

Object-oriented (شی گرا):

1) object (شی)

2) interaction (تعاملات)

وژگی ها: Reuse (۲) سرویس گذار از هر طرف به هر طرف (۳) قابل تجزیه (decomposition) نیست.  
☆ برای تجزیه کردن، Complexity Management استفاده کنیم.

Service-oriented (خدمات گرا):

از قراردادی (S.C) Service consumer, (S.P) Service Provider  
استاد c می کند که S.C ها؛ S.P ها استاد c می کنند و به کار بر یا می خورند و S.P ها  
درون خود سیستم نیستند و در یک سیستم دیگری باشند.

Software-Processes (فرآیندهای نرم افزاری):

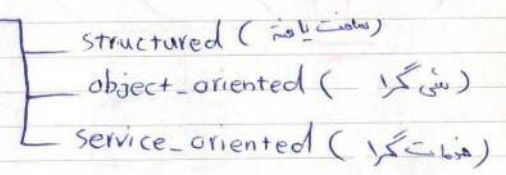
1) Productivity (مردودی) 2) Quality (کیفیت)  
در راستای بالا بردن و کیفیت

استانداردهای فرآیندهای نرم افزاری: bootstrap - CMM - IEEE - ISO 20000 - ISO 9000

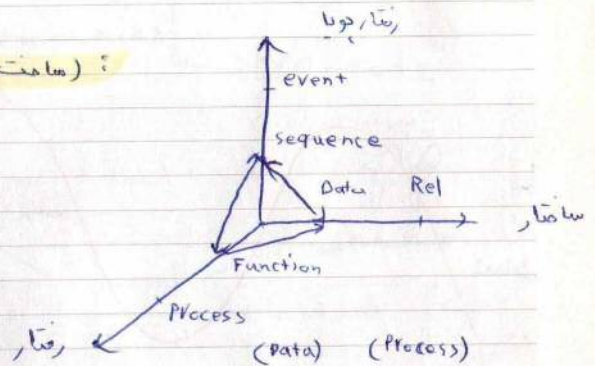
Software Development Method (S.D.M):

- 1) Concept → مفاهیم
- 2) Identification technique → باز شناسی
- 3) Representation technique → باز نیایی
- 4) Transition technique → گذر

Concept (مفاهیم):



structured (ساماندهی یافته):



1) Data oriented: از رفتار به ساختاری رسم مدل IE  
مدل داده ای است که رفتار را نشان می دهد.

2) Process oriented: از ساختار به رفتار رسم مدل  
مدل رفتاری است که ساختار را نشان می دهد.

CDM, SSADM, BSP

شهادت حضرت امام علی نقی الهادی علیه السلام (۲۵۲ هـ.ق)

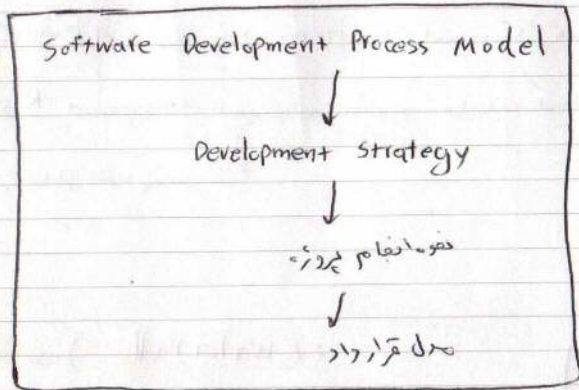
✓ **Change Management (مدیریت تغییرات):**

نظای که محصول ضرورتاً تغییر پذیر نباشد، کیفیت  
تغییرات با اجازت مدیریت تغییرات

✓ **Configuration Management (مدیریت پیکربندی):**

کنترل، سازگاری، نسخه های مختلف محصولات

✓ **" مدل فرآیند توسعه S.D.P.M "**



در صورت تغییر اضافه شدن

به صورت بی با خبری

(۱) خطی (Linear) - بازگشت به عقب ندارد (تغییرات کمپلکس جدا در زمان بی نمود)

(۲) آبشاری (Waterfall) - بازگشت در طولی درون loop های است

(۳) تپاکی (Evolutionary) - رسیدن های تپاکی، انبارد بقیه رسیدن بکار چینی را دارد

✓ **مهمترین فرآیندهای نرم افزار:**

- Project Planning → برنامه ریزی پروژه
- Project Tracking → ردیابی یا کنترل پروژه
- Requirement Management → مدیریت نیازمندی ها
- Software Quality Assurance (S.Q.A) → فرآیند تعیین کیفیت
- Change Management → مدیریت تغییرات
- Configuration Management → مدیریت پیکربندی

✓ **Project Planning (برنامه ریزی پروژه):**

شامل برنامه ریزی فعالیت ها بر مبنای منابع (انسان، ... ) است ← (Task, Starttime, Endtime, Resource)

✓ **Project Tracking (ردیابی پروژه):**

بررسی میزان پیشرفت ها و هزینه ها

✓ **Requirement Management (مدیریت نیازمندی ها):**

بررسی تحقق نیازمندی های پروژه چون هزینه های از طرف کارفرما و سطح کار مدیریت نمی شود

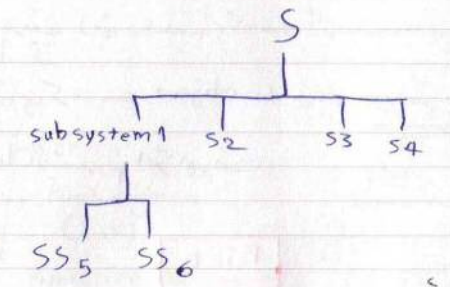
✓ **(S.Q.A) (فرآیند تعیین کیفیت):**

به آن VV گفته می شود: Validation معتبر بودن نتیجه کار

Verification درست انجام شدن کار

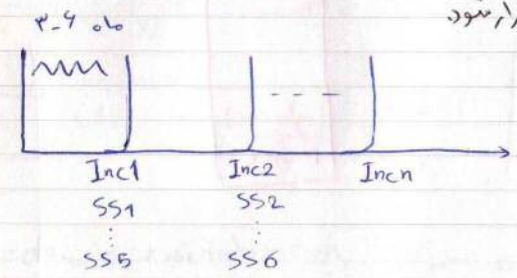
روز بزرگداشت شیخ صدوق - روز جهانی ماما

Incremental



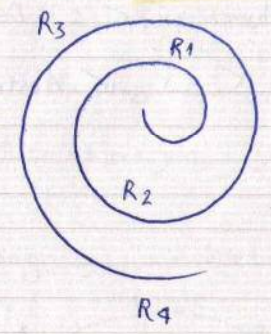
انبار سبب تکلیف می (Integration Risk)

دریم که تا به بهاری نرم افزار شود



روز جهانی صلیب سرخ و هلال احمر

"spiral"



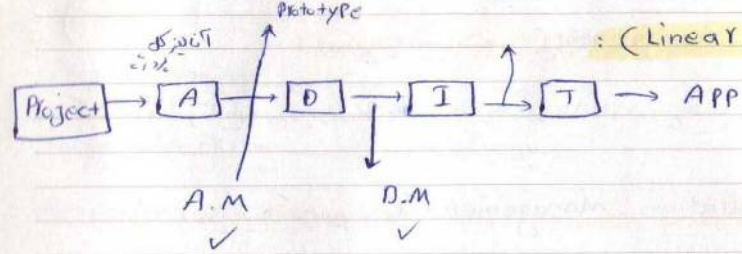
Release اول R1

Release دوم R2

function های سیستمی داشته باشد

روز بزرگداشت شیخ کلینی - روز اسناد ملی و میراث مکتوب

I.M



خطی (Linear)

1) انعطاف پذیری Flexability

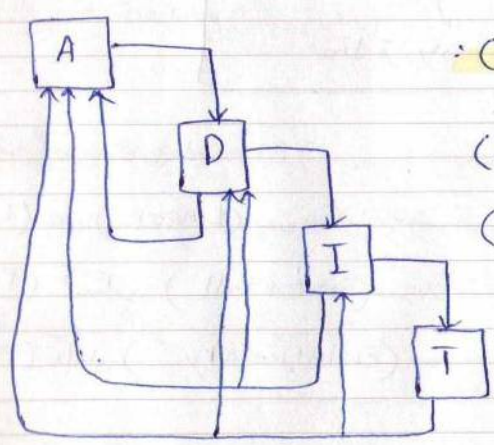
2) ریسک پذیرش (Acceptance Risk) دار یعنی احساس می شود که کاری انجام گرفته است

3) برای کاهش A.R از نمونه مدل (Prototype) استفاده می کنیم تا مشتری کار را ببیند

4) Management Risk: مدیریت ریسک مشتری پروژه را هدایت کند یعنی با تصمیم مدیریت کارفرما

مردم پروژه همکاری می کنند و به وقت می شود

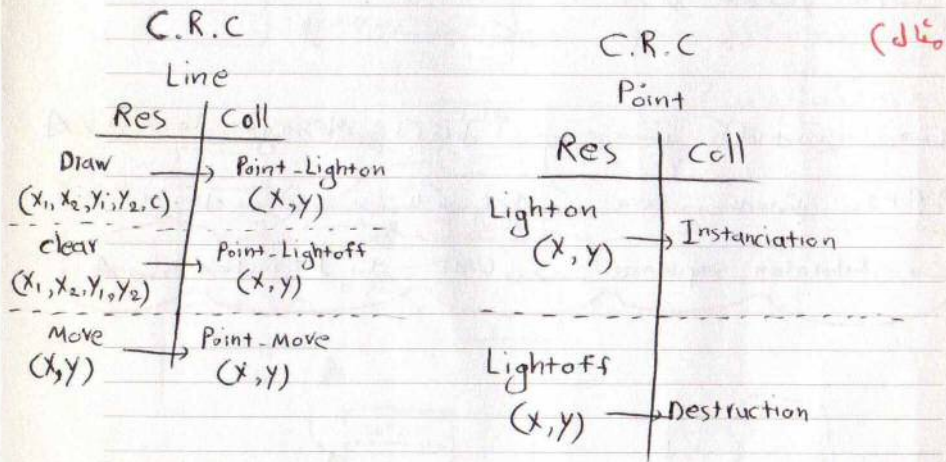
آشنا (Water Fall)



1) گام به گام (Incremental)

2) تدریجی (Spiral)

Interaction در مدل‌های R.D.D مهمی بنام Collaboration Responsibility  
 داریم که برای هر class یک C.R.C (C.R card) تولید می‌کنند.



نکته: / Responsibility ما برای انجام کار Collaboration ما را در برده.

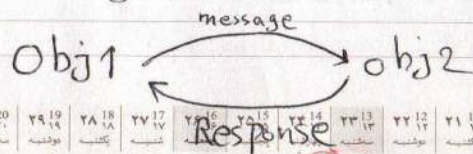
در مدل‌های Event-Response Event یک ext event وارد می‌شود و آن event

در باره Int event ایجاد می‌شود و تمام Response ما به آن آوری شود و ما به آن در می‌شود.

ایراد event قابل آوری نیست

قابل دسترسی نیست (object با آن event آوری می‌شود و گرفته می‌شود)

☆ در مدل‌های اصلی امروزی Interaction به عنوان یک Message در نظر گرفته می‌شود.

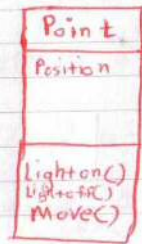
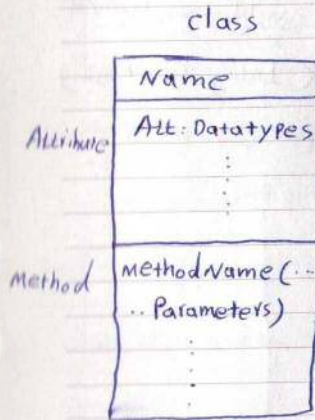


# Object Oriented

Object: هر چیزی (شیء یا مفهوم) که بتوان آنرا با مجموعه‌ای از خصوصیات (Att) رفتار داشت

خطی بازنمایی کرد. Object همانا برای نوع خصوصیات با نوع رفتار Classify می‌کنند

☆ Class یک مفهوم انتزاعی است



☆ Object می‌تواند Object

ایجاد می‌کنند و سیستم باید Instantiation کند.

☆ Instantiation: وقتی که باعث ایجاد Class می‌شود هر کلاس حداقل یکی دارد!

☆ Destruction: وقتی که یک Object از Class حذف می‌کنند در نظر می‌گیرند!

☆ Abstract Class، کلاسی است که Instantiation ندارد.

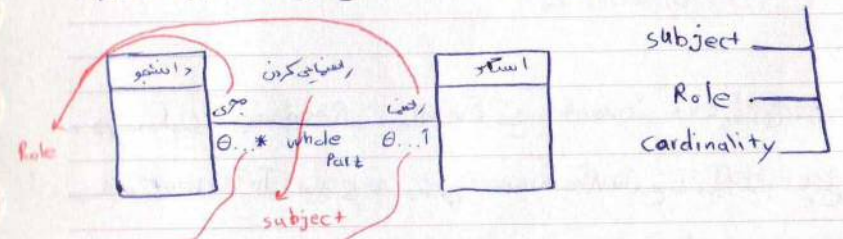


Message ها (دو نوع اند: ۱) سفرون : بهوز ارسال، منتظر جواب (Res) میماند  
 ۲) آسفرون : بهوز ارسال منتظر نمیماند و هر زمان که جواب (Res) آید از آن استفاده می کنند.

☆ ویژگی های Message : ۱) قابل آدرس دهی هستند ۲) قابل ذخیره سازی هستند  
 ۳) پروتکل پذیر است یعنی میتونی Net، ویک، Java، پروتکل ها مثل...، POP3، HTTP  
 ☆ نمودار Interaction صادر UML دو مورد Collaboration، sequence هستند

روابط بین Class ها :

Association (1) ✓ نشان دهنده مشارکت object های طریق رابطه در تمام یک عمل می باشد

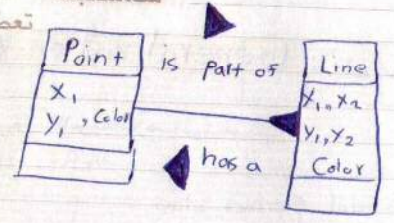


Aggregation (2) ✓ رابطه ای که نشان می دهد یک عنصر از مفهوم دیگری تشکیل شده است : cardinality

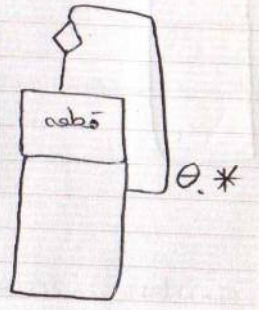
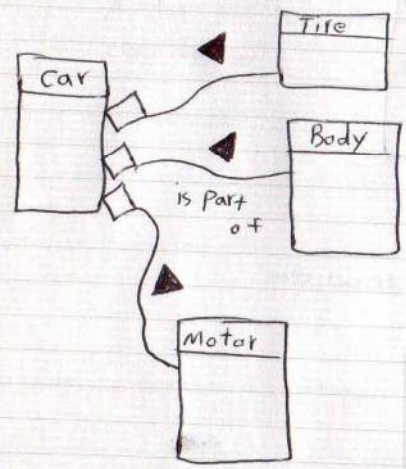
\* Point is a Part of Line  
 \* Whole-class has a Part-class



تعطیل

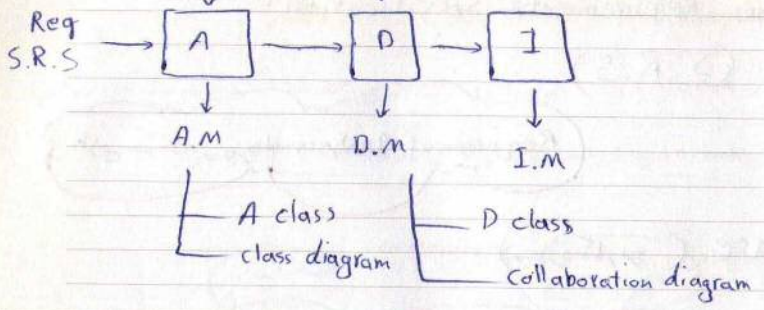


☆ اگر تکی باشی لزومی تو خالی است ولی اگر چند تا باشی تو میرات  
 ریب نقطه در ساعت بیش از یک خط مشارکت دارد



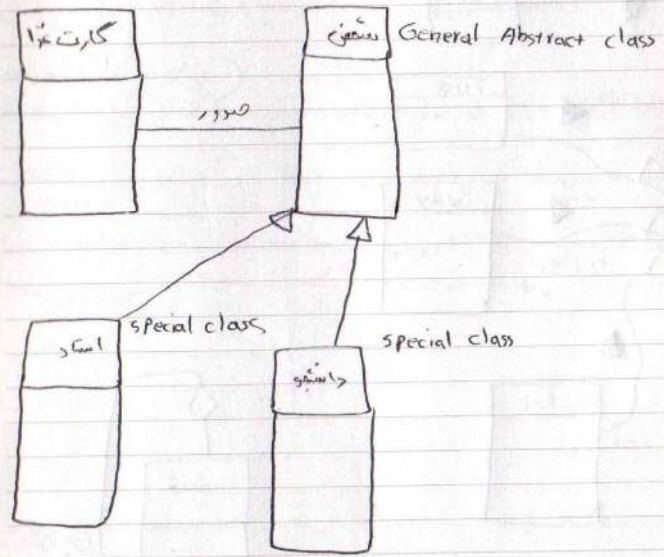
☆ نکته / رابطه های Association و Aggregation، رابطه های غیر کلاسی هستند یعنی یک Job از هر کدام از Class ها باهم ارتباط دارند ولی Generalization، رابطه کلاسی ها است.

ولادت حضرت امام علی علیه السلام (۲۳ سال قبل از هجرت) (تعطیل) - آغاز ایام البیض (اعتکاف)



Generalization (3) ✓ : برای بهره‌رسانی و تکرار و جلوگیری از تکرار

که روی کلاس‌ها تعریف می‌شود.

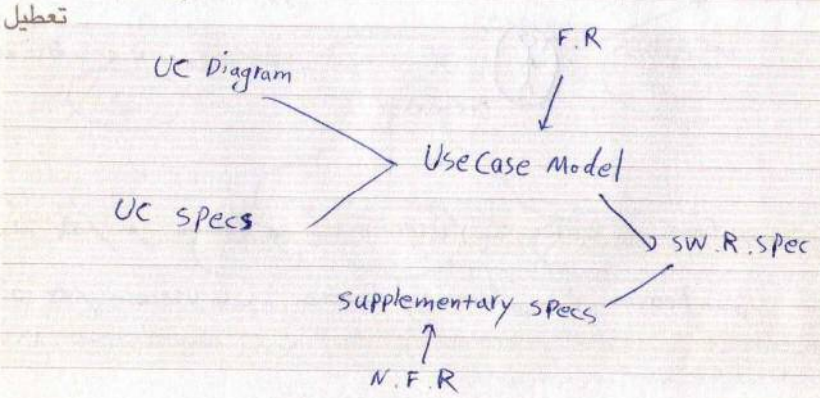


نکته: / \* شرط Abstract بودن یک Class، شناسایی همه special های Class است.

• سوالات مدل Object Oriented

class Diagram (1) Collaboration (Sequence) Diagram (2)

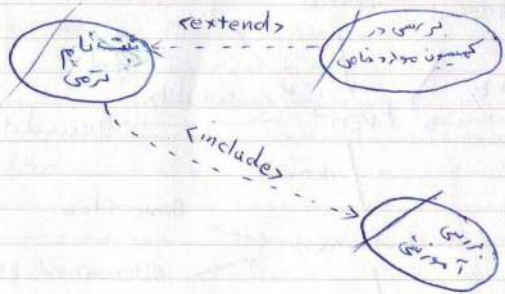
لغو امتیاز تنباکو به فتوای آیت‌الله میرزا حسن شیرازی (۱۲۷۰ هـ.ش)



روابط بین Usecase

extend, include : اگر یوزکسی فرعی ظاهری باشد که اگر انجام نمیشود کار یوزکسی اصلی را انجام میدهد

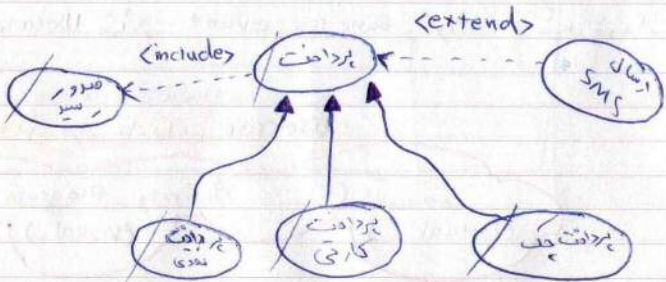
include و اگر انجام نمیشود هم تا وقتی فلزارد extend می شود.



☆ جهت include از یوزکسی اصلی به یوزکسی فرعی و جهت extend از فرعی به اصلی است.

☆ رابطه generalization Usecase های که رابطه نسلان و مسترک دارند را به یک Usecase

general وصل کرده در رابطه آنها از نوع generalization انتخاب می کنند.



☆ نقشه بخش (Requirement Analysis) R.A نمودار (Usecase Diagram) U.C.D است.

روز بزرگداشت حکیم عمر خیام - روز جهانی موزه و میراث فرهنگی

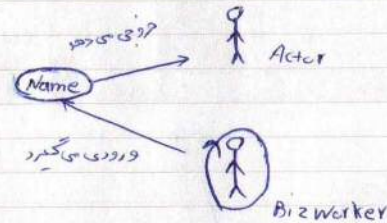
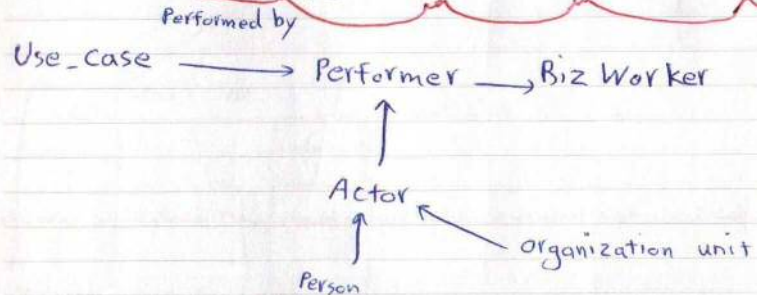
Software Requirements Specification:

(S.R.S)

طرحی مستند از مرحله Requirement Analysis

☆ Usecase (موارد کاربرد):

صرف یک Function یا یک عمل در سیستم است.



☆ Actor : U.C فرعی می گیرد

☆ Biz Worker : U.C و فرعی می گیرد

☆ Note / هر Button یعنی یک Usecase (پشت Button یک Function اجرا می شود).

☆ Note / Usecase ها حالت Elementary Function می باشند و Reuse هستند.

روز ارتباطات و روابط عمومی

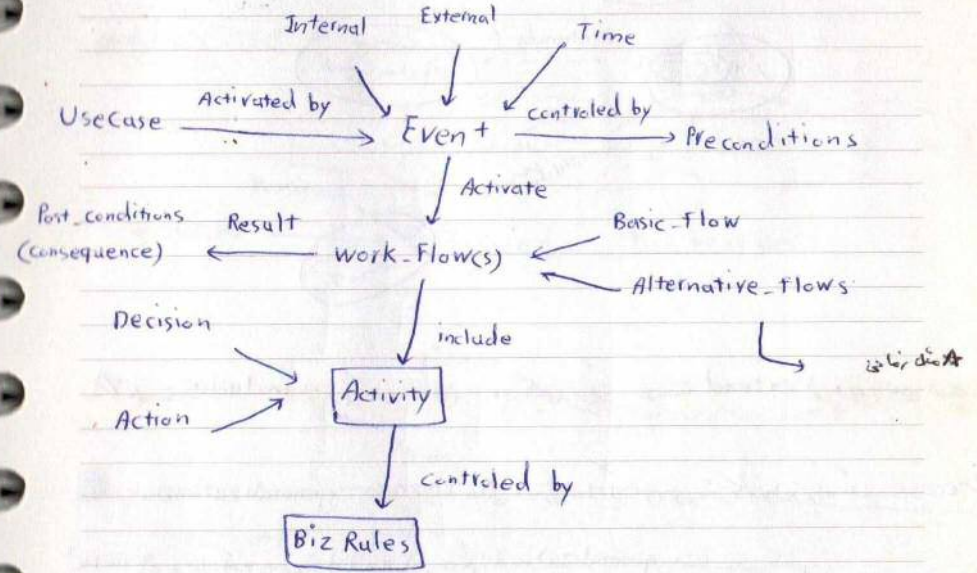
Usecase Template

U.C Name:	Project Name
* Event:	
Pre-condition:	
* Basic flow:	
Alternative flow:	
Biz-Rule(s):	BR1 BR2 :
* Post-condition(s):	
Information Resource:	

Basic-flow  
ایجاد و حالت عادی  
اجرای UseCase ها  
است.

Usecase Specification

U.C زمانی که یک event اجرا شود و Pre-condition ما هم برقرار باشد.  
در اجرای U.C، انجام event اجباری است اما ممکن است Pre-condition نداشته باشد.



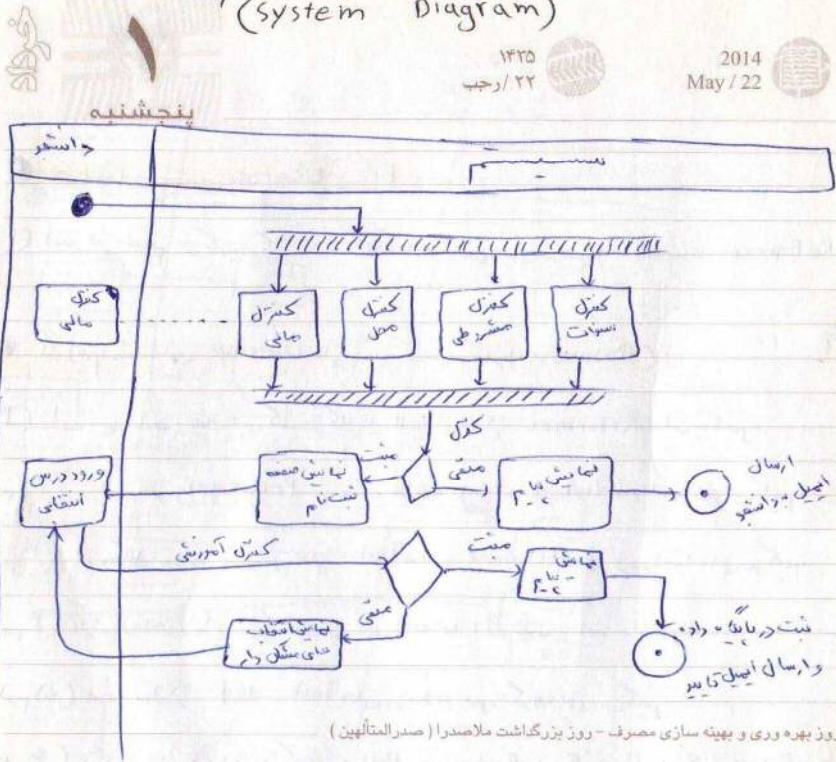
Usecase متناظر با: event, Basic flow, activity شامل شود است.

روش برای نمایش UseCase

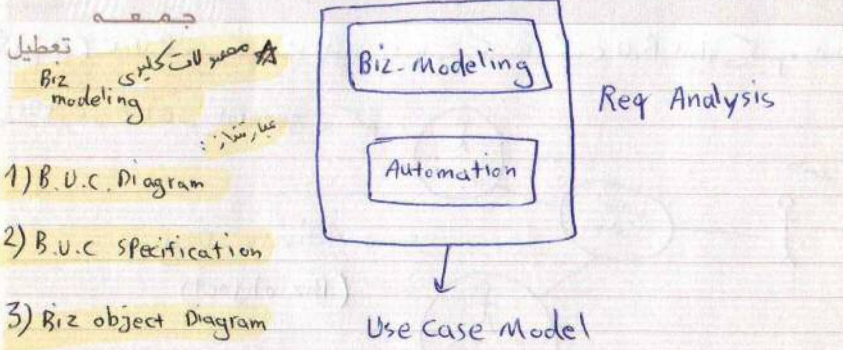
Activity Diagram (۱) (Visual)  
Usecase (۲) (Textual)



# "Activity Diagram for Registering" (system Diagram)



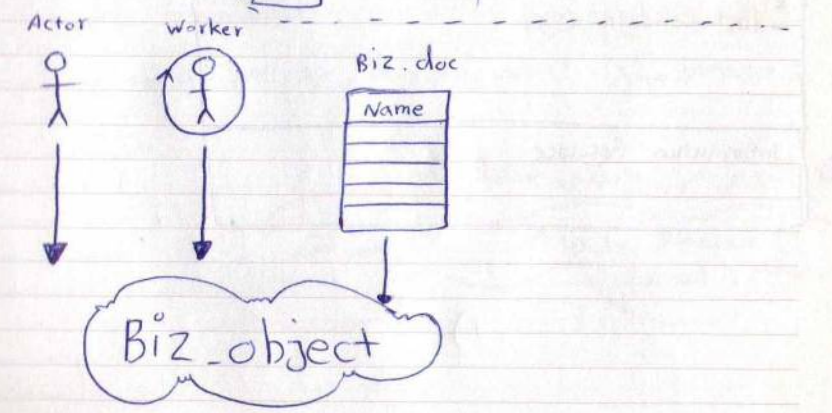
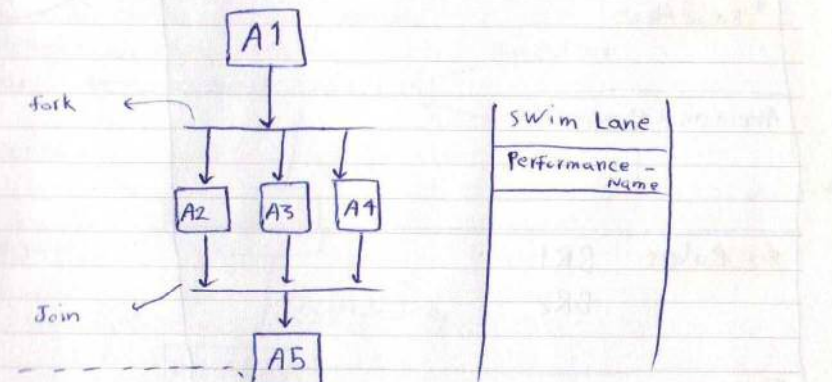
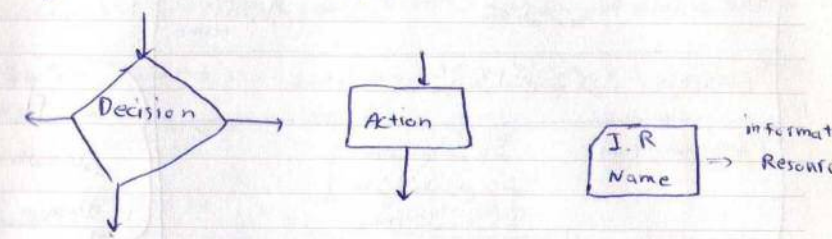
روز بهره وری و بهینه سازی مصرف - روز بزرگداشت ملاصدرا (صدرالمعالمین)



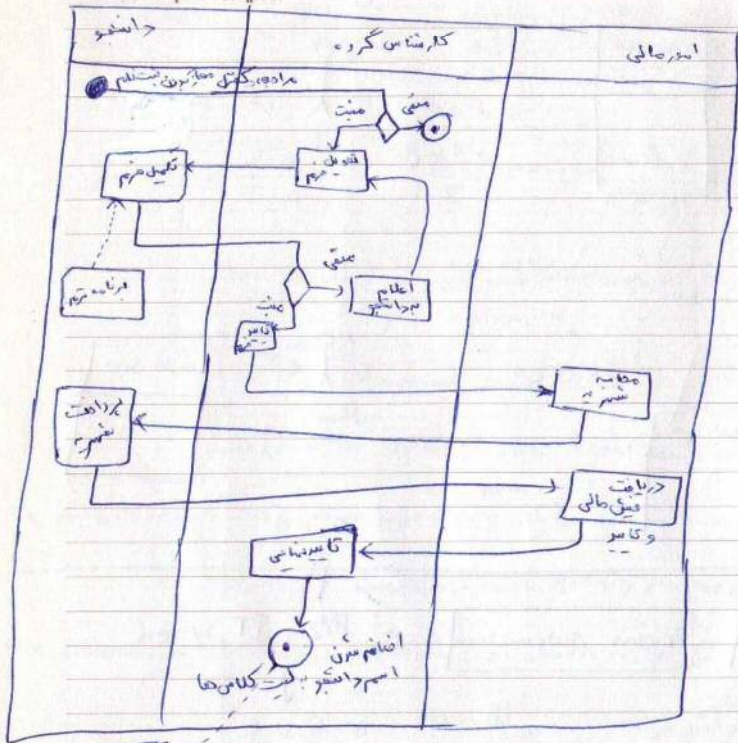
- 1) B.U.C Diagram
- 2) B.U.C Specification
- 3) Biz object Diagram

☆ Biz object Diagram = تعامل بین اشیاء Biz، امکان می دهد.

Event name / Pre-condition  
 flow  
 Post-condition Name  
**Activity Diagram**



"B.U.C specification (Activity Diagram)"



الکت کلاس

بخش ۳ : Biz-Usecase

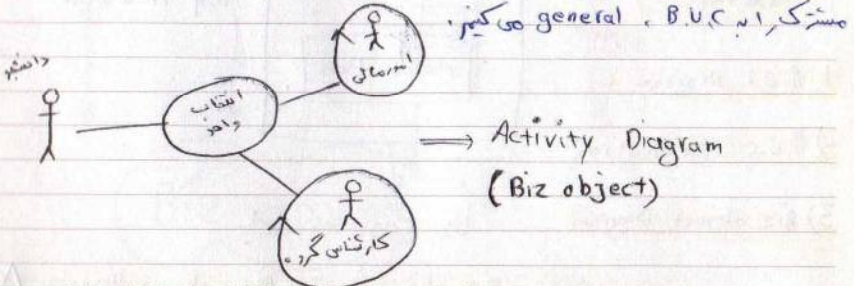
۱) استخراج مستقیم از کتب و کار (۲) استفاده از مدل‌های مرجع (Reference Model)

۲) الگوریتم شناختی Usecase های تک کسب و کار (Business):

- گام ۱) از روی جداول سازمانی کسب و کار یک لیست اولیه از Performer صادر می‌کنیم.
- گام ۲) به ازای هر Performer ضرورت و وظایف (Task List) را استخراج می‌کنیم.
- گام ۳) از وظایف بر روی دفره‌های Task ها، Performer های جزو شناختی می‌کنیم.
- گام ۴) تکرار مرحله ۲ و ۳ تا زمانی که Performer جدیدی اضافه نشود.
- گام ۵) صرف نظر از P ها، Task های مرتبط هم را گروه بندی می‌کنیم.
- گام ۶) هر گروه از آنها را یک B.U.C می‌دهیم P های آن، P های آن B.U.C هستند.
- گام ۷) B.U.C هایی که آمشاب دارند، آمشاب را هم در برگیرند به یک B.U.C جزو می‌کنیم که B.U.C می‌باشد include یا extend دارند.

گام ۸) B.U.C هایی که بی بی یا جزو رابطه مشترک دارند به یک B.U.C ایجاد می‌کنیم و رابطه

مشترک را به B.U.C general می‌کنیم.





سه شنبه

۱۴۳۵  
۲۷ / رجب

2014  
May / 27



تعطیل  

$$\text{Total Activities (T.A)} = \text{system Assisted Activities (As.A)} + \text{Manual Activities (A.M)} + \text{Automated Activities (A.A)}$$

$$\text{سطح اتوماسیون} = \frac{A.A}{T.A} \quad \text{در حد اتوماسیون} \rightarrow \frac{A.A + A.S.A}{T.A}$$

☆ هر چه درجه سطح اتوماسیون بالاتر باشد میزان است (ماکزیم ۱۰۰)

● **Process Automation** (فراآینز اتوماسیون):

گام ۱) B.U.C Spec یا Performer چه نوع IT system ارائه می کنند.

گام ۲) تعیین مفروضات با مشتری و توافق بر آن (نیاز به یک مدل مفروضات، حذف کلمات، حذف غیر ضروری استثنای)

گام ۳) باز نویسی Activity Diagram با در نظر گرفتن مفروضات اتوماسیون

گام ۴) ۳ حالت نظیر Activity ها (الف) manual (ب) Automated (ج) system Assisted

گام ۵) کندا گذاشتن Activity های Manual

گام ۶) هر S.A یا A به عبار خودیش تا S.A بقوی و دست گروه می گذاریم و تبدیل به S.U.C System می کنیم.

گام ۷) اگر یک Activity در سبب S.U.C مشترک باشد آنرا تبدیل به یک S.U.C جدا می کنیم و با S.U.C های قبلی رابطه extend یا include می زنیم.

گام ۸) S.U.C های که بیش از یک رابطه داشته باشند را در یک Usecase generalization می سازیم و

رابطه های مشترک را به general usecase مقصر می کنیم.

میث حضرت رسول اکرم صلی الله علیه وآله (۱۳ سال قبل از هجرت) (تعطیل)

(ادامه دارد در صفحه ۶۱)

۳۱	۳۰	۲۹	۲۸	۲۷	۲۶	۲۵	۲۴	۲۳	۲۲	۲۱	۲۰	۱۹	۱۸	۱۷	۱۶
۲۱	۲۰	۱۹	۱۸	۱۷	۱۶	۱۵	۱۴	۱۳	۱۲	۱۱	۱۰	۹	۸	۷	۶
شبه	جمعه	پنجشنبه	چهارشنبه	سه شنبه	دوشنبه	یکشنبه	شنبه	جمعه	پنجشنبه	چهارشنبه	سه شنبه	دوشنبه	یکشنبه	شنبه	جمعه

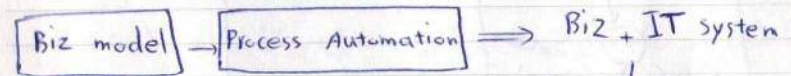
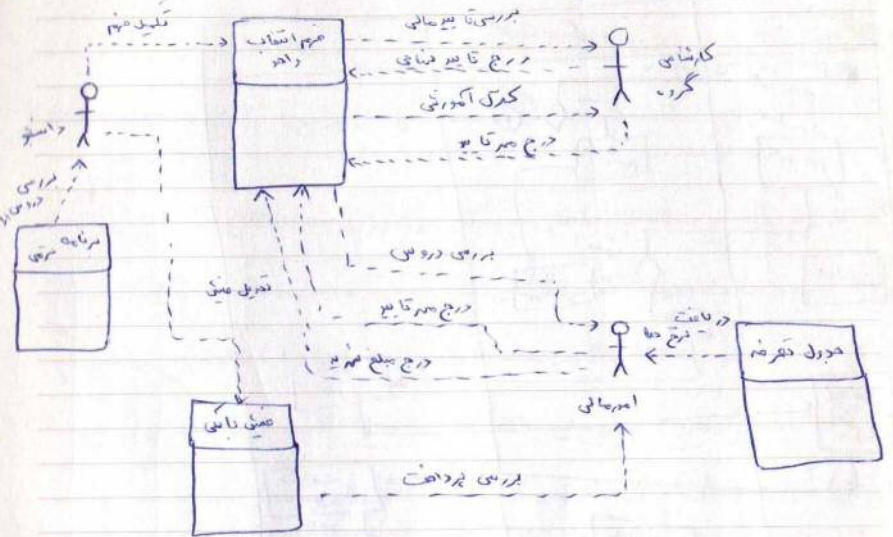
# "Biz object Diagram"

2014  
26 / May

۱۴۳۵  
۲۶ / رجب



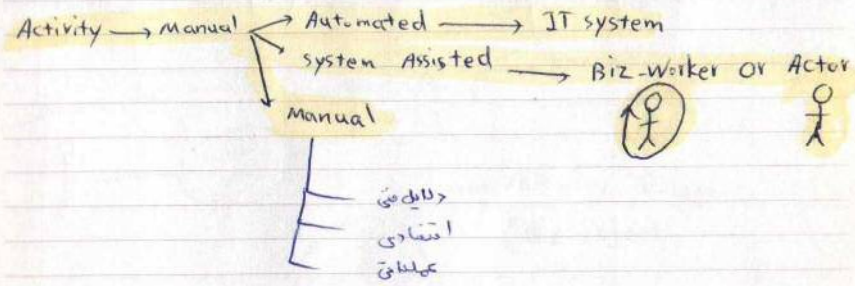
دوشنبه ● Biz object Diagram



$$N = B.U.C$$

$$M = \text{Activities}$$

$$T.A = N * M$$



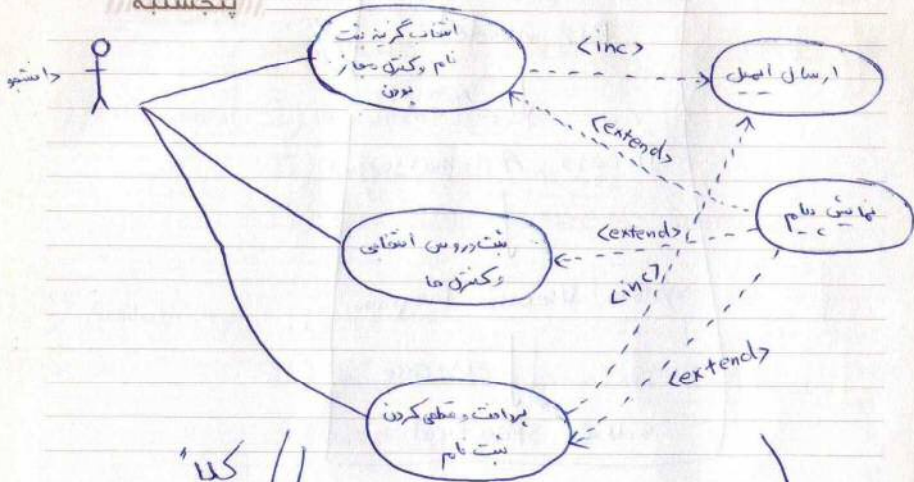
روز نهم مهر (روز حمایت از خانواده زندانیان)

۱۵	۱۴	۱۳	۱۲	۱۱	۱۰	۹	۸	۷	۶	۵	۴	۳	۲	۱
۵	۴	۳	۲	۱	۳۱	۳۰	۲۹	۲۸	۲۷	۲۶	۲۵	۲۴	۲۳	۲۲
پنجشنبه	چهارشنبه	سه شنبه	دوشنبه	یکشنبه	شنبه	جمعه	پنجشنبه	چهارشنبه	سه شنبه	دوشنبه	یکشنبه	شنبه	جمعه	پنجشنبه

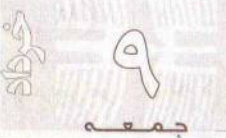
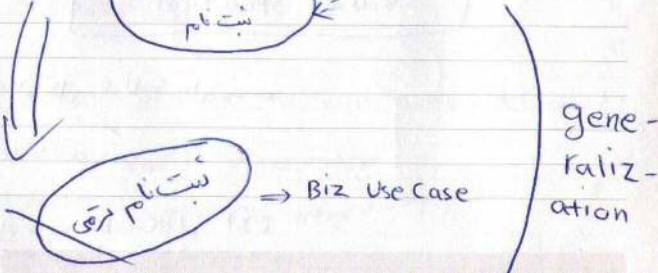
# "System Use Case"



1435 29 / رجب  
2014 May / 29

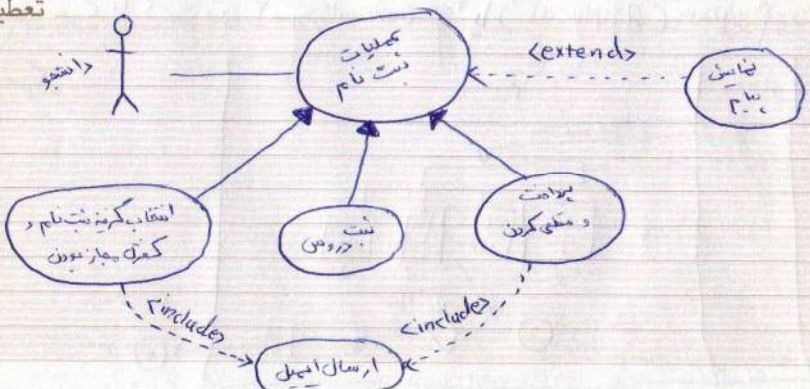


کلاس می شود



1435 1 / شعبان  
2014 May / 30

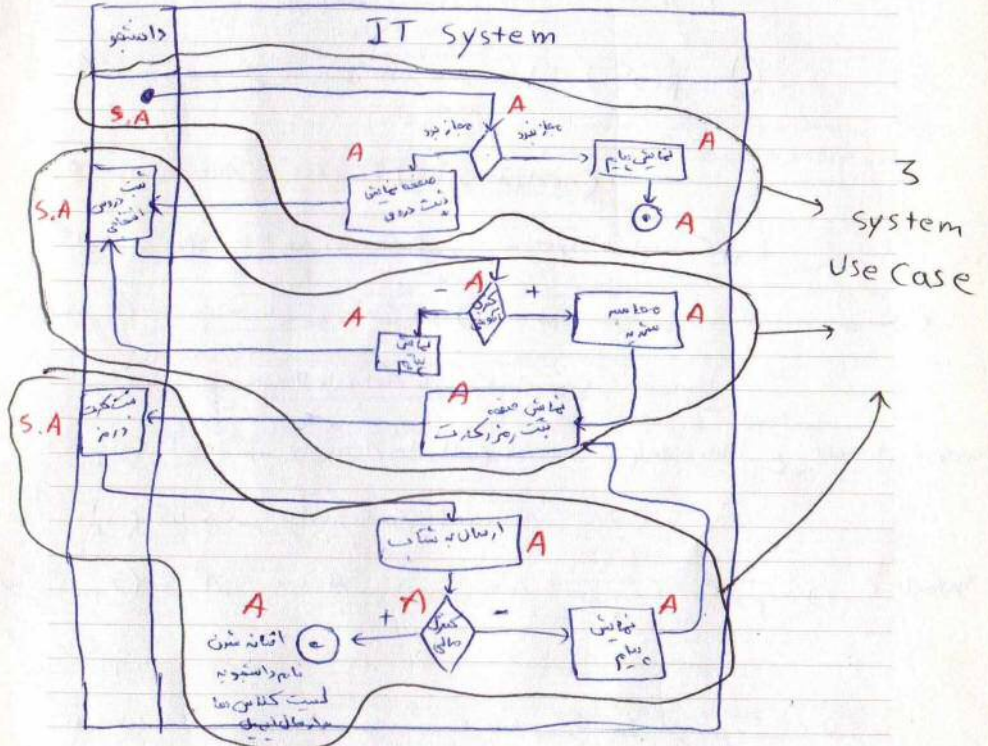
تعطیل



2014 28 / May  
1435 28 / رجب



نمونه Activity / Activity های S.A ، اکتم بر (USER) مثال می کند ، Activity های Automated ، activity های system assisted فعال می کند ، تسلیک ثبت S.U.C می دهند .  
(گام ۶)





۱۳۹۵  
۳ شعبان

2014  
June / 1

Use case : Quality Attribute

- 1) performance:
  - I) Concurrent: 30000 users
  - II) Response time: 12 sec
  - III) Data Load: 10000000 Records
- 2) Availability:
  - I) 24x7
  - II) (midtime to response) MTTR: 1 hour
  - III) Data-Lost = 0
- 3) Security:
  - I) Authentication: sms-based (username & Password)
  - II) Authorization Policy: صریحاً دسترسی
  - III) Audit: (سابقه ورود خروج کاربر) تغییرات حساسی کاربر
  - IV) Integrity: بررسی جامع داده ها، نفوذ جلوگیری از دسترسی بد داده ها modify
- 4) Useability
- 5) Portability
- 6) Modifiability
- 7) Manageability

☆ برای دسترسی استاندارد مستقر  
معروف ترین آئین عبارتند از:

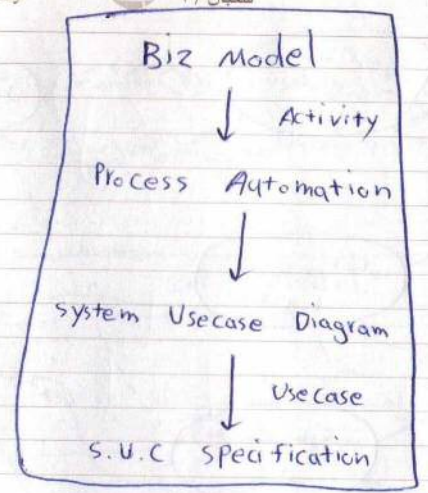
- Iso/IEC
- NASA
- BOHEM
- MCCALL
- IEEE

ولات حضرت امام حسین علیه السلام (۳ هجری) و روز پاسدار



2014  
31 / May

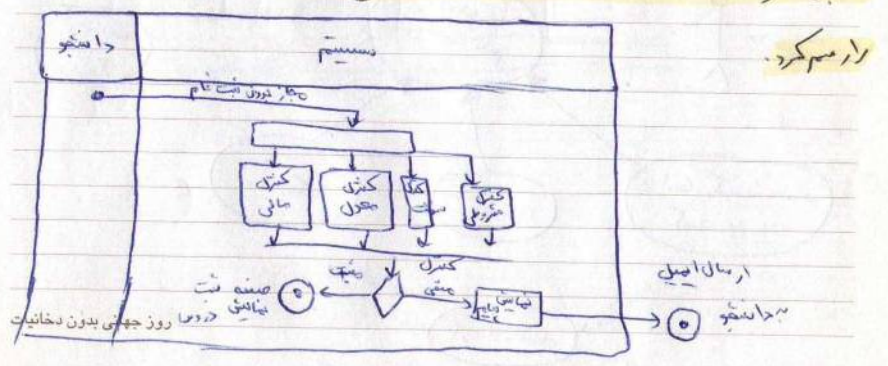
۱۳۹۵  
۲ شعبان



- گام ۹) ایجاب لیست از کل S.U.C های موجود
- گام ۱۰) حذف S.U.C های تکراری
- گام ۱۱) یکپارچه سازی کامل بر روابط ext, inc
- گام ۱۲) generalization

☆ برای هر S.U.C یک Activity Diagram داریم که S.U.C صرف یک ER است.

☆ برای هر S.U.C می توانیم فرمت (textual) یا Activity-Diagram (Graphical) را رسم کرد.



۱۵	۱۴	۱۳	۱۲	۱۱	۱۰	۹	۸	۷	۶	۵	۴	۳	۲	۱
۳۱	۳۰	۲۹	۲۸	۲۷	۲۶	۲۵	۲۴	۲۳	۲۲	۲۱	۲۰	۱۹	۱۸	۱۷
۲۱	۲۰	۱۹	۱۸	۱۷	۱۶	۱۵	۱۴	۱۳	۱۲	۱۱	۱۰	۹	۸	۷
۳۱	۳۰	۲۹	۲۸	۲۷	۲۶	۲۵	۲۴	۲۳	۲۲	۲۱	۲۰	۱۹	۱۸	۱۷

۵ رسم از Usecase Spec به A.C (Analysis Class):

گام ۱) به زنی هر V.C, U.C Spec, U.C, Collaboration رسم می کنیم.

گام ۲) پیام های ورودی به هر کلاس تبدیل به متد آن کلاس می شوند.

گام ۳) پیام های ورودی هر متد در کلاس های Boundary و Entity تبدیل به Att های

کلاس می شوند.

گام ۴) پیام های پیام ده یا من از کلاس های Entity تبدیل به Att های کلاس می شوند.

گام ۵) تکرار به زنی هر Usecase Spec

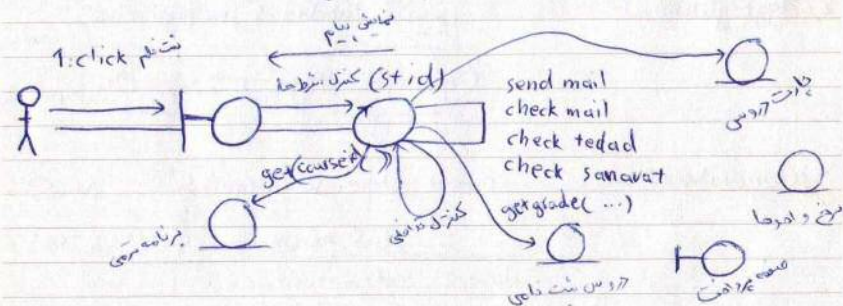
گام ۶) تهیه کد لیت از کلاس ها به فصل Boundary, Control و Entity

گام ۷) پالایش کلاس های متد

گام ۸) رسم B.C.D (Boundary class Diagram)

گام ۹) رسم C.C.D (Control class Diagram)

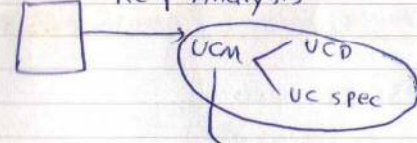
گام ۱۰) رسم E.C.D (Entity class Diagram)



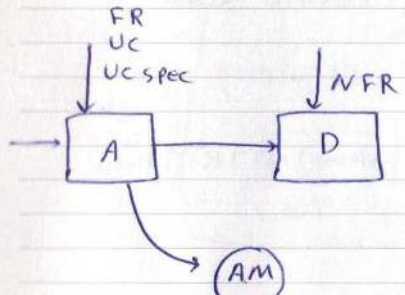
نکته: هر سه B.C.D, C.C.D, E.C.D می شود Analysis Class Diagram

ولادت حضرت امام زین العابدین علیه السلام (۲۸ هـ ق) - آغاز ماه مهجوریت (۱۵-۵ شعبان)

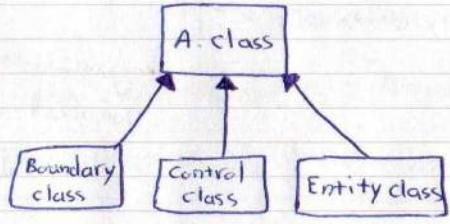
Req Analysis



supplementary - specification



class Diagram  
collab Diagram



- Boundary: و خطه ارتباطی بیرون سیستم با Actor یا Performer
- Control: منطق و محاسبات (Logic & Calculation)
- Entity: تعادل هر جدول در DB

ولادت حضرت ابوالفضل العباس علیه السلام (۲۶ هـ ق) و روز جانان

تعطیل

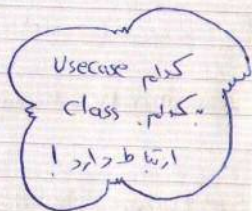
(C.C.D) Control Class Diagram

Generalization bias

(E.C.D) Entity Class Diagram

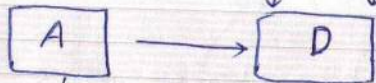
Generalization, Aggregation, Association

U.C \ AC	AC <sub>1</sub>	AC <sub>2</sub>	...	AC <sub>n</sub>
U.C1	✓		✓	✓
U.C2	✓			✓
...				
U.Cn	✓	✓	✓	✓



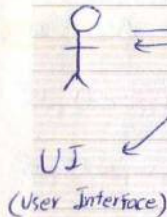
قیام خونین ۱۵ خرداد ۱۳۴۲ هـ ش (تعطیل) - روز جهانی محیط زیست

تعطیل



A.M	
C.C.D	
B.C.D	
E.C.D	- A.C.D

Collabrations class Diagram

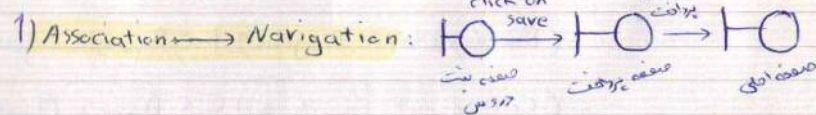


AL (Application Logic)

DM (Data management)

نکته: Analysis / class (Realize) (U.C) F.R. class

(B.C.D) Boundary Class Diagram



2) Aggregation:

نردب Page N حاوی که tabulare مستر یعنی فرم و ساب فرم دارد. (sub page, Page)

3) Generalization:

و رنگی همان مشترک همه Page های کل cancel, close, minimize, Color, font, ...

رحلت حضرت امام خمینی (ره) رهبر

کبیر انقلاب و بنیانگذار جمهوری اسلامی ایران (۱۳۶۸ هـ ش) - انتخاب حضرت آیت الله خامنه‌ای به رهبری (۱۳۶۸ هـ ش) (تعطیل)



۱۸

یکشنبه

۱۳۳۵  
شعبان / ۱۰



2014  
June / 8



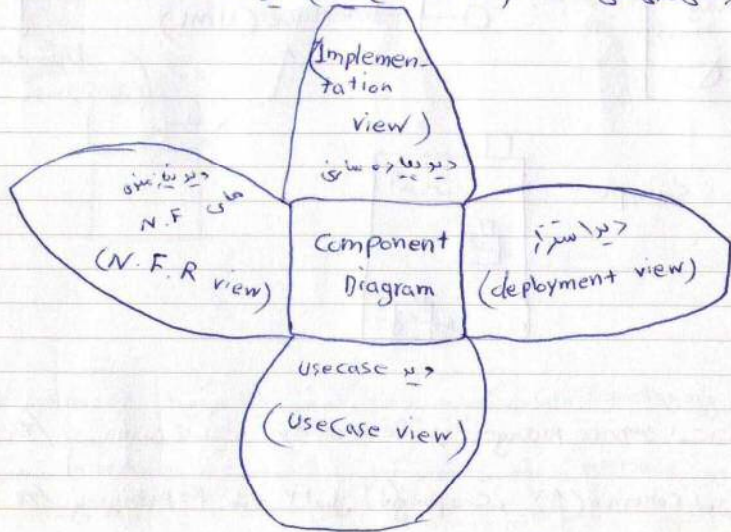
### Component (مؤلفه):

تبدیل ساختار و فرم معماری (structure) برای encapsulate کردن class ها داخل خود و ارائه عملگرها بصورت واسطه (Interface) به بیرون.

### فرآیند تدوین طرح معماری نرم افزار:

۱) مؤلفه سازی (Componentization)

۲) افعال دیگرام های معماری: ۱) دیو استر (deployment) ۲) دیو پیاده سازی (Implementation) ۳) دیو نیازهای مندی های N.F (N.F.R) ۴) دیو Usecase view



بعد از اتمام Component Diagram وارد Detail Design می شویم



شنبه

2014  
7 / June

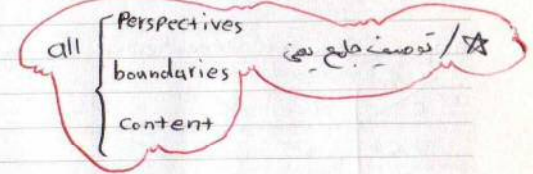


۱۳۳۵  
شعبان / ۹

### معماری و معماری نرم افزار:

تبدیل توصیف جامع یا پرهیز از ورود به جزئیات شامل عناصر ساختار و اصول و قواعد طراحی هر رابط

عناصر با کنترل معنایی است (Rational)

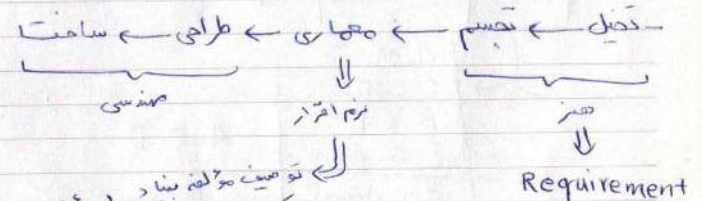


Complexity Management  $\Rightarrow$  Change Management Needs

Software Architecture: { Component (توصیف نرم افزار در قالب مؤلفه ها) Connection (ارتباطات) }

معماری نرم افزار: ۱) شناسایی مؤلفه ها ۲) افعال دیگرام های معماری روی مؤلفه ها

Change management  $\leftarrow$  Complexity Management  $\leftarrow$  معماری نرم افزار



توصیف مؤلفه بنیاد از یک سیستم (مؤلفه ها و ارتباطات)





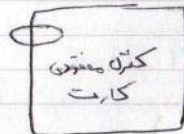
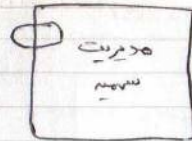
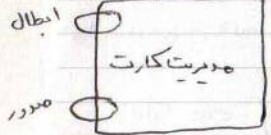


۲۲

پنجشنبه

۱۴۲۵  
شعبان / ۱۴

2014  
June / 12



Component Node	C1	C2	C3
N1	✓		
N2		✓	
N3	✓		✓
⋮			
Nn	✓	✓	✓



۲۳

جمعه

۱۴۲۵  
شعبان / ۱۵

2014  
June / 13



تعطیل

ولادت حضرت قائم عجل الله تعالی فرجه (۲۵۵ هـ ق) (تعطیل) و روز جهانی مستضعفان

۳۱	۲۹	۲۸	۲۷	۲۶	۲۵	۲۴	۲۳	۲۲	۲۱	۲۰	۱۹	۱۸	۱۷	۱۶
۲۳	۲۱	۲۰	۱۹	۱۸	۱۷	۱۶	۱۵	۱۴	۱۳	۱۲	۱۱	۱۰	۹	۸
شنبه	یکشنبه	دوشنبه	سه‌شنبه	چهارشنبه	پنجشنبه	جمعه	شنبه	یکشنبه	دوشنبه	سه‌شنبه	چهارشنبه	پنجشنبه	جمعه	شنبه

۲۱

چهارشنبه

2014  
11 / June

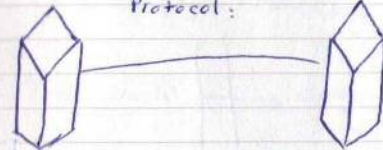
۱۴۲۵  
شعبان / ۱۳

در استراحت

نمایشی محیط استراحت ← Deployment Diagram

Connection type:

Protocol:



Node1

Node2

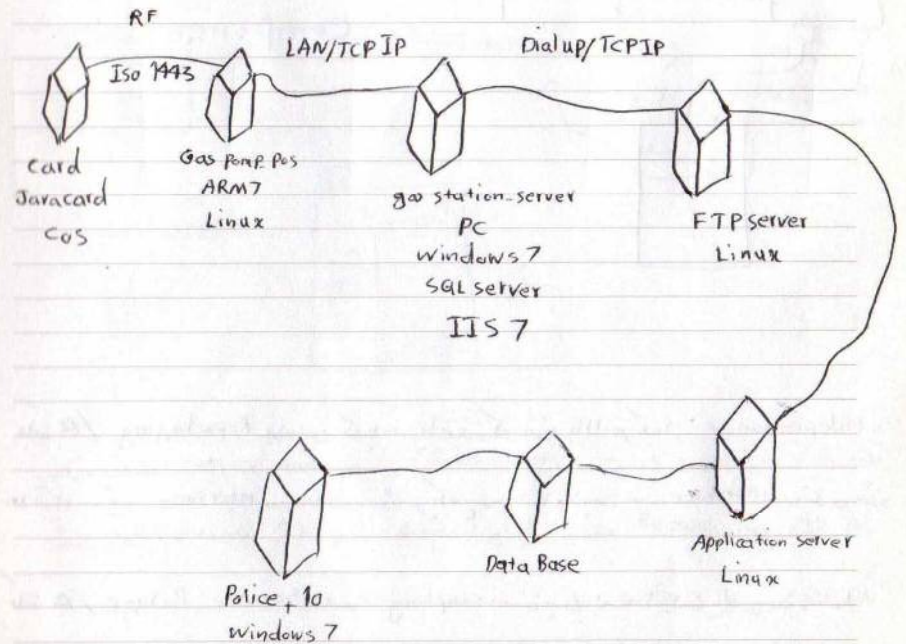
Node name:

OS:

Hardware:

Middleware:

کارت سوخت و کپی بردن



۱۵	۱۴	۱۳	۱۲	۱۱	۱۰	۹	۸	۷	۶	۵	۴	۳	۲	۱
۲۳	۲۲	۲۱	۲۰	۱۹	۱۸	۱۷	۱۶	۱۵	۱۴	۱۳	۱۲	۱۱	۱۰	۹
شنبه	یکشنبه	دوشنبه	سه‌شنبه	چهارشنبه	پنجشنبه	جمعه	شنبه	یکشنبه	دوشنبه	سه‌شنبه	چهارشنبه	پنجشنبه	جمعه	شنبه



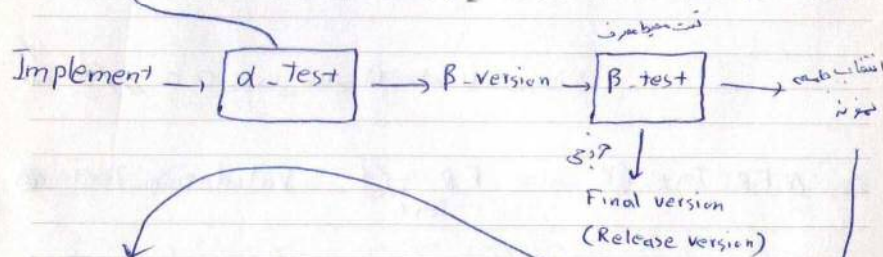
احتمال دیگر : Usecase View

افزانه کردن یک دید دیگری از طریق نمودارهای Sequence امکان می شود.

Software - Test

د، α-test، ε-ارزود  
خطا با شناسایی می شوند

تجویل دادن نرم افزار بدون تست دیگر فرما مثل شیرجودون درون استراکب است.

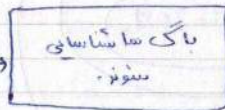


تدوین سناریوهای تست

اجرای آزمون‌ها

جمع آوری مشکلات (Defect)

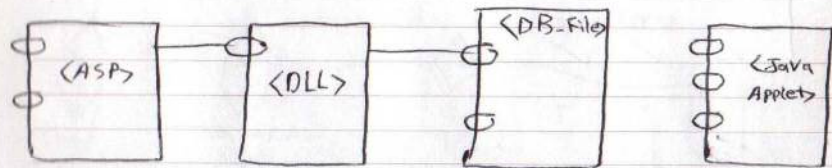
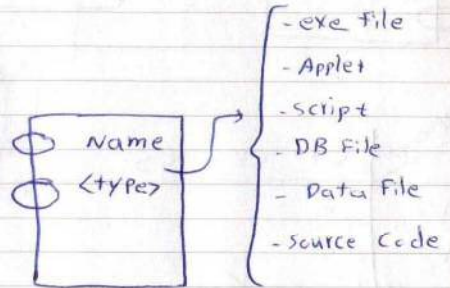
تحلیل مشکلات



در باره نیازها

رابطه نویسی از ابتدا شروع می شود.

- Java - Applet
- C/Qt
- C#.NET/SQL server
- Eclipse (Java)
- Oracle



نیازهای N.F

افزانه کردن Component های N.F بنی Node ها بر اساس بعضی نیازهای NF

نکته اهمیت، سرعت، دسترسی

مثلاً قابل های ابطالی هر یک ساعت فروش شوند و اهمیت کارت هار Pos تا حفظ شود

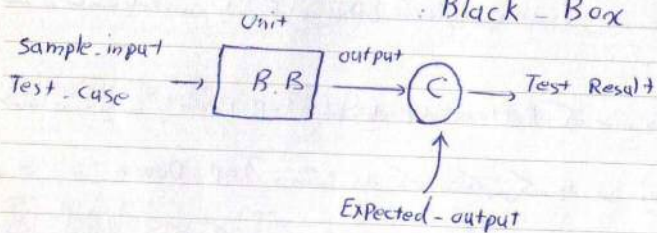
نکته / برای مدیریت APP server های Load Balancer (موزون)



### Black-Box Testing (1: Unit Test)

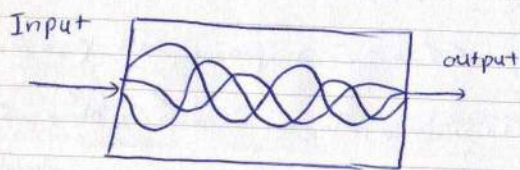
### White-Box Testing (2)

### Black-Box Testing



### White-Box Testing

پیدا کردن گرهایی است که با تحلیل مسیر گراف، تمامی مسیرهای داخل شناختنی می شود و آنها را تست می کنیم تا از درست کار کردن آنها مطمئن شویم.



### Test-case Generation

الگوریتمهایی است که sample input های خوبی را برای test ایجاد می کند تا بتوانیم نتیجه بگیریم که تقریباً تمام Unit ها در تمام مواقع درست کار می کنند.

نقطه /  $\star$   $\rightarrow$  Unit test دو پالشی داریم (1) شناختنی (Test-case ها دوری) Expected-Test

معمولاً ممکن است خود این Expected-Test ها اشتباه باشند.

روز جهاد کشاورزی (تشکیل جهاد سازندگی به فرمان حضرت امام خمینی (ره) - ۱۳۵۸ هـ.ش) - روز جهانی بیابان زدایی



### Verification Test (1: $\alpha$ -test)

### Validation Test (2)

### Unit Test (1: Verification Test)

### Integration Test (2)

نقطه /  $\star$   $\rightarrow$  O.O مخرهای کلاس ها، Unit های باشند.

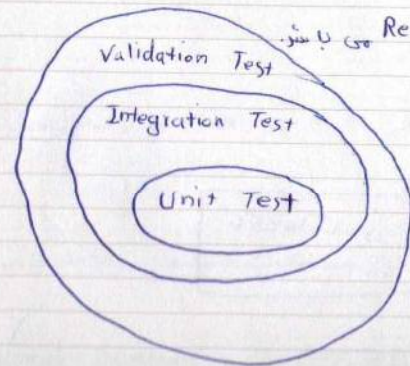
### Validation Test (1: F.R Test)

### (2: N.F.R Test)

مطالعه Unit Test مربوط به کد نویسی (Implement) است و خطاهای Integration

test مربوط به Design می باشد و خطاهای Validation Test مربوط به Analysis

، Requirements Analysis می باشد.



نقطه /  $\star$  هر چه زمان validation Test بیشتر است.



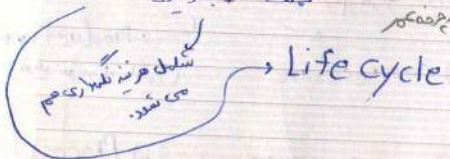
CAPITAL = CAPEX  $\star$   
EXPENSE  $\leftarrow$  شامل هزینه تابلای هم می شود.

Difference between life & lifecycle  $\star$

۱۳۲۵ ۲۱ جمادی الاول  
2014 March / 23

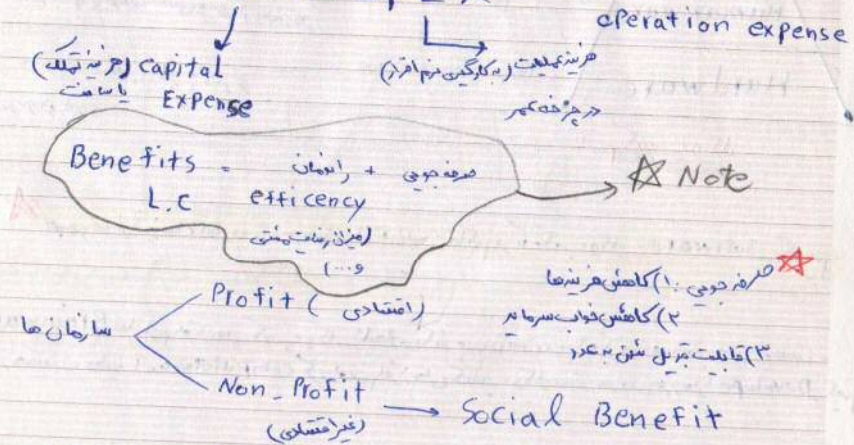
operation = OPEX  $\star$   
Expense

یکشنبه  
تعلیل  
Cost = Benefit  $\gg$  Cost: مقرون به صرفه  
تبلای و بلگرگی Cost = Benefit  $\gg$  Cost: مقرون به صرفه



Life cycle, Life  $\star$  Life: مشخص از بین می رود و مشخص بر روی پای آن برای یکدیگر مثل خانه  $\leftarrow$  مثل دلی Lifecycle مقرون به صرفه است مثل خانه کلنگی  $\leftarrow$  خانه موزان عمری هم اقرارها و حق تمام می شود که نیاز است یکسره اقرار از بین برود.

Cost = CAPEX + OPEX



Economic Feasibility Study  $\star$  به تعادل هزینه و فایده عمل می شود و شروع پروژه امکان نظر سنجی اقتصادی یا

کار: صرف: حضور پروژه و صنعتی شروع نشود. مطالعه امکان نظر سنجی

عید نوروز (تعلیل) - روز جهانی هواشناسی

۱۶	۵	۱۷	۶	۱۸	۷	۱۹	۸	۲۰	۹	۲۱	۱۰	۲۲	۱۱	۲۳	۱۲	۲۴	۱۳	۲۵	۱۴	۲۶	۱۵	۲۷	۱۶	۲۸	۱۷	۲۹	۱۸	۳۰	۱۹	۳۱	۲۰
شنبه	یکشنبه	دوشنبه	سه شنبه	چهارشنبه	پنجشنبه	شنبه	یکشنبه	دوشنبه	سه شنبه	چهارشنبه	پنجشنبه	شنبه	یکشنبه	دوشنبه	سه شنبه	چهارشنبه	پنجشنبه	شنبه	یکشنبه	دوشنبه	سه شنبه	چهارشنبه	پنجشنبه	شنبه	یکشنبه	دوشنبه	سه شنبه	چهارشنبه	پنجشنبه	شنبه	یکشنبه

تعلیل نوروز (۱)  $\star$   $\leftarrow$  کار زیاد با کدی  $\leftarrow$  بالابردن Cost پروژه  
تعلیل نوروز (۲)  $\leftarrow$  عدم خواب سرمایه  $\leftarrow$  90% done 90% overtime  $\leftarrow$  شنبه پروژه ها

2014 22 / March  
۱۳۲۵ جمادی الاول / ۲۰



۱) جدول بودجه (نیم اقرار: ۱)  $\leftarrow$  کارها و دو بار کاری یعنی کارهایی که قبلاً در یک پروژه بود و دوباره از نو نوشته می شود که این کار Cost پروژه را بالا می برد.  
۲) عدم اتمام به موقع نیم اقرار مناسی:  
پروژه: حاصل جمع کارهای تمام نمی شوند (Remain / done 90%) یعنی آن ۱۰٪ بقیه وقت به موقع تمام نمی شود و پروژه ها overtime می شوند و Time از دست می رفت.

۳) بصیرگی: تبلای نیم اقرار: تغییر در تصمیمی از نیم اقرار کل سیستم را بهم می ریخت. در آن سالها هزینه تبلای نیم اقرارها  $\leftarrow$  ابرام هزینه تبلای بود.  
۴) مناسی نیم اقرار: آنچه معای از روش ها و تبلای های معتبر مناسی به منظور توسعه و بهره برداری یک نیم اقرار (من در مناسی ماقت و تولید سایر محصولات این تبلای ها جواب در ماقت).  
۵) مناسی ماقت و تولید یعنی یک سری روشی ماقت و تولید که سه هدف دارند به منظور تولید و بهره برداری

نیم اقرار: (۱) به موقع Time: پروژه با on Time باشد  
Actual Time  $\leq$  planned Time  
(زمان برآوردی) (زمان واقعی)

۲) Quality: کیفیت یک امر نسبی است که از دید مصرف کننده تعریف می شود.

Request + Need = Requirement  $\star$   
Quality  $\star$   $\leftarrow$  قایلین نیاز مندی های یک مشتری  $\leftarrow$  کیفیت  
Time  $\star$   $\leftarrow$  ۱- به موقع  
Quality  $\star$   $\leftarrow$  ۲- اهداف مناسی  
Benefits  $\gg$  Cost  $\star$   $\leftarrow$  ۳- ساخت و تولید پروژه  
(تعلیل) هجوم مأموران ستم شاهی پهلوی به مدرسه فیضیه قم (۱۳۲۲ هـ ش) - آغاز عملیات فتح المبین (۱۳۶۱ هـ ش) - روز جهانی آب  
Requirement = Request + Need  $\rightarrow$  Cost = CAPEX + OPEX

۱۵	۱۴	۱۳	۱۲	۱۱	۱۰	۹	۸	۷	۶	۵	۴	۳	۲	۱
شنبه	یکشنبه	دوشنبه	سه شنبه	چهارشنبه	پنجشنبه	شنبه	یکشنبه	دوشنبه	سه شنبه	چهارشنبه	پنجشنبه	شنبه	یکشنبه	دوشنبه