

سناریو های مختلف برای توسعه یک طرح پروژه:

مجموعه ای از نیازمندی ها و محدودیتها (روی منابع و بودجه ... )، آیا قابل انجام هست؟

ویژگی های محصول و صفات کیفی رو به ما میدن، تخمین میزنیم چه بودجه و منابعی لازم

مدت انجام پروژه رو میدن و پول، چه محصولی میتونیم تولید کنیم

فارغ از سناریو باید طرح پروژه ای ارائه کنیم که بین

نیازمندی

زمان

پول

منابع

تکنولوژی

سازگار باشد و در صورت تغییر نایزمندی و بقیه فاکتورها باید تعادل بماند

بعضی مواقع پلن به ما دیکته می شود ،

نیازمندی را به ما میدن و باید در زمان و بودجه برنامه رو بسازیم

یا تغییراتی انجام بدیم

پلن های دیکته شده زیاد شکست میخورن

برنامه ریزی موج غلتان

غیر ممکنه که ابتدا طرح پروژه کامل ارائه کنیم (به دلیل چیزهای نامعلوم)

برای ماه آتی پلن جزئی ارائه میکنیم و برای ماه های بعدی پلن کلی تر

استراتژی پلن پروژه

اولین گام بدون محدودیت

سپس محدودیت هارا درنظر میگیریم

ابزار های پلنینگ

Architecture decomposition view

Work breakdown structures

Work packages

Activity networks

Cpm pert gantt

Resource loading histograms

فعالیت های اولیه

توسعه یک دید تجزیه معماری

توسعه ساختار تفکیک کار

تعریف بسته های کاری wbs

تعریف زمان بندی با milestone

شبکه زمانبندی

Pert

تعداد و نوع منابع

تخمین تلاش هزینه منابع و ...

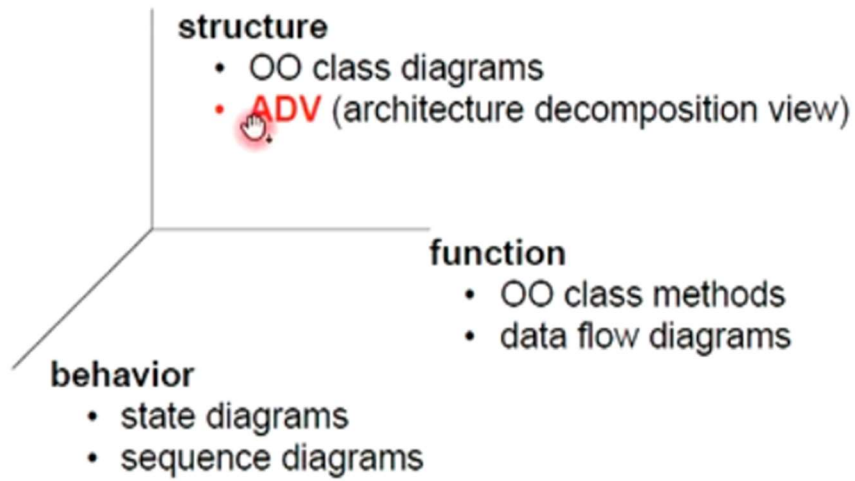
مذاکره با مشتری

سه زاویه معماری نرم افزار

دید ساخت یافته ( کلاس دیاگرام و Adv

دید کاربردی ( متد های کلاس و دیاگرام Dataflow)

دید کاربردی (State diagram و sequence diagram)



the ADV is the project manager's view of a product

مثال: adv برای atm

درایور سخت افزاری

قسمت تراکنش مالی

پشتیبانی

ارتباطات

خود قسمت تراکنش به 4 قسمت

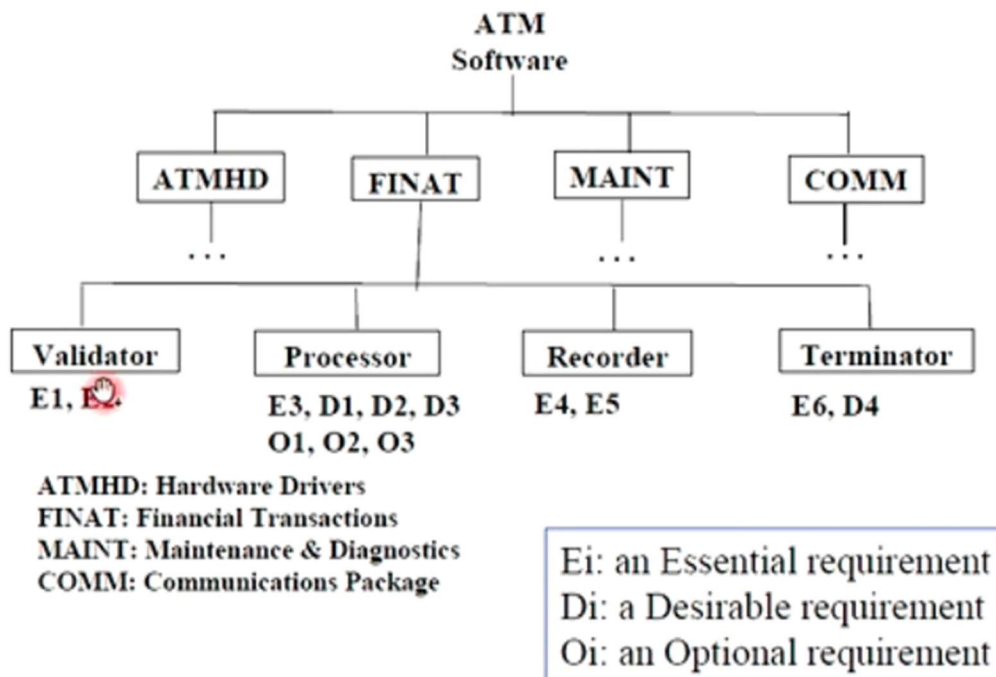
Validator

Processor

Recorder

Terminator

تقسیم میشود



## Outline Representation of the Partial ADV

ATM Software

1. ATMHD

2. FINAT

2.1. Validator [E1, E2]

2.2. Processor [E3, D1, D2, D3, O1, O2, O3]

2.3. Recorder [E4, E5]

2.4. Terminator [E6, D4]

3. MAINT

4. COMM

حداکثر 7 بخش در هر سطر

حداکثر 4 یا 5 سطح


شماره گذاری decimal

نیازمندی هارا به هر کدام از المان ها تخصیص بدیم

سپس به تخصیص کار به افراد میرسیم

اگر محصول پیچیده است از چند Adv استفاده کنیم

## Traceability of Functional Requirements

	Validator	Processo	Reorde	Terminator
E1	X	r	r	
E2	X			
E3		X		
E4			X	
E5			X	
E6				X
D1		X		
D2		X		
D3		X		
D4				X
O1		X		
O2		X		
O3		X		

- Note that each functional requirement is allocated to one, and only one, element of the FINAT ADV

تخصیص ویژگی های کیفی

ممکن است به یک المان چند یا همه المان ها مربوط باشد

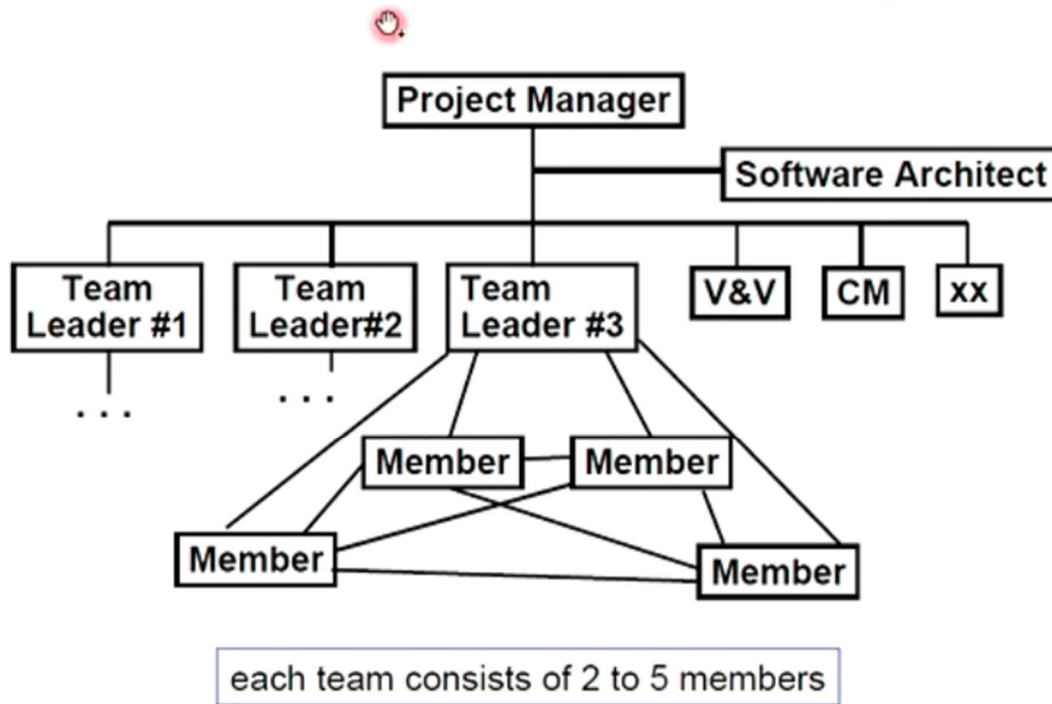
قانون Conway

ساختار یک سیستم منجر با ساختار تیم میسود

Fairley

سلسله مراتب نرم افزار باید به شکلی باشد که تخصیص کار ها را نمایش دهد

## A Structural Model for Software Projects



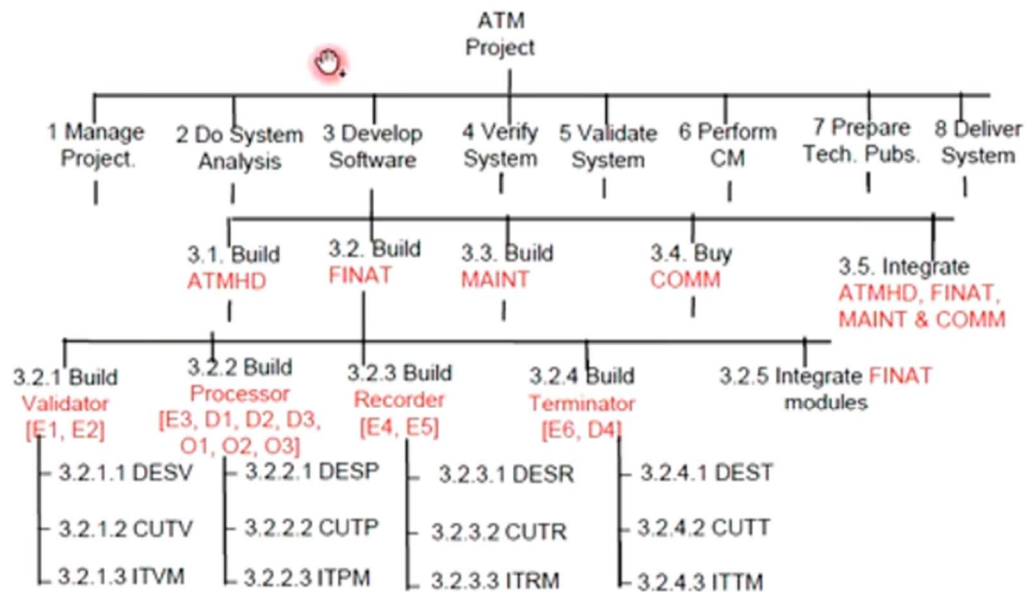
Wbs

تجزیه سلسله مراتبی فعالیت هاست

نمایش با درخت یا لیست indented

تقسیم فعالیت ها به بخش های جدا نیازمندی ها و کار ها به گروه ها و افراد مختلف تخصیص میشود

## Graphical Form of a Partial WBS with Embedded ADV and Allocated Requirements



DESx: detailed design of module x; CUTx: coding & unit testing x; ITxC: integration and testing of x

## Outline Form of the Partial WBS

- 1 Manage Project
- 2 Do System Analysis
- 3 Develop Software
  - 3.1 Build ATM Hardware Drivers
  - 3.2 Build Financial Transaction Handler
    - 3.2.1 Build Validator [E1, E2]
      - 3.2.1.1 Design Validator
      - 3.2.1.2 Code & Unit Test Validator
      - 3.2.1.3 Integrate & Test Validator
    - 3.2.2 Build Transaction Processor [E3, S1, D2, D3, O1, O2, O3]
      - 3.2.2.1 Design Transaction Processor
      - 3.2.2.2 Code & Unit Test Transaction Processor
      - 3.2.2.3 Integrate & Test Processor Components
    - 3.2.3 Build Recorder [E4, E5]
      - 3.2.3.1 Design Recorder
      - 3.2.3.2 Code & Unit Test Recorder
      - 3.2.3.3 Integrate & Test Recorder Module
    - 3.2.4 Build Terminator [E6, D4]
      - 3.2.4.1 Design Recorder
      - 3.2.4.2 Code & Unit Test Recorder
      - 3.2.4.3 Integrate & Test Recorder Module
  - 3.3 Build Maintenance & Diagnostic Module
  - 3.4 Buy the Communications Package
  - 3.5 Integrate ATMHD, FINAT, MAINT, and COMM modules
- 4 Verify System
- 5 Validate System
- 6 Perform CM
- 7 Prepare Technical Publications
- 8 Deliver System



## WBS Elements for “Manage Project”

### 1 Manage Project



#### 1.1 Initiate Project

- 1.1.1 Identify stakeholders
- 1.1.2 Develop/clarify requirements
- 1.1.3 Prepare initial estimates
- 1.1.4 Prepare initial project plan
- 1.1.5 Obtain commitment to the plan

#### 1.2 Conduct Project

- 1.2.1 Measure and control project
- 1.2.2 Lead and direct personnel
- 1.2.3 Communicate and coordinate
- 1.2.4 Manage risk

#### 1.3 Closeout Project

- 1.3.1 Obtain product acceptance
- 1.3.2 Conduct post-mortem sessions
- 1.3.3 Prepare and distribute lessons-learned report
- 1.3.4 Assist in reassigning project personnel

1 and Learning Software Projects

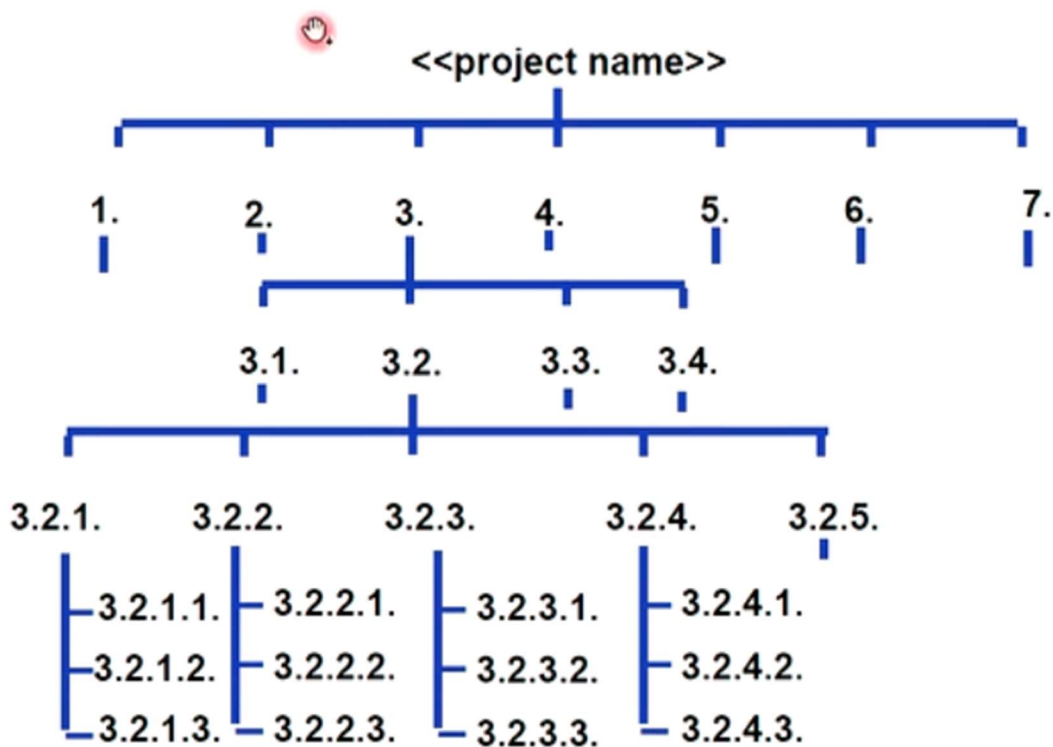
rhaz

نکات

Adv اسم دارد ولی wbs فعل دارد چون ناظر به انجام کار است

کار های adv در wbs تعبیه شده است

## Structure & Numbering of a Partial WBS



شکستن کارها تا کجا ادامه دارد

تا جایی که پیچیدگی ها آشکار شود.

فرصت برای استفاده از Component های موجود فراهم شود

نیاز های سخت افزاری مشخص شود

بتوانیم تخمین زمان و هزینه بزنیم

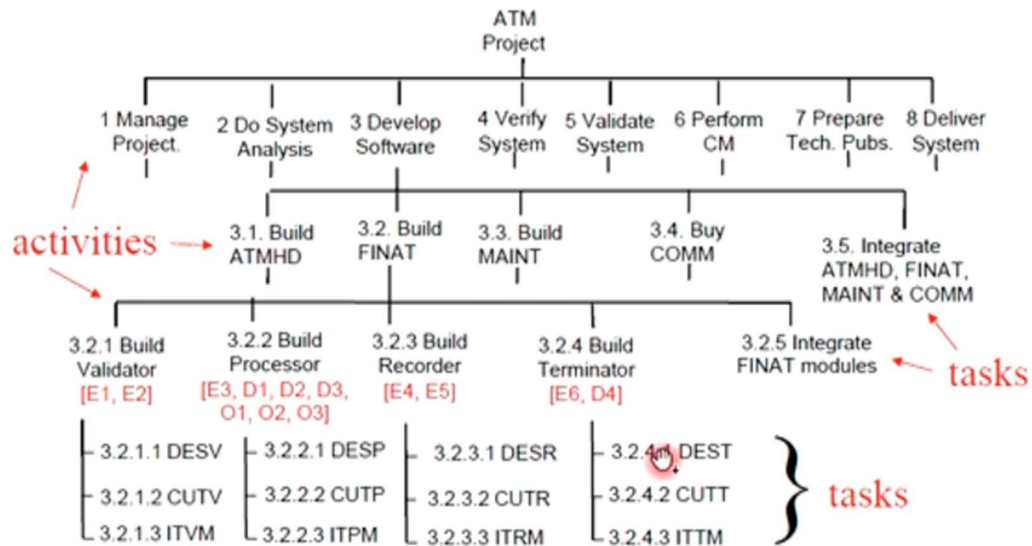
پکیج های کاری

کوچکترین واحد مدیریت و کنترل task نام دارد

مراحل بالاتر که تجمعی از task ها هستند activity نام دارند

پکیج های کاری برای مستندسازی task ها به کار می رود (و قاعدتا برای activity)

## Partial WBS for an ATM Project



Work package حاوی چیست

شماره و نام متناظر در Wbs

خلاصه از task

زمان

منابع

Task قبلی و بعدی

محصولات ( و محصولات Version control )

عوامل ریسک

معیارهای پذیرش عینی

## A Work Package Template and Example

<i>Task identifier:</i>	3.2.2.1 Design Transaction Processor
<i>Task description:</i>	Specify internal architecture of the Transaction Processor module
<i>Estimated duration:</i>	2 weeks
<i>Resources needed:</i>	
<i>Personnel:</i>	2 senior telecom designers
<i>Skills:</i>	Designers must know UML
<i>Tools:</i>	One workstation running Rapsody
<i>Travel:</i>	3 day Design Review in San Diego for 2 people
<i>Predecessor tasks:</i>	3.2.1 - Develop system architecture
<i>Successor tasks:</i>	3.3.2.2 - Implement Transaction Processor
<i>Work products:</i>	Architectural specification for Transaction Processor Test Plan for Transaction Processor
<i>Baselines created:</i>	Architectural Specification and Test Plan
<i>Risk factors:</i>	Designers not identified
<i>Acceptance criteria:</i>	Successful design inspection by peers and approval of Transaction Processor design by the Software Architect

چه کسی Work package را آماده میکند

در ابتدا تیم برنامه ریز

اما در طی انجام پروژه رهبر هر تیم

رهبر وظیفه دارد بر اساس موج غلطان پکیج را آماده میکند و همچنین مسئول کیفیت و گزارش دهی نیز هست

در پروژه های کوچک معمار نرم افزار و مدیر پروژه و رهبر تیم یکی هستند

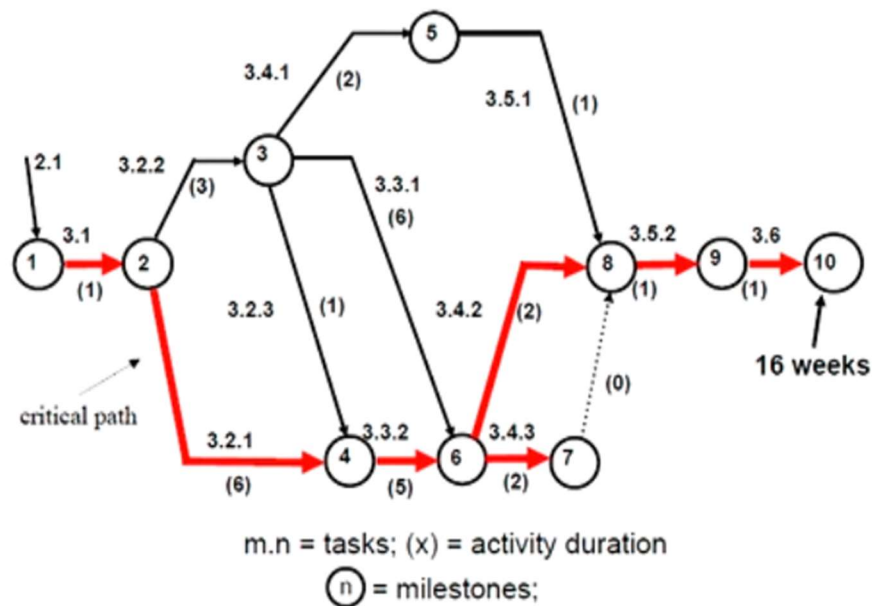
## A WBS Task List

<u>Task #</u>	<u>Description</u>	<u>Predecessors</u>	<u>Duration</u>	<u>#Staff</u>
2.1	Receive approval to proceed	--	--	--
3.1	Analyze requirements	2.1	1	2
3.2	Design			
3.2.1	Redesign existing components	3.1	6	4
3.2.2	Design new components	3.1	3	1
3.2.3	Design interfaces	3.2.2	1	2
3.3	Implement			
3.3.1	Implement new code	3.2.2	6	2
3.3.2	Modify existing code	3.2.1, 3.2.3	5	1
3.4	Finish implementation			
3.4.1	Develop integration plan	3.2.2	2	2
3.4.2	Finish unit testing	3.3.1, 3.3.2	2	2
3.4.3	Update documentation	3.3.1, 3.3.2	2	3
3.5	Integrate and test			
3.5.1	Develop integration tests	3.4.1	1	3
3.5.2	Perform integration tests	3.4.2&3, 3.5.1	1	2
3.6	Perform acceptance tests	3.5.2	1	1

نمودار cpm

شبکه زمانبندی

## A Critical-Path Schedule Network Generated from the Task List



نکات

وظایف (Task) پیکان ها هستند

گره ها milestone هستند

در ابتدا باید قاعده ی پیش نیازی رعایت شود


در گره 7 یال dominance گفته میشود و دیگر زمانی نمیبرد

Slack time

مثلا در 3.2.2 و 3.2.3 5 واحد زمان میبرد و در مسیر بحرانی 7 واحد، یعنی در آن مسیر بالا 3 هفته میتوان تاخیر کرد که به آن زمان استراحت slack time گفته میشود

لیست milestone ها

## Milestone Event List

<u>Event</u>	 <u>Description</u>
1	Project initiation
2	Requirements analysis completed
3	Design of new components completed
4	Existing components redesigned Interfaces to new components designed
5	Integration plan completed
6	New code implemented Existing code modified
7	Documentation updated
8	Unit testing completed Documentation updated Integration tests ready
9	Integration tests completed
10	Acceptance tests completed

achievement of a milestone requires tangible  
evidence of work products completed

## Critical-Path Milestone Chart

<u>critical milestone</u>	<u>elapsed time</u>	<u>cumulative time</u>
3.1	1 week	1 week
3.2	6 weeks	7 weeks
3.7	5 weeks	12 weeks
3.8, 3.9	2 weeks	14 weeks
3.11	1 week	15 weeks
3.12	1 week	16 weeks

note: some weekly milestones should be inserted  
between 3.1 & 3.2 and between 3.2 & 3.7

مشاهدات

Wbs درختی است و رابطه کل به جز نمایش میدهد، تجمیعی سلسله مراتبی است و توالی ندارد

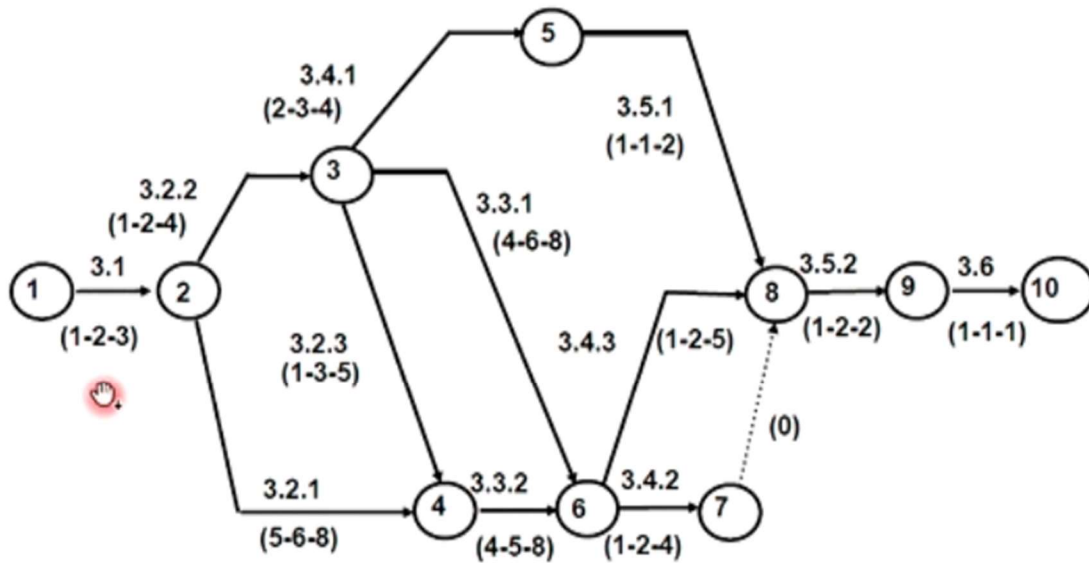
نمودار زمانبندی رابطه پیش نیازی و ترتیب زمانی است سلسله مراتبی نیست و توالی دارد

هر دو لازم هستند

نمودار pert



## A PERT Network

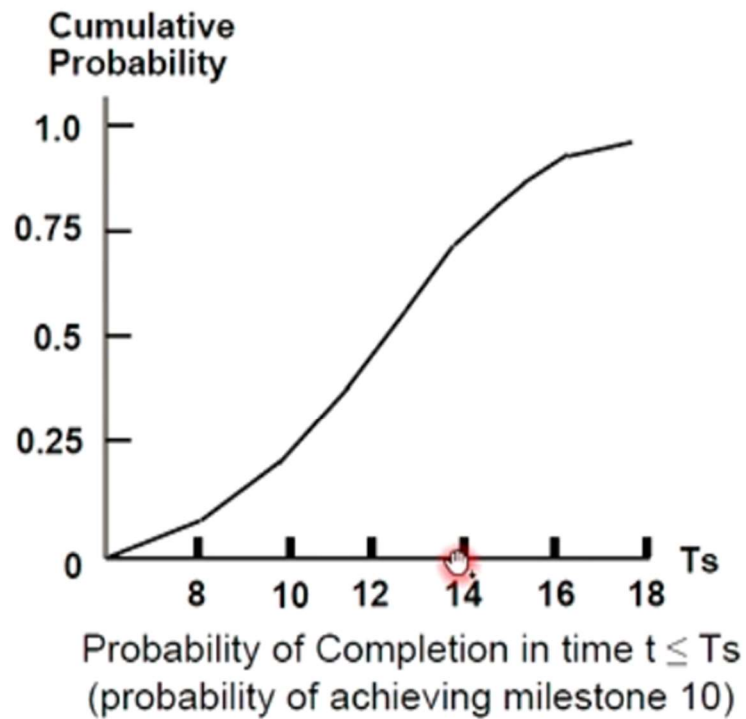


m.n = activities;  $\textcircled{n}$  = milestones;  
(a-m-b) = activity duration estimates

فرقش با cpm اینه که برای task ها سه زمان کم متوسط و زیاد در نظر میگیرد

همچنین یک نمودار احتمال داریم

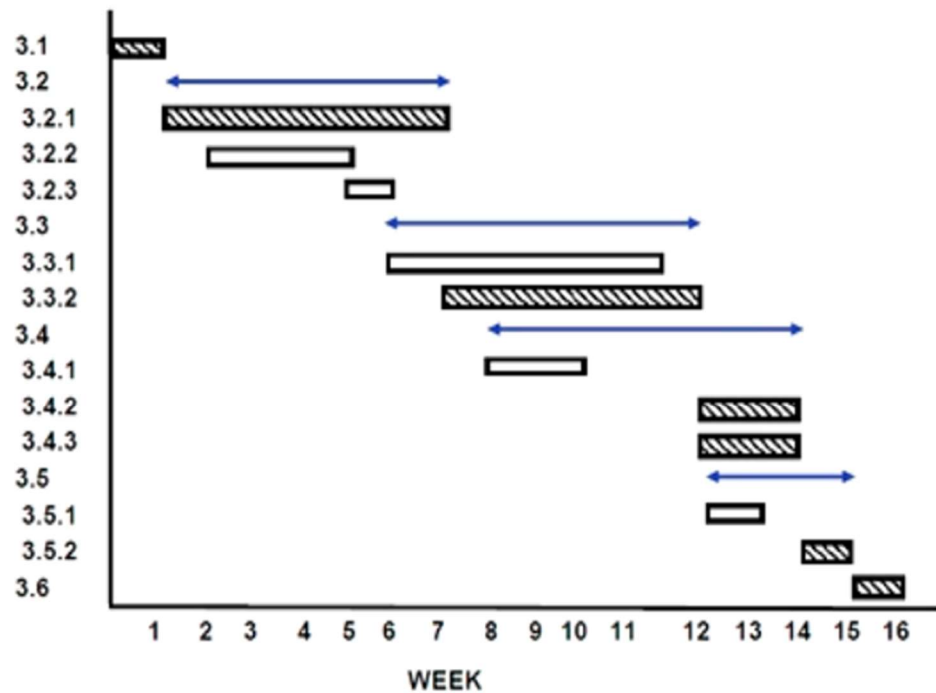
## A Cumulative PERT Probability Distribution for a Schedule



احتمال موفق شدن پروژه مثلاً در 14 هفته 75 درصد است

نمودار گانت Gantt

## A WBS – Gantt Chart



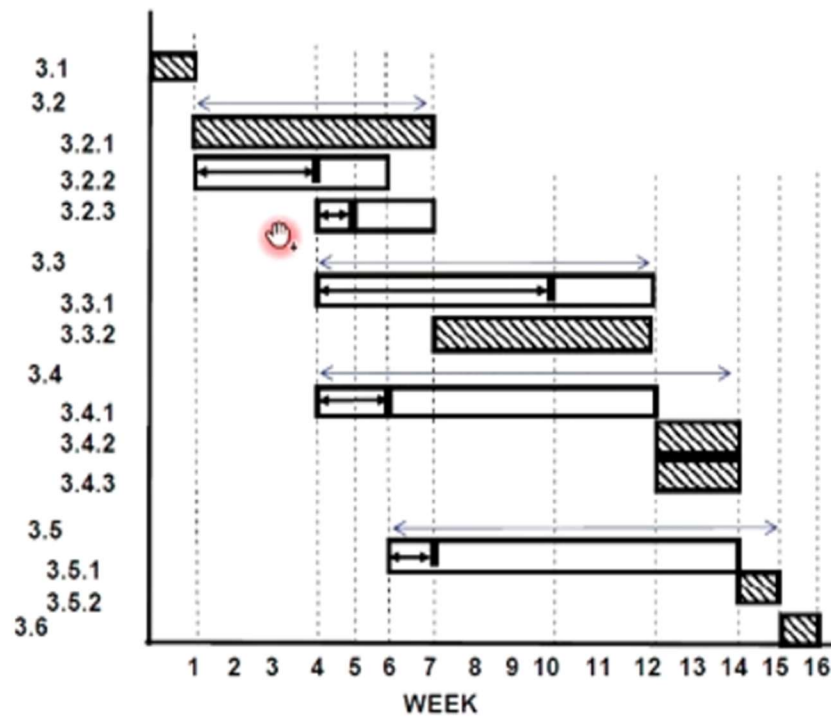
یکی از چالش ها توزیع منابع در زمان پروژه است بنابراین از نمودار گانت استفاده میکنیم

در یک بعد زمان و در دیگری task

Task های هاشور خورده مسیر بحرانی هستند

نمودار slack با gantt

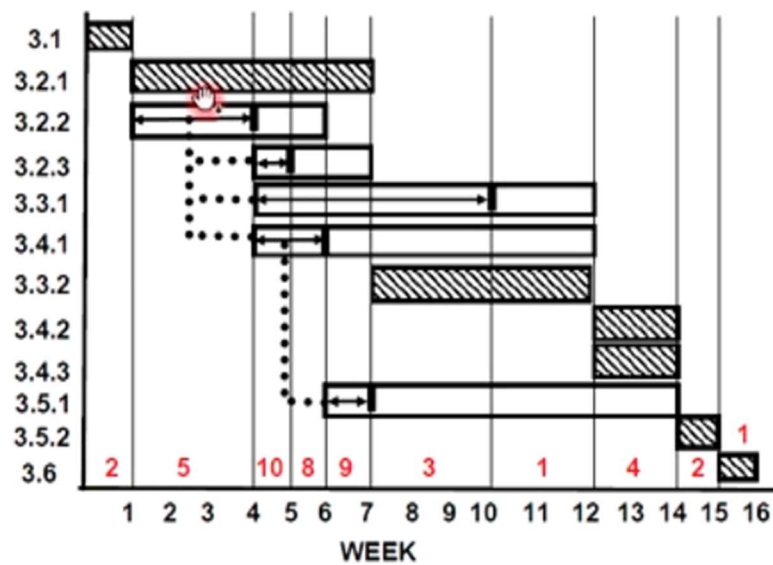
## A Gantt Chart with Slack Times



cross-hatched boxes indicate critical path tasks  
clear boxes indicate tasks with slack times

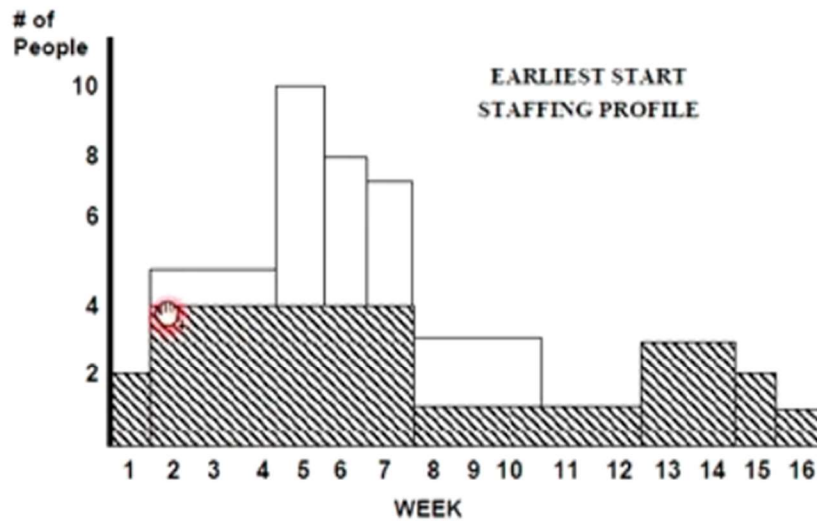
نمودار گانت با نیروها

## A Linked-Task Gantt Chart with # of Personnel Needed



نمودار منابع و افراد (زودترین زمان)

## Earliest Start Time Staffing Profile



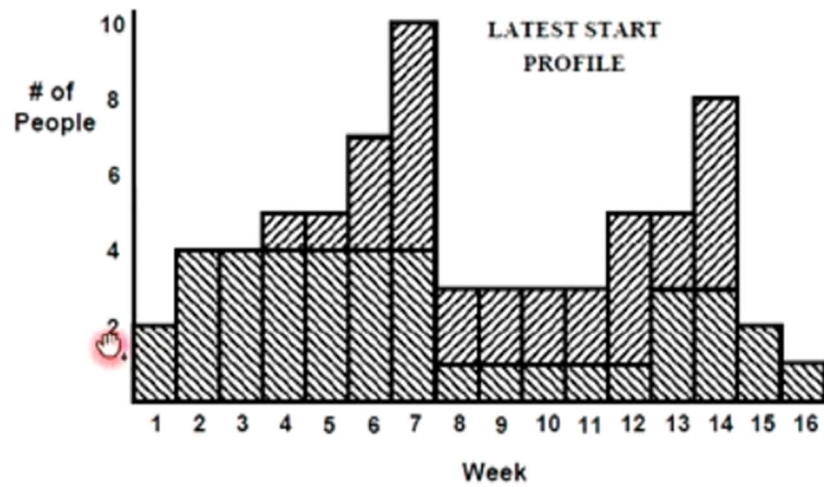
cross-hatched boxes indicate critical path tasks  
clear boxes indicate tasks with slack times

قسمت هاشور خورده مربوط به task های مسیر بحرانی هستند

نیروهایی که روی مسیر بحرانی نیستند را میتوان جابجا کرد. مثلاً task هایی که slack دارند را در زودترین زمان شروع کرد

نمودار دیرترین زمان شروع

## Latest Start Time Staffing Profile

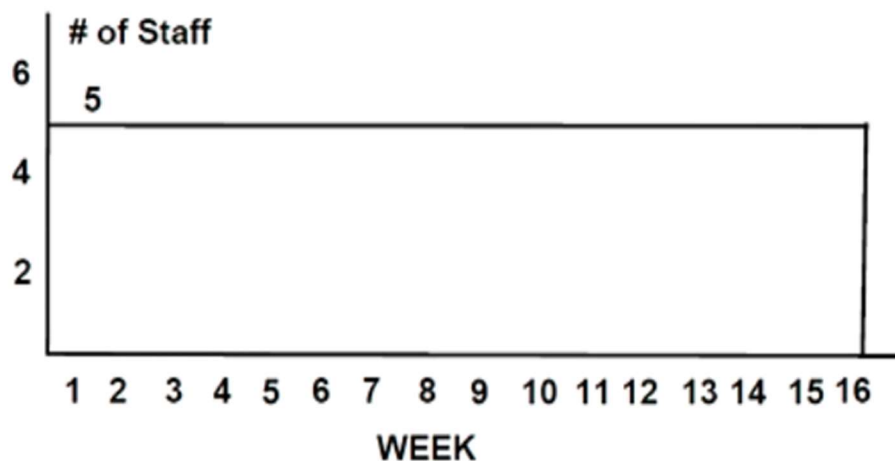


cross-hatched boxes indicate all tasks are on critical paths

بستگی به توزیع نیروی انسانی دارد

اگر مشتری محدودیت نیروی انسانی بگذارد

## A Desirable, but Unobtainable Schedule for the Example



The schedule/resource allocation problem in this example is caused by the large number of software developers needed in weeks 4, 5, and 6

اگر به نمودار زودترین نگاه کنیم مثلاً هفته 5 10 نفر میخواد که میتوان 5 نفر از آنها را تقسیم کرد و تعداد هفته ها را زیاد کرد تا توازن ایجاد شود

ایجاد توازن

افزایش زمانبندی که در هفته های پیک از منابع کمتر استفاده کنیم

افزایش منابع

استفاده از منابع کارا تر

حوزه بندی مجدد نیازمندی ها

دوباره مرتب کردن Task ها که در هفته های پیک منابع کمتری استفاده کنیم

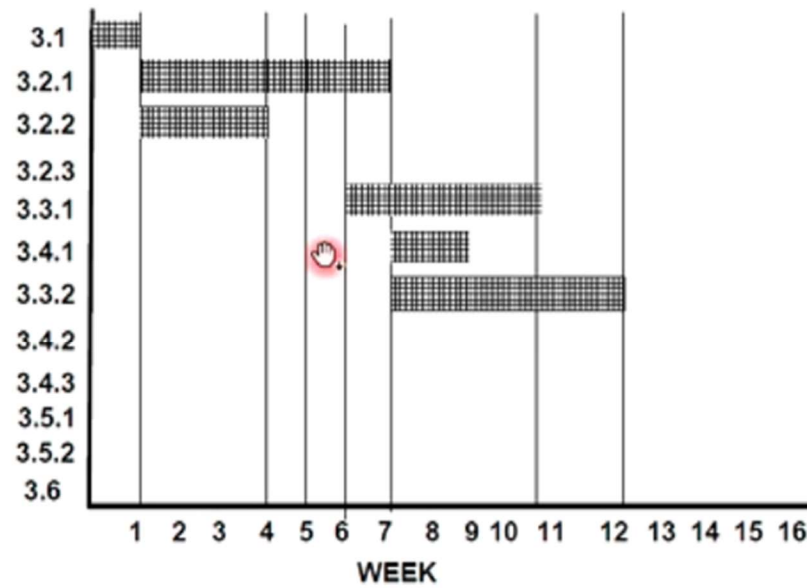
پلن غیرواقعی نداشته باشیم

فعالیت های مربوط به کنترل کیفیت را حذف نکنیم

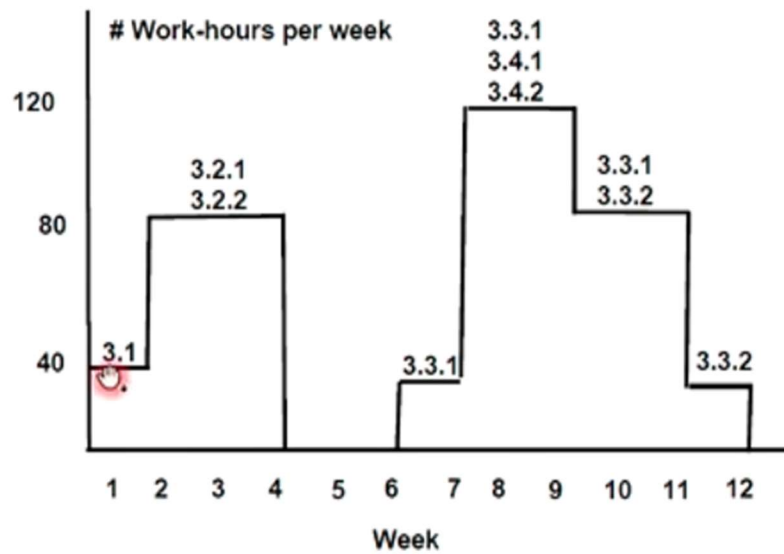
برنامه ریزی برای وقت گرفتن نکنیم



## A Resource Gantt Chart for Joe Hotshot



## An Infeasible Staffing Profile for Scarce Resource Joe Hotshot



## A WBS Task List with Calculated Effort

Task #	Description	Predecessors	Duration	#Staff	Effort
2.1	Receive approval to proceed	--	--	--	
3.1	Analyze requirements	2.1	1	2	2
3.2	Design				
3.2.1	Redesign existing components	3.1	6	4	24
3.2.2	Design new components	3.1	3	1	3
3.2.3	Design interfaces	3.2.2	1	2	2
3.3	Implement code				
3.3.1	Implement new code	3.2.2	6	2	12
3.3.2	Modify existing code	3.2.1, 3.2.3	5	1	5
3.4	Finish implementation				
3.4.1	Develop integration plan	3.2.2	2	2	4
3.4.2	Finish unit testing	3.3.1, 3.3.2	2	2	4
3.4.3	Update documentation	3.3.1, 3.3.2	2	3	6
3.5	Integrate and test				
3.5.1	Develop integration tests	3.4.1	1	3	3
3.5.2	Perform integration tests	3.4.2&3, 3.5.1	1	2	2
3.6	Perform acceptance tests	3.5.2	1	1	1

Effort:  $\Sigma(\text{Duration} \times \text{\#Staff}) = 68 \text{ staff-weeks}$

ستون Effort میزان تلاش (نفر هفته)

معمولا هزینه نیروی انسانی 30 تا 50 درصد کل هزینه پروژه میگیرند