Örneğin kullanıcı kılavuzunun tamamlanması ya da geliştirilen kodlamalardan bir tanesinin (modülün) tamamlanması bir milestone olabilir.

Yönetim için, milestone'lar gereklili olup, milestone'ların tamamlanması yöneticiye yazılım geliştirmenin nasıl ilerlediğini değerlendirme fırsatı verir

YAZILIM PROJELERİNDE FAALİYETLER

A) Yazılım Yaşam Döngüsünde gerçekleştirilen faaliyetler:

1. Fizibilite (Feasibility): Yapılacak geliştirmenin ve masrafların projeyi yapmaya deşip/değmeyeceği belirlenir.

Pazar Analizi (Market Analysis): Geliştirilecek ürün için bir potansiyel Pazar olup / olmadığı belirlenir / araştırılır.

2. Gereksinimler (Requirements): Geliştirilecek yazılımın hangi işlevleri gerçekleştireceği belirlenir.

Gereksinimleri ortaya çıkarma (Requirement Elicitation): Kullanıcı ve müşterilerden gelen talep ve gereksinimler belirlenir.

Etki Alanı Analizi (Domain Analysis): Geliştirilecek olan yazılımı kullanacak yapılar ve görevler belirlenir.

YAZILIM PROJELERİNDE FAALİYETLER

3. Proje Planlama (Project Planning): Yazılımın nasıl geliştirileceğinin tüm planlaması yapılır.

Maliyet Analizi (Cost Analysis): Maliyet kestirimi yapılır

Geliştirme Programı (Scheduling): Geliştirmenin programı / planı yapılır.

Yazılım Kalite Güvencesi (Software Quality Assurance): Ürünün kalitesini garanti edecek faaliyetler belirlenir.

İş Kırılım Yapısı (Work Breakdown Structure): Ürün için geliştirilecek detay alt kırılımlar belirlenir.

4. Tasarım (Design): Yazılımın işlevselliği nasıl sağlayacağı belirlenir

Mimari Tasarım (Architectural Design): Sistemin Yapısı Tasarlanır

YAZILIM PROJELERİNDE FAALİYETLER

Arayüz Tasarımı (Interface Design): Sistemin parçaları / bölümleri arasındaki arayüzler belirlenir

Detaylı Tasarım (Detailed Design): Kişisel kısımlar için algoritmalar tasarlanır

5. Geliştirme (Implementation): Yapısal ya da nesne yönelimli modüler yazılım geliştirilir.

YAZILIM PROJELERİNDE FAALİYETLER

6. Test Etme (Testing): Geliştirilen yazılımın doğru şekilde çalıştığını garanti etmek için data ile birlikte çalıştırılır

Birim / Ünite Testi (Unit Testing): Geliştiricinin (developer) kendisi tarafından yazılımın ana hatlarıyla yapılan testtir.

Bütünleştirme/Birleştirme Testi (Integration Testing): Yazılımın entegrasyonu aşamasında yapılan testtir.

Sistem Test (System Testing): Yazılımın çalışacağı operasyonel ortama benzeyen bir ortamda yazılımın testidir.

Alpha Testing: Müşteri tarafından geliştiricinin ortamında yapılan testtir

Beta Testing: Müşteri tarafından, kendi (müsterinin) ortamında yapılan testtir.

YAZILIM PROJELERİNDE FAALİYETLER

Kabul Testi (Acceptance Testing): Yazılımı satın alacak müşterinin, yazılımı onay testidir

Değişiklik Kontrol Testi (Regression Testing): Canlı ortamda çalışan kodun üzerinde yapılan değişiklıkların control testidir. Regresyon testleri genellikle değişiklikler son aşamaya geldiğinde ve yazılımın yeni sürümü yayınlamadan önce gerçekleştirilir. Regresyon testlerinin öncelikli amacı, uygulamanın kritik alanlarının hala bekendiği gibi çalıştığını kontrol etmektedir.

YAZILIM PROJELERİNDE FAALİYETLER

7. Teslim (Delivery): Müşterinin istekleri doğrultusunda çalışan bir yazılımı müşteriye teslim etmek

Yazılımın Kurulumu (Installation): Müşterinin ortamında yazılımı çalışır duruma getirme

Eğitim (Training): Yazılımı kullanabilmeleri için kullanıcılarla eğitim verme

Yardım Masası (Helpdesk): Kullanıcıların sorularını cevaplamak ve gerekli durumlarda problemlerine çözüm üretilemesini sağlamak

8. Bakım (Maintenance): Devamlı kullanılabilirliği sağlamak için yazılımda güncelleme ve iyileştirme yapmak

YAZILIM PROJELERİNDE DOKÜMANLAR

B) Yazılım Yaşam Döngüsünde hazırlanan standard dokümanlar

- 1. İşin Tanımı (Statement of Work):** Bu doküman, geliştirilecek yazılıma ait ilk müşteri taleplerini ve yine daha sonra müşteri tarafından sık sık ortaya çıkacak olan yeni müşteri taleplerini içerir
- 2. Yazılım Gereksinimleri Şartnamesi (Software Requirements Specification):** Tamamlanan yazılımın hangi işlevleri yerine getireceğini tanımlar
Nesne Modeli (Object Model): Ana nesne ve sınıfları gösterir

YAZILIM PROJELERİNDE DOKÜMANLAR

Kullanım Durumu Senaryoları (Use Case Scenarios): Kullanıcının bakış açısından mümkün olan akışları gösterir

3. Proje Planı (Project Schedule): Yazılım proje planında; yapılacak işlerin sırasını, her bir işin süre tahminini ve iş gücünü belirtir

YAZILIM PROJELERİNDE DOKÜMANLAR

4. Yazılım Test Planı (Software Test Plan): Yazılımın istenen işlevleri yerine getirip / getirmedğini anlayabilmek için nasıl ve nelerin test edilmesi gerektiğini belirtir

Kabul (Onay) Testleri (Acceptance Tests): Sistemin Kabul edilebilirliğini belirleyebilmek için müşteri tarafından tasarlanan test işlemlerine ait dokümandır

5. Yazılım Tasarım (Software Design): Bu doküman da yazılım yapısını tanımlayan diyagramlar ve bilgiler mevcuttur

Mimari Tasarım (Architectural Design): Üst düzeyde yazılımın yapısı ve arabağlantıları belirtiler

YAZILIM PROJELERİNDE DOKÜMANLAR

Detaylı Tasarım (Detailed Design): Daha alt seviyede modüller ve nesnelerin gösterildiği tasarım dokümanıdır

6. Yazılım Kalite Güvence Planı (Software Quality Assurance Plan – SQA Plan): Kaliteyi garanti etmek için gerçekleştirilmesi gereken faaliyetlere ait planların bulunduğu dokümandır

7. Kullanıcı Kılavuzu (User Manual): Tamamlanan yazılımın nasıl kullanılacağını anlatan kullanıcı kılavuzudur

YAZILIM PROJELERİNDE DOKÜMANLAR

- 8. Kaynak Kod (Source Code):** Geliştirilen yazılıma ait kod dosyası
- 9. Test Raporu (Test Report):** Hangi testlerin yapıldığını ve systemin nasıl davranışlığını belirten dokümanlar
- 10. Hata Raporu (Defect Report):** Müşterinin ürünü kullanımı aşamasında karşılaştığı hata, eksiklik ve kusurların belirtildiği dokümanlar

Proje Sonu Raporu

Yazılım projesi geliştirmenin önemli ve kritik olan bir yönü de; hatalardan ders çıkarmak ve projeyi başarılı bir şekilde tamamlamaktır.

Her projede; resmi bir raporun üretilmesi ve dağıtılması da bu durumu her zaman desteklemektedir.

Project Name Project X	Start Date—Sept. 5, 00					Completion Date – Dec 8, 00																											
Management measures	Size <table border="1"> <tr> <th>Estimated</th> <th>Actual</th> </tr> <tr> <td>3000 LOC</td> <td>5000 LOC</td> </tr> </table>					Estimated	Actual	3000 LOC	5000 LOC	Effort <table border="1"> <tr> <th>Estimated</th> <th>Actual</th> </tr> <tr> <td>12,000 min</td> <td>10,000 min</td> </tr> </table>			Estimated	Actual	12,000 min	10,000 min																	
Estimated	Actual																																
3000 LOC	5000 LOC																																
Estimated	Actual																																
12,000 min	10,000 min																																
Subjective comments on estimation	Good Effort was close in total.					Bad Imp effort was underestimated.																											
Subjective comments on process	Good					Bad Team members did not complete asgn on time.																											
Subjective comments on schedule	Good					Bad Not enough time for imp.																											
Quality	Errors found. <table border="1"> <tr> <th>Req</th> <th>Design</th> <th>Unit</th> <th>Integ</th> <th>Postdel</th> </tr> <tr> <td></td> <td></td> <td></td> <td>30</td> <td></td> </tr> </table>					Req	Design	Unit	Integ	Postdel				30		<table border="1"> <thead> <tr> <th></th> <th>ave</th> <th>max</th> </tr> </thead> <tbody> <tr> <td>mccabe</td> <td>4</td> <td>30</td> </tr> <tr> <td>Method/class</td> <td>6</td> <td>10</td> </tr> <tr> <td>Attributes/class</td> <td>10</td> <td>15</td> </tr> <tr> <td>LOC/class</td> <td>150</td> <td>500</td> </tr> </tbody> </table>				ave	max	mccabe	4	30	Method/class	6	10	Attributes/class	10	15	LOC/class	150	500
Req	Design	Unit	Integ	Postdel																													
			30																														
	ave	max																															
mccabe	4	30																															
Method/class	6	10																															
Attributes/class	10	15																															
LOC/class	150	500																															
Subjective comments on quality	Good					Bad System not tested well.																											
Problem: Initial req ambiguity	Description: Format of input file was initially wrong					Impact: 2 weeks wasted																											
Problem:	Description:					Impact:																											
Problem:	Description:					Impact:																											
Problem:	Dr. Üygentim Üyesi Yüksel BAL					Description:																											
Problem:	Description:					Impact:																											

SORU-CEVAP

1) Fazlandırılmış bir yaşam döngüsü modeli yönetime nasıl yardımcı olur?

- Projenin görünülürlüğünü arttırmır,
- Projede, fazlar milestone olarak kullanılıp, proje daha kolay yönetilebilir,
- Daha detaylandırılmış fazlar projede ilerlemenin izlenmesini kolaylaştırır.

2) Proje Planındaki bir milestone'un iki önemli karakteristiği nedir?

- Yazılım geliştirmede bir milestone; ilerlemenin durumunu gösterecek şekilde seçilmelidir, yani milestone'dan ilerlemenin durumunu anlamak mümkün olmalıdır.
- Bir milestone başarı ile tamamlandığında herkes için açık, net ve anlaşılabilir olmalıdır, kafa karışıklığına neden olmamalıdır.

SORU-CEVAP (devam)

3) Aşağıda belirtilen dokümanlar, hangi yazılım geliştirme sürecinde üretilir?

Döküman

- Son kullanıcı kılavuzu
- Mimari tasarım
- Kalite güvence (SQA) planı
- Modül özellikleri,
- Kaynak kod
- İş beyanı
- Test planı
- Ön hazırlık
- Detaylı tasarım
- Maliyet kestirimi
- Proje planı
- Test raporu

Süreç

- Geliştirme
- Tasarım
- Planlama
- Tasarım
- Geliştirme
- Fizibilite Aşaması
- Gereksinim Analizi
- Planlama
- Tasarım
- Planlama
- Planlama
- Test

YAZILIM PROJESİ PLANLAMA

YAZILIM PROJE PLANLAMA

Başarılı bir yazılım projesi geliştirebilmek için projenin tüm detaylarının belirlenmesi, proje planlama çalışmaları sonucunda ortaya çıkar. Proje planlama aşamasında yapılması gereken çalışmalar:

- Proje kaynaklarının belirlenmesi
- Proje maliyetlerinin kestirimi
- Proje ekip yapısının oluşturulması
- Ayrıntılı proje planı yapılması
- Projenin izlenmesi

YAZILIM PROJE PLANLAMA

biçiminde özetlenebilir. Proje planlama aşamasının genel çıktısı olan proje planı, tüm proje süresince sürekli olarak kullanılacak, güncellenecek, gözden geçirilecek bir belgedir. Bu anlamda, planlama aşaması diğer aşamalardan farklıdır.

YAZILIM PROJE PLANLAMA

Proje Kaynakları

Bir yazılım projesi planlanırken, projede kullanılacak kaynaklar dikkatlice ele alınmalıdır. Proje kaynaklarını üç ana başlıkta toplayabiliriz.

- İnsan kaynakları
- Donanım kaynakları
- Yazılım kaynakları

Planlama, bu kaynakların tanımlarını yapar ve zaman kullanımı, görev süreleri, edinilme zamanlarını planlar.

YAZILIM PROJE PLANLAMA

- **İnsan kaynakları**

Bir yazılım projesinde yer alabilecek kişilerin görev tanımları aşağıda tabloda verilmiş olup, belirtilen kaynaklara içerik sağlayıcılar, senaryo uzmanları, çoklu ortam uzmanları ve danışmanlar vs yazılım projesinin içeriğine / konusuna bağlı olarak dahil edilebilir veya bazıları çıkarılabilirler.

Yazılım projesinin boyutuna göre, bu görev tanımlarının tümünde görev yapacak elemanlar, **tam zamanlı** ya da **parça zamanlı** olarak kullanılabilirler.

Planlama; hangi tür elemanın, hangi süre ile ve projenin hangi aşamalarında yer alacağını **belirler.**

YAZILIM PROJE PLANLAMA

Örnek Yazılım Proje Ekibi

Proje Yöneticisi	Kalite Uzmanı
Yazılım Ekip Lideri	Donanım Ekip Lideri
Web Tasarımcısı	Donanım Mühendisi
Proje Sekreteri	Bilgisayar Ağ ve Güvenlik Uzmanı
Gereksinim ve Sistem Çözümleyici	Yazılım Destek Elemanı
Sistem Tasarımcı	Donanım Destek Elemanı
Programcı (lar)	Eğitim Ekip Lideri
Sistem Yöneticisi	Eğitmen
Veri Tabanı Yöneticisi / Uzmanı	Denetleyici
Kalite Güvence Yöneticisi	Çağrı Merkezi Elemanı

Dr. Öğretim Üyesi Yüksel BAL

YAZILIM PROJE PLANLAMA

Donanım kaynakları

Günümüzde donanım sistemleri giderek açık sistem mimarisine dönüşmektedir. Geçmişteki marka bağımlılığı giderek ortadan kalkmaktadır. Yazılım projelerinde kullanılacak donanım kaynakları:

- Ana bilgisayarlar
- Sunucular (Web, Uygulama, Veri tabanı, e-posta vb.)
- Kullanıcı bilgisayarları
- Lokal alan ağı alt yapısı
- Geniş alan ağı alt yapısı

YAZILIM PROJE PLANLAMA

olarak sınıflandırılabilir. Donanım kaynakları planlanırken, yazılımın geliştirileceği ortamın;

- Gerçek kullanım ya da uygulama ortamı dışında bulundurulmasına özen gösterilmelidir.
- Bunun nedeni; geliştirme sırasında ortaya çıkabilecek donanımsal hataların uygulamayı etkilememesini sağlamaktır.
- Diğer taraftan geliştirme ortamı ile uygulama ortamlarının aynı konfigürasyonda olmaları, daha sonra kurulum sırasında ortaya çıkabilecek taşıma sorunlarını büyük ölçüde giderecektir.

YAZILIM PROJE PLANLAMA

- **Yazılım kaynakları**

Günümüzde, yazılım projelerinin geliştirilmesinde kullanılan araç ve yöntemler büyük ölçüde otomatik hale getirilmiş ve bilgisayar destekli olarak kullanılmaktadır. Bu araçlar kısaca aşağıda özetlenmiştir.

Yönetsel Araçlar

Proje planlama ve izleme araçları

Süreç ölçme araçları

Risk yönetim araçları

Çözümleme ve Tasarım Araçları

Çözümleme ve tasarım araçları

Prototipleme ve benzetim araçları

Talep izleme araçları

YAZILIM PROJE PLANLAMA

Gerçekleştirim Araçları

Programlama araçları

Tümleşik yazılım geliştirme ortamları

Arayüz geliştirme araçları

Bakım Araçları

Tersine mühendislik araçları

Yeniden yapılandırma araçları

Destekleyici Araçlar

Belgeleme araçları

Düzenleşim yönetim araçları

İletişim araçları

YAZILIM PROJE PLANLAMA

Araç Seçimi

Belirtilen bu araçların seçiminde çeşitli özellikler belirleyici etmenler olabilirler. Her firma veya kuruma göre değişiklik gösterebilen, öncelikleri değiştirebilen etmenler arasında aşağıdakiler sayılabilir:

- Verimliliğe yararları
- Nitelik iyileştirmeye yararları
- İlgili iş sürecinin işleyişine uygunluğu
- Üretim teknolojisinin yeniliği
- Araçtan yararlanacak proje sayısı
- Kullanıcı sayısı
- Mevcut sistemlerle arayüz olanakları

YAZILIM PROJE PLANLAMA

- Aracın fiyatı (İlk satın alma ve sürüm yükseltme)
 - Üretici firma veya markası
 - Üretici firmanın teknik destek, eğitim ve danışmanlık hizmetleri
 - Aracın güvenilirliği
 - Yeni sürümlerin ortaya çıkış sıklığı ve tutarlılığı
- Geliştirici tarafından en uygun araç seçimi yapılmalı ve olabildiğince uzun süre, değiştirmeden aynı araç kullanılmaya devam edilmelidir.

YAZILIM PROJE PLANLAMA

Proje maliyetleri

Maliyet kestirimi, herhangi bir bilgi sistemi ya da yazılım projesi için gerekli;

- İş gücü
- Zaman

maliyetlerinin, üretimlerinden önce belirlenmeye çalışılması faaliyetleri olarak tanımlanabilir ve bilgi sistemi geliştirme projesinin zamanı ve iş planlaması açısından oldukça önem taşır.

Giderek artan geliştirme maliyetleri, projelerin uzun zaman aralığına yayılması ve bitirilememesi gibi etkenler, sözkonusu maliyetlerin ayrı bir önemle ele alınmasını gerektirmiştir ve bu konuda çeşitli yöntemler geliştirilmiştir.

YAZILIM PROJE PLANLAMA

Maliyet kestirim yöntemleri olarak anılan bu yöntemler;

- Geçmiş projelere ilişkin bilgileri,
- Proje ekibinin deneyimlerini,
- İzlenen geliştirme modeli

gibi unsurları kullanırlar ve duyarlık artttırmak amacıyla geliştirme süreci boyunca birden çok kez uygulanma yoluyla yaşama geçirilirler.

Geliştirme süreci içerisinde maliyet yönetimi önemli bir yer tutar.

Maliyet yönetimi sayesinde;

- Gecikmelerin önlenmesi
- Bilgi sistemi geliştirme sürecinin kolaylaştırılması
- Daha etkin kaynak kullanımının sağlanması

YAZILIM PROJE PLANLAMA

- İş zaman planının etkin olarak gerçekleştirilmesi
- Ürünün sağlıklı olarak fiyatlandırılması
- Ürünün zamanında ve hedeflenen bütçe sınırları içerisinde bitirilmesi sağlanır.

Maliyet yönetimi, genellikle üretim sürecinin değişik aşamalarında bir önceki aşamanın ve daha önce başarılı projelerin gözlemlenen somut bilgileri kullanılarak, ileriye yönelik olarak yapılacak kestirimler kümesi olarak tanımlanır.

Bu nedenle maliyet kestirim yöntemleri, proje yönetiminde önemli yer tutar.

YAZILIM PROJE PLANLAMA

Bir projenin tümü ya da belirli bir kısmı bitirildikten sonra projeye ilişkin bir takım bilgiler, daha sonraki projelerde maliyet kestirimini açısından oldukça önem taşır. Bu bilgilere örnek olarak;

- Projenin toplam süresi
- Projenin toplam maliyeti
- Projede çalışan eleman sayısı, niteliği, çalışma süreleri
- Yazılımın toplam satır sayısı
- Bir satırın ortalama parasal maliyeti
- Bir kişinin gerçekleştirdiği ortalama satır / ay sayısı
- Toplam işlev noktası sayısı
- Bir işlev noktasının ortalama parasal maliyeti

YAZILIM PROJE PLANLAMA

- Bir ayda gerçekleştirilen ortalama işlev noktası sayısı
- Bir kişinin ayda gerçekleştirdiği ortalama işlev noktası sayısı
- Bir kişinin aylık maliyeti verilebilir.

Bu değerler, bitirilmiş projeler için kolayca elde edilebilir ve sonraki projeler için maliyet kestiriminde veri olarak kullanılabilir.

YAZILIM PROJELERİNDE KESTİRİM YÖNTEMLERİ

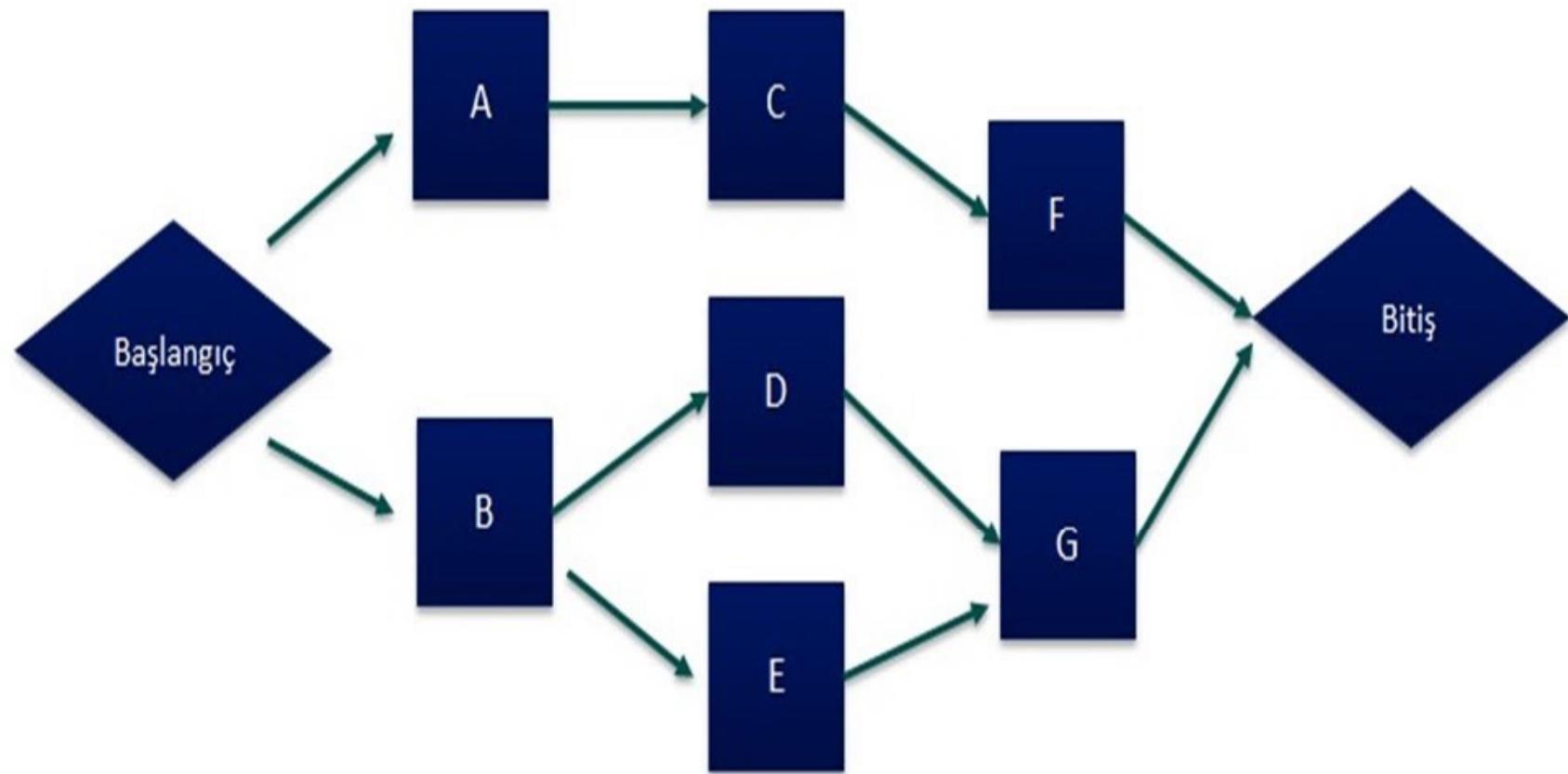
Aktivitelerin Belirlenmesi ve Sıralanması

Bir yazılım projesinde planlama aşamasında öncelikle; gerçekleştirilecek faaliyetlerin belirlenmesi ve tanımlanması, daha sonra bu faaliyetlerin sırası ile (ardışıl veya paralel) birlikte birbirleri arasındaki bağımlılıkların belirlenmesi ilk adımlardan biridir. Daha sonra faaliyetlerin adını, sırasını ve faaliyetlerin birbirlerine olan bağımlılıklarını gösteren ağ diyagramı çizilir. Ağ diyagramı çizilirken;

- İlk ve son aktiviteler hariç, her aktivitenin en az bir öncülü ve en az bir ardılı olmalıdır.
- Aktiviteler arasında, gereksiz bağ yapılmaması tavsiye edilir.
- Faaliyetler arasındaki bağlantılarla, geriye doğru döngü olmamalıdır.

Aktivitelerin Belirlenmesi ve Sıralanması

Ağ diyagramı



Dr. Öğretim Üyesi Yüksel BAL

Aktivite Sürelerinin Tahmin Edilmesi

- ❑ Tahmin yöntemleri:
 - Örneksel tahmin
 - Parametrik tahmin
 - CPM ve PERT
 - Aşağıdan yukarıya tahmin

Efor:

- Bir aktivitenin gerçekleştirilemesi için harcanacak toplam mesai
- 1 adam 1 gün çalışırsa: Efor 1 adam x gün
- 2 adam 1 gün veya 1 adam 2 gün çalışırsa: Efor 2 adam x gün

Süre:

- Bir aktivitenin gerçekleştirilemesi için geçen gün sayısı (5 gün)

Aktivite Sürelerinin Tahmin Edilmesi

Örneksel Tahminleme (Analoji,Yukarıdan-Aşağı Tahminleme)

- Daha önce gerçekleştirilmiş benzer proje verileri ve uzman görüşleri temel alınır,
- Özellikle projenin başında sınırlı veriye sahip olunduğunda kabaca bir maliyet çıkarmak için kullanışlı bir yöntemdir,
- Tahminleme yapılmış projelerde benzer karmaşıklığa ve büyülükle bakılır,
- Tahminlemede bulunulan kişinin konuya ilgili uzmanlığına ve eski projelerin benzerlik oranlarının yüksekliğine bağlı olarak tahminin doğruluğu artar,

Aktivite Sürelerinin Tahmin Edilmesi

- Az maliyetli ve az zaman alan bir yöntem olmakla beraber doğruluk derecesi de düşüktür.
- Diğer yöntemlerle veya tek başına kullanılabilir.

Aktivite Sürelerinin Tahmin Edilmesi

Parametrik Tahminleme:

- Tahminlemeyi yapmak için bazı parametreleri ve bunlar arasındaki ilişkiyi kullanır,
- Daha yüksek kesinlik düzeyine ulaşılabilir.
- Parametrik modellerde genelde kullanılan formül proje karakteristiğine göre değiştirilmektedir.
- Modelin doğruluğu ve tutarlılığı analiz yapan çalışanın bu parametreleri ortaya çıkarma yeterliliği ile doğru orantılı olarak değişir,

Aktivite Sürelerinin Tahmin Edilmesi

Aşağıdan yukarıya tahminleme:

- Bütünü oluşturan parçaların maliyetlerinin tahmin edilmesi ve bu maliyetlerin toplanarak parçaların oluşturduğu üst seviye parçanın maliyetinin hesaplanmasına dayanır,
- Parçaların tahminlemesinin doğru yapılması halinde bütününe maliyetinin tahmininde daha doğru sonuçlar verir,

CPM ve PERT Metodları ile Tahminleme:

Detayları sonraki slaytlarda

CPM ve PERT METODLARI

Dr. Öğretim Üyesi Yüksel BAL

CPM/PERT ile neler belirlenir?

- Proje ne zaman bitecek?
- Projedeki kritik faaliyetler hangileri?
- Kritik olmayan faaliyetler hangileri?
- Belirli bir zamanda projenin bitme olasılığı nedir?
- Proje, plana göre yürüyor mu? Planın önünde mi?
- Planın gerisinde mi?
- Proje bütçenin üzerinde mi? Altında mı?
- Projeyi zamanında bitirebilmek için yeterli kaynak var mı?
- Eğer proje planlanandan önce bitirilmek isteniyorsa bu en az maliyet ile nasıl yapılabilir?

CPM ve PERT Yöntemleri

CPM: Critical Path Method - Kritik Yol Metodu

PERT: Program Evaluation & Review Technique

Program Değerlendirme ve Gözden Geçirme Tekniği

- CPM ve PERT birçok faaliyet içeren büyük ve karmaşık projelerin planlanması, çizelgelenmesi ve kontrolüne yardımcı olmak için tasarlanmış şebeke asıllı modellerdir.
- CPM ve PERT'in amacı; çizelgeleme faaliyetlerine analitik anlamlar kazandırmaktadır.
- CPM; faaliyet sürelerini deterministik (önceden belirlenmiş ve kesin) olarak kabul ederken, PERT; bu sürelerin olasılıklı olduğunu kabul etmektedir.

CPM ve PERT Yöntemleri

- Diğer bir deyişle; **faaliyetlerin süreleri kesin olarak biliniyorsa** projenin tümünün bitirilmesi için gerekli süre Kritik Yol Yöntemi (CPM - Critical Path Method) ile belirlenebilir.
- CPM; faaliyetlerin proje toplam süresini uzatmadan ne kadar ertelenebileceğini de bulmak için kullanılabilir.
- Eğer faaliyetlerin süreleri kesin olarak bilinmiyorsa proje için belirlenmiş bir teslim zamanında bitirme olasılığını bulmak için Program Değerlendirme ve Gözden Geçirme Tekniği (**PERT**) kullanılabilir.

CPM METODU

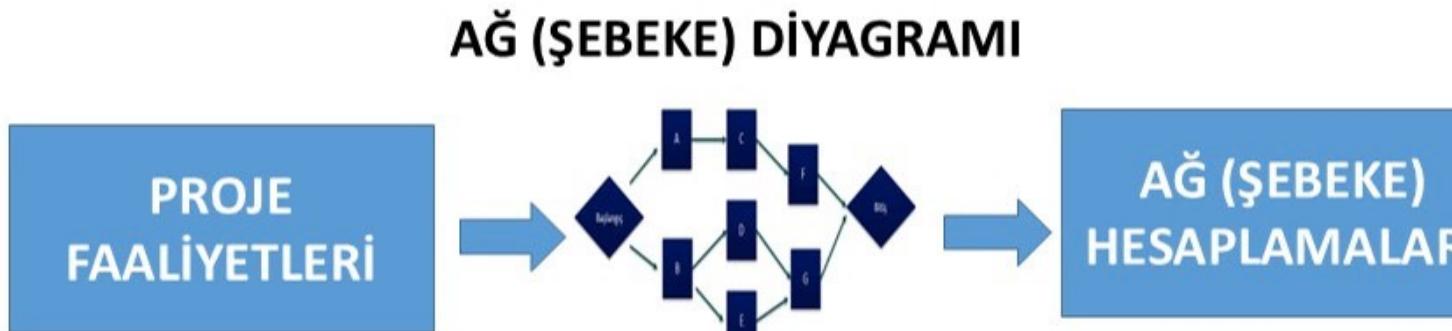
Dr. Öğretim Üyesi Yüksel BAL

CPM METODU

Kritik Yol Metodu (CPM)

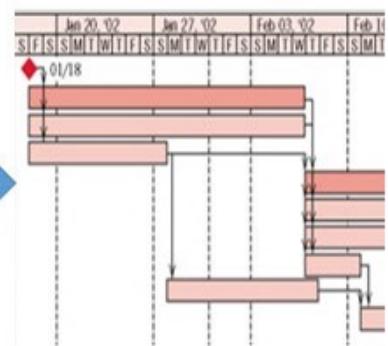
- İleriye doğru hesaplama,
- Geriye doru hesaplama,
- En erken başlama, en erken bitme zamanları,
- En geç başlama, en geç bitme zamanları,
- Serbestlik sürelerinin (bolluk) hesaplanması,

CPM METODU



- FAALİYETLER,
- FAALİYET SÜRELERİ
(SÜRE BELLİ DEĞİLSE HESAPLANMALI)
- FAALİYET İLİŞKİLERİ
(BAĞIMLILIKLAR BELİRTİLMELİ)

ZAMAN PLANI (PROJE PLANI)



CPM – HESAPLAMALARI

- 1) Projenin tamamlanması için gereken süre,
- 2) Proje faaliyetlerinin kritik ve kritik olmayan faaliyetler olarak sınıflandırılması,
- 3) Her bir faaliyetin esneklik sürelerinin belirlenmesi

Notlar:

- Başlangıç ve bitiş zamanı arasında esneklik (gevşeklik) olmayan faaliyetlere **kritik faaliyetler** denir.
- Projenin zamanında bitirilebilmesi için her kritik faaliyetin zamanında başlayıp, bitirilmesi gerekmektedir.

CPM – HESAPLAMALARI

- Kritik olmayan faaliyetlerde bazı çizelgeleme gevşemelerine izin verilir. Gevşeklik zamanı sınırları içerisinde kalmak kaydıyla bu faaliyetin başlangıç zamanı farklı olabilir, bu durum tüm proje süresini etkilemez.
- Şebekede (Ağ'da) düğümlere "**olay**" da denir, olay; gerekli hesaplamaları yapabilmek için bazı faaliyetlerin bitip başka faaliyetlerin başladığı noktalardır.

ÖZET

- Kritik faaliyetler zamanında başlamalı ve bitmelidir, aksi halde projenin süresi uzayacaktır.
- Kritik olmayan faaliyetlerin mümkün olduğunca erken başlanması tercih edilmelidir, aksi halde süre bolluklarını önceden harcamamız daha sonra problem olarak karşımıza çıkacaktır,
- Bir faaliyeti en erken başlama zamanından geciktirmek gerekebilir, Örneğin; E ve F faaliyetlerinin her birinde aynı kaynağa ihtiyaç olabilir ve bu kaynak tek olabilir. E ve F mümkün olduğunca erken çizelgelenip kaynağın kullanımı ile ilgili alınacak önlemde (zamanlama) önceden belirlenmelidir.

CPM UYGULAMA-1

Dr. Öğretim Üyesi Yüksel BAL

ÖRNEK

Bir yazılım projesinde, aşağıdaki tabloda belirtildiği gibi, 8 ayrı parçadan oluşan geliştirme çalışmalarının tamamlanması gerekmektedir. Her bir iş için tamamlanma zamanları ve gerçekleştirilen faaliyetlerin kendi içerisinde bağımlılıkları da tabloda verilmiştir.

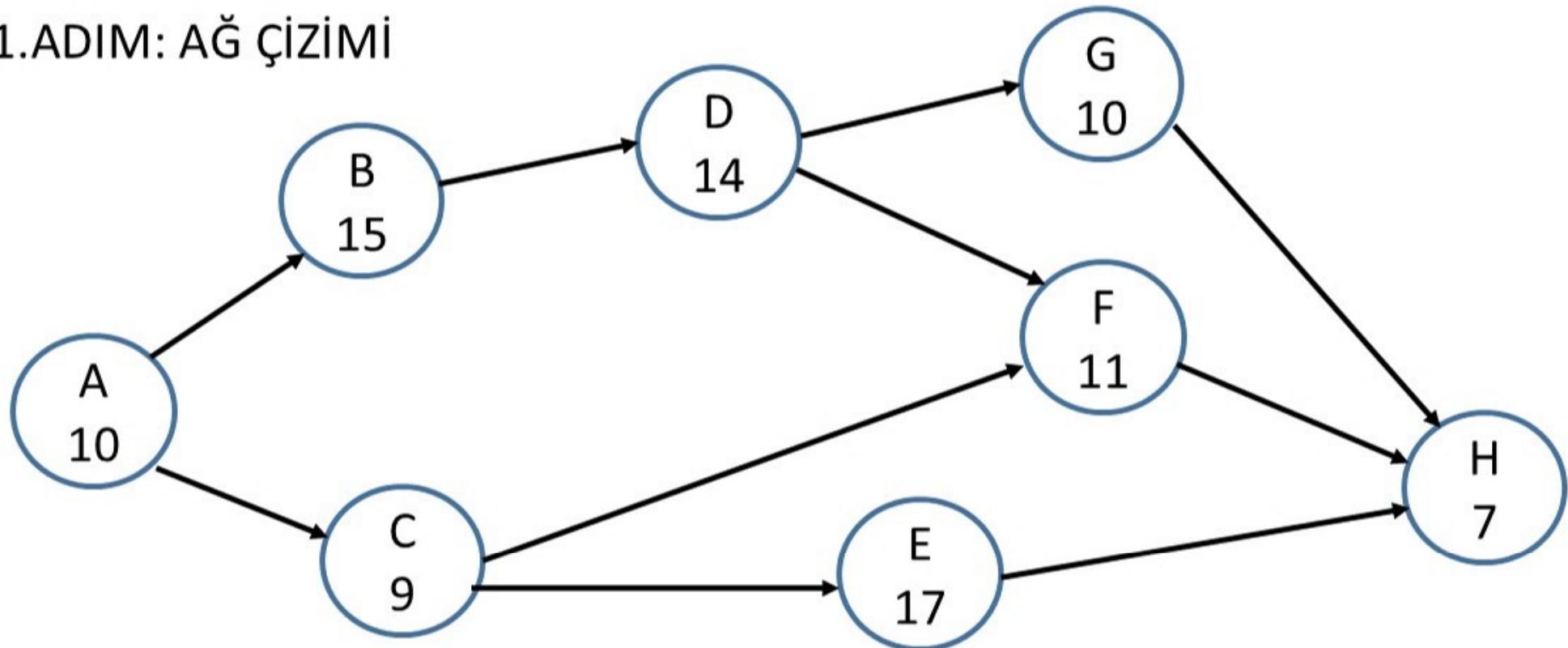
Bu proje için;

- CPM metodу ile kritik yolu/yolları,
 - Projenin tamamlanma süresini,
 - C ve D parçalarından herhangi birinin 2 haftalık geçikmesi, projede gecikmeye neden olur mu?
- belirleyiniz.

FAALİYET ADI	TAMAMLANMA SÜRESİ (HAFTA)	BAĞIMLILIKLAR (DEPENDENCIES)
A	10	-
B	15	A
C	9	A
D	14	B
E	17	C
F	11	C,D
G	10	D
H	7	E,F,G

ÖRNEK

1.ADIM: AĞ ÇİZİMİ



ÖRNEK

2.ADIM: İLK FAALİYETTEN BAŞLAYARAK İLERİ YÖNDE EB VE ES SÜRELERİ HESAPLANIR

A bitiş süresi,

B başlangıç süresi
olmaktadır.

A ilk faaliyet,
başlangıç = 0

0 | 10

A | 10

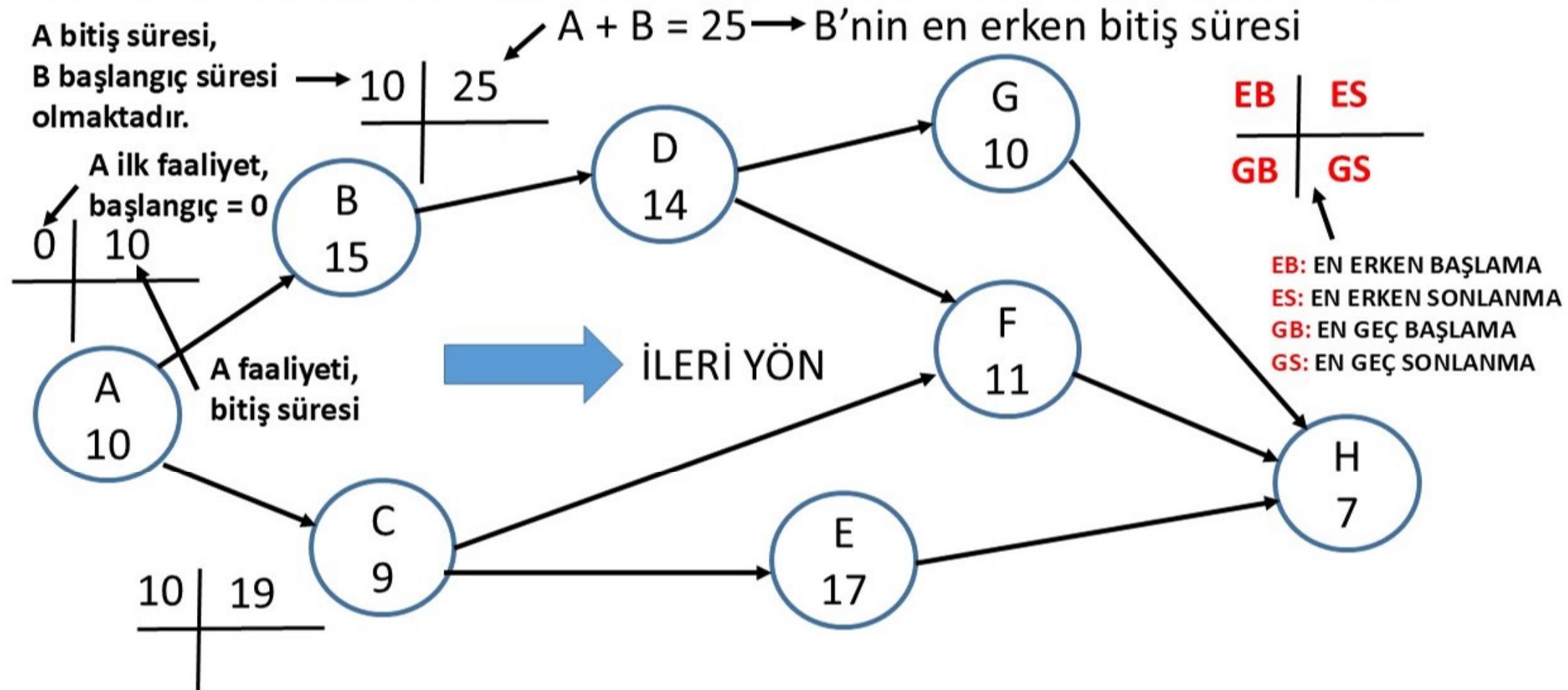
A 10

A faaliyeti,
bitiş süresi

10 | 19

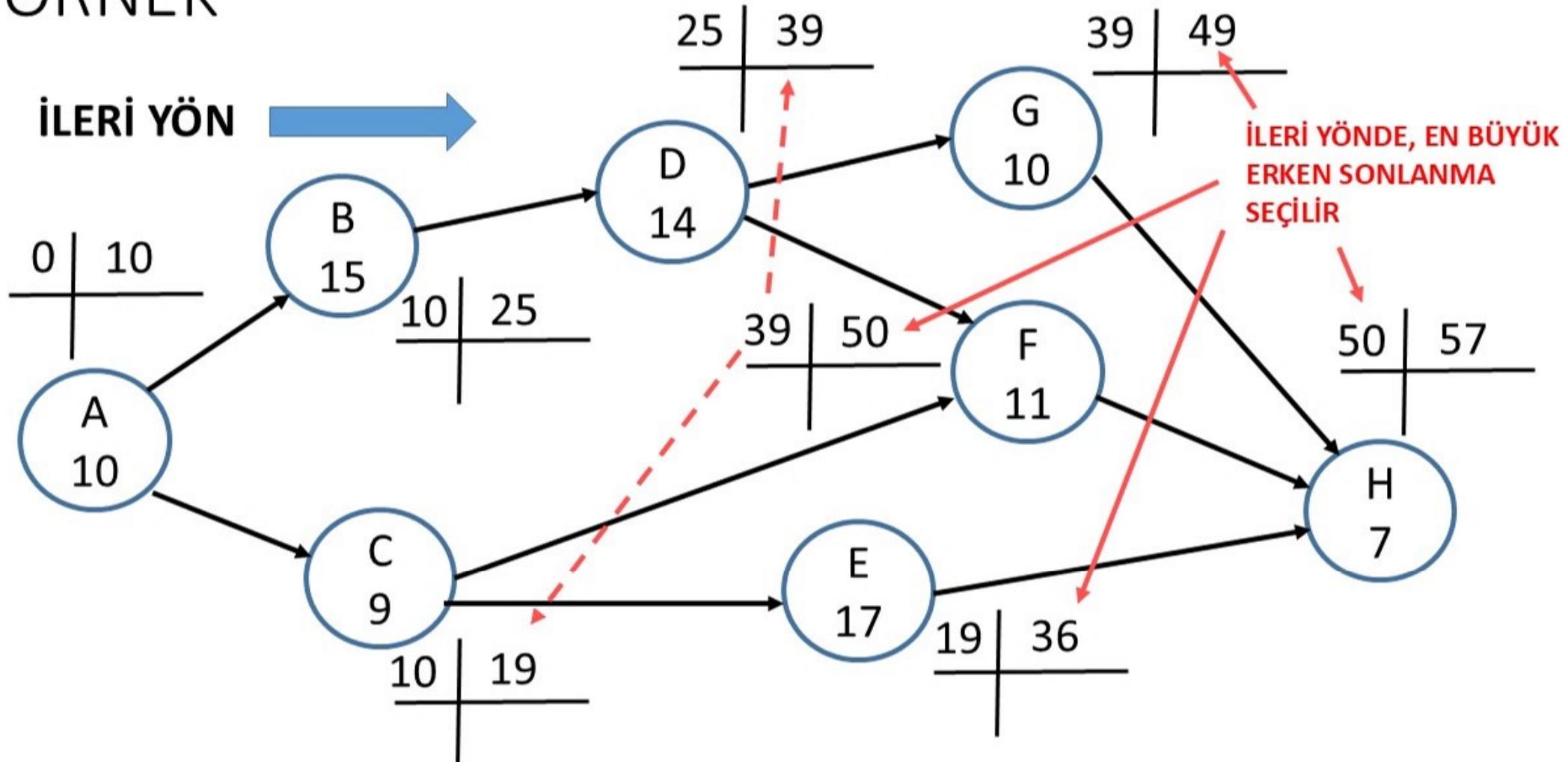
10 | 19

10 | 19



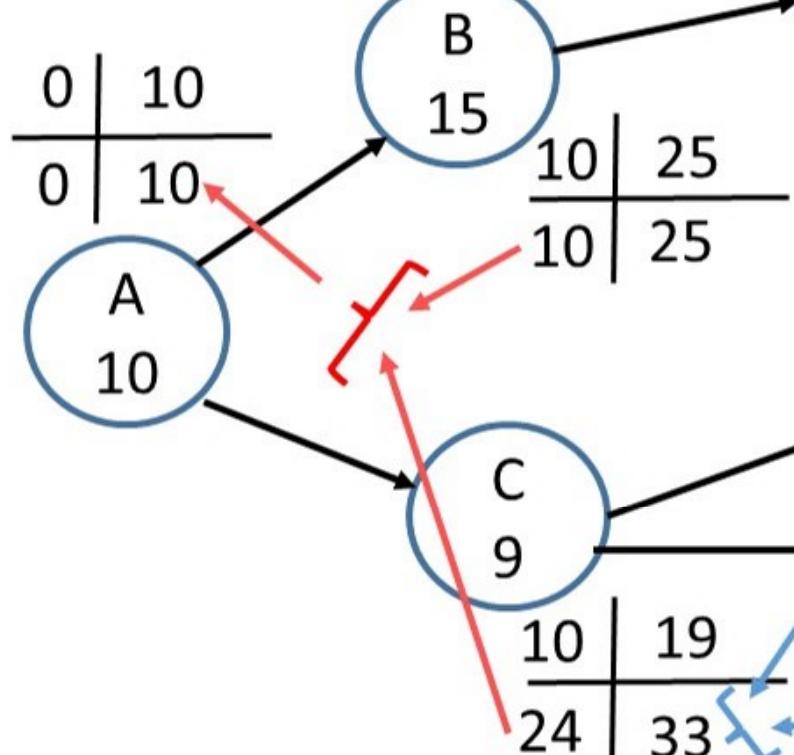
ÖRNEK

İLERİ YÖN

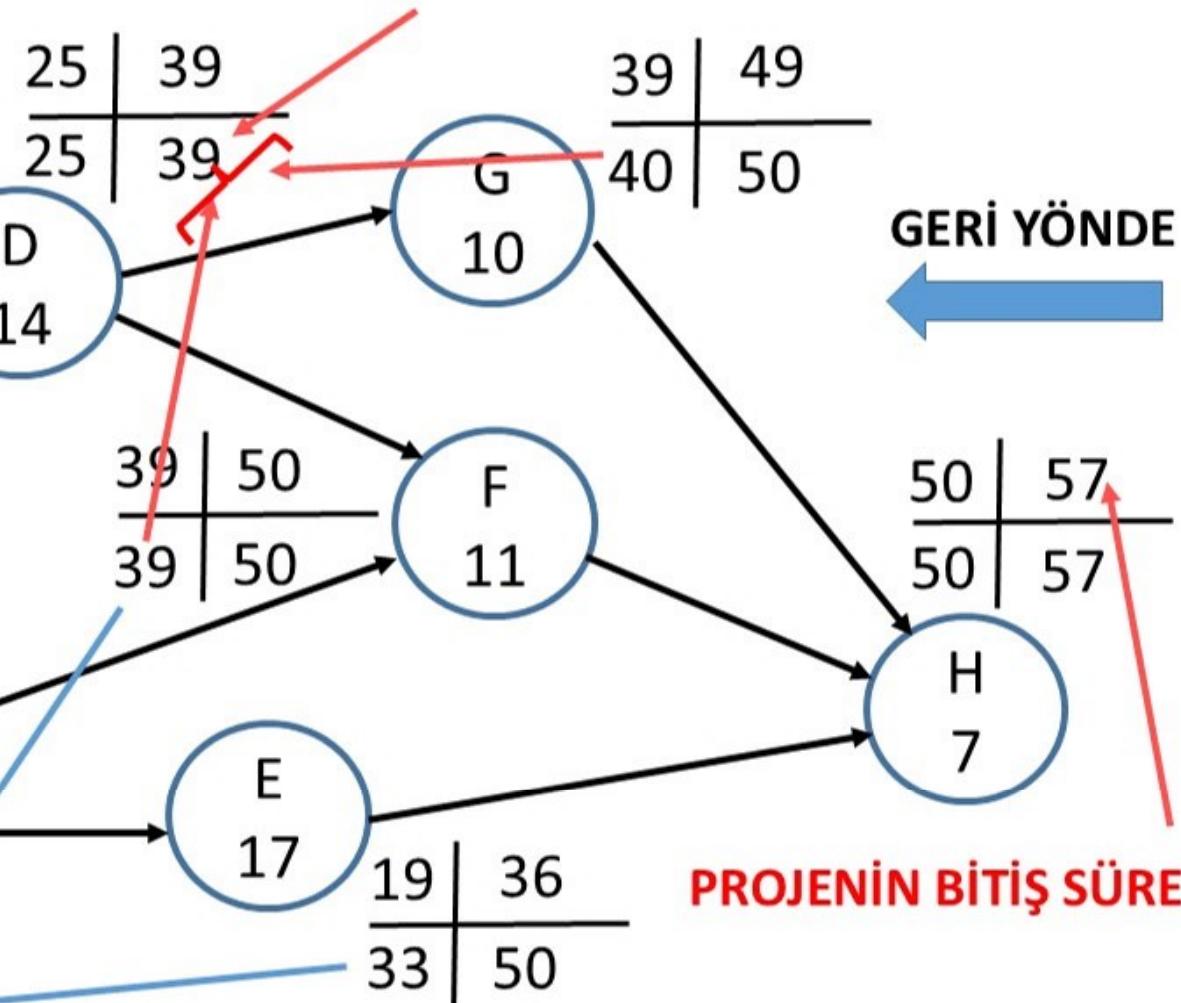


ÖRNEK

3.ADIM: SON FAALİYETTEN BAŞLAYARAK GERİ YÖNDE GS VE GB SÜRELERİ HESAPLANIR

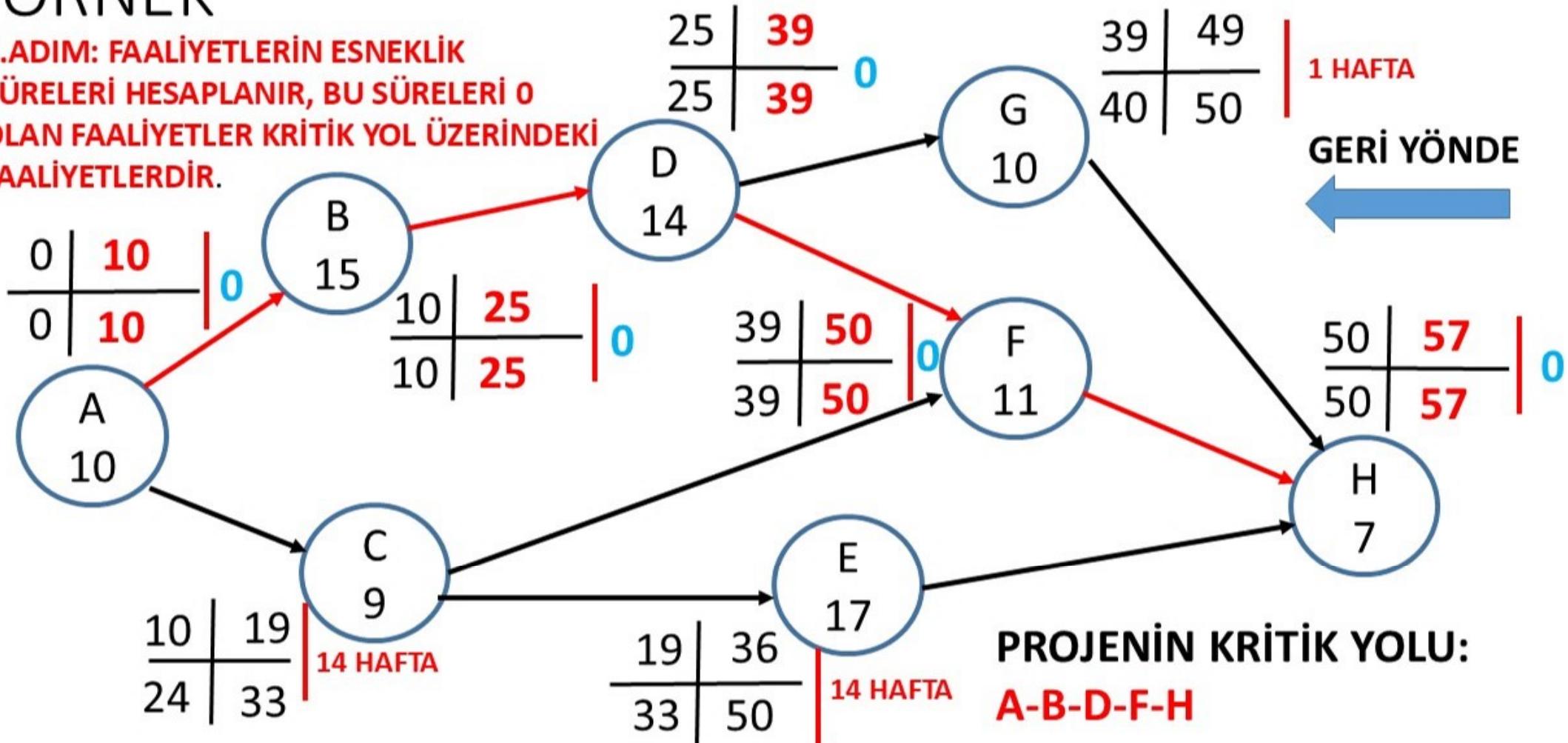


GERİ YÖNDE EN GEÇ BAŞLAMALARDAN EN KÜCÜK OLAN SEÇİLİR



ÖRNEK

4.ADIM: FAALİYETLERİN ESNEKLİK SÜRELERİ HESAPLANIR, BU SÜRELERİ O OLAN FAALİYETLER KRİTİK YOL ÜZERİNDEKİ FAALİYETLERDİR.



ÖRNEK

Soruda C ve D faaliyetlerinden herhangi birinde 2 haftalık gecikmenin proje süresini uzatıp uzatmayacağı sorulmuştu.

Projenin kritik yolu: A-B-D-F-H idi.

- C faaliyeti kritik yol üzerinde olmadığından 2 haftalık gecikme proje süresini etkilemez. C faaliyetinin gevşek zamanı 14 haftaya kadar olup, yeterince gevşek zamana sahiptir. Ancak gecikme, 14 haftayı aşarsa 14 haftadan sonraki geciken her bir hafta kadar proje süresi uzar.
- D faaliyeti kritik yol üzerinde olup, doğrudan proje süresinin gecikme süresi kadar uzamasına neden olacaktır.

Kaynaklar

- David Gustafson, ‘Software Engineering’
- M. Erhan Sarıdoğan, ‘Yazılım Mühendisliği’,
- Ali Arifoğlu, Ali Doğru, ‘Yazılım Mühendisliği’
- Oya Kalıpsız, Ayşe Buharalı, Ayşe Biricik, ‘Sistem Analizi ve Tasarımı’
- Yüksel Bal, ‘Yazılım Projesi Geliştirme’ Ders Notları,
- Yüksel Bal, ‘Yazılım Mühendisliği ve Sistem Analizi’ Ders Notları,
- Çeşitli İnternet Kaynakları,