Systems Analysis and Design With UML 2.0

An Object-Oriented Approach, Second Edition

تهیه کننده : سید رضا شاه امیری

کارشناس ارشد مهندسی نرم افزار

Chapter 1: Introduction to Systems Analysis and Design

Alan Dennis, Barbara Wixom, and David Tegarden
© 2005
John Wiley & Sons, Inc.

Translated by : Reza Shahamiri
Email Address 1 : RShahamiri@yahoo.com
Email Address 2 : Reza@RSH.ir
Web site : www.RSH.ir
Group Home page : http://groups.yahoo.com/group/SADU2
Group Email address : SADU2@yahoogroups.com

مقدمه

خیلی از سیستمها در فرآیند ساخت با شکست مواجه میشوند و زیرا تحلیلگر و سازنده به دنبال ساخت یک سیستم فوق العاده هستند بدون اینکه درک درستی از سیستم و سازمان داشته باشند .

هدف اصلی تحلیلگر انجام کاری است که برای سازمان مربوطه ارزشمند باشد .

چرخه حیات توسعه سیستم

SDLC (The System Development Lifi Cycle)

چرخه حیات توسعه سیستم فرآیندی است که در طی آن مشخص میکنیم که :

۱ – یک سیستم اطلاعاتی (Information System) چگونه میتواند نیازهای تجاری را با ساخت سیستم اطلاعاتی برطرف کند .

۲ – این سیستم چگونه طراحی شود .

۳ – چگونه ساخته شود .

۴ – چگونه در اختیار کاربر قرار بگیرد .

نقش کلیدی را در SDLC تحلیلگر سیستم بازی میکند وظیفه او تحلیل تجارت , شناسایی نقاط بهبود فرآیند تجارت و طراحی سیستم اطلاعاتی میباشد .

SDLC از چهار فاز اصلی تشکیل شده است که هر فاز شامل یکسری مراحل است که هر مرحله توسط یکسری از تکنیکهای خاص انجام میشود و ماحصل این مراحل deliverable ها میباشند . اینها مصنوعات ساخته شده توسط افرادی هستند که در SDLC مکشغول بکارند و در حقیقت خروجیهای هر فاز میباشند . هر فاز از deliverable های فازهای قبلی استفاه میکند .

باید توجه شود که تکامل در SDLC به صورت تدریجی انجام میشود . یعنی در هر فاز نتایج بدست آمده در فازهای قبلی میتوانند تغییر کرده و بهبود یابند .

برای ساخت یک سیستم اطلاعاتی توسط SDLC میتوان به صورت زیر عمل کرد:

۱ به ترتیب فازهای SDLC را طی کنیم

۲ در هر فاز deliverable ها را تولید کنیم

۳ از deliverable های بدست آمده برای پیاده سازی استفاده کنیم

٤ سیستم اطلاعاتی را بسازیم

۵ سیستم را پالایش کنیم

چهار فاز اصلی در SDLC

1 -Planning (طرح ریزی)

Why build the system?

2 -Analysis(تحلیل)

Who, what, when, where will the system be?

3 -Design(طراحی)

How will the system work?

4 -Implementation(پیاده سازی)

۱ – طرح ریزی (Planning)

در مرحله طرح ریزی مشخص میکنیم که " چرا یک سیستم اطلاعاتی باید ساخته شود " و همچنین اعضای تیم پروژه را مشخص میکنیم . این فاز از دو مرحله تشکیل شده است :

الف – شروع پروژه (Project Initiation): در این فاز ما مشخص میکینم که چه منافع تجاری برای سازمان ارزشمند هستند(Identifying business value) . میتوان با پاسخ به سئوال " چگونه میتوان هزینه های سازمان را کاهش داد و منافع بیشتری برای سازمان فراهم کرد ؟ " این منافع را شناسایی کرد . نتیجه این سئوال را در گزارشی بنام درخواست سیستمها (System Request) ارائه میدهیم . در این گزارش بطور خلاصه بیان میکنیم که نیازمندیهای تجاری سازمان کدامند و ساخت سیستم اطلاعاتی چه منافعی را برای سازمان به ارمغان می آورد . سپس به تحلیل امکان سنجی (Feasibility analysis) می پردازیم که در آن مشخص میکنیم آیا ساخت سیستم از لحاظ تکنیکی , اقتصادی و سازمانی انجام پذیر است یا خیر . سپس درخواست سیستمها و نتیجه امکان– سنجی را به یک انجمن بنام کمیته تایید (Approval committee) برای گرفتن مجوز انجام کار ارسال میکنیم.

ب – مدیریت پروژه(Project management) : در صورت تایید پروژه وارد مرحله مدیریت پروژه میشویم که در این مرحله طرح کاری پروژه , کارمندان پروژه و کنترل پروژه مشخص و انجام میشوند . نتایج این مرحله در غالب گزارشی بنام طرح پروژه (Project Plan) جمع آوری میشوند . در این گزارش مشخص میکنیم که چگونه تیم پروژه عملیات توسعه سیستم را انجام میدهد .

۲ – تحلیل (Analyzing)

در فاز تحلیل مشخص میکنیم که :

الف – چه کسانی از سیتم استفاده میکنند .

ب – سیتم چه کاری انجام میدهد .

ج – سیتم چه زمانی و در کجا قابل استفاده خواهد بود .

در طی این فاز تیم پروژه به بررسی سیستمهای موجود پرداخته , نقاط بهبود سیستم را پیدا و سیستم جدید را شناسایی میکند . این فاز از سه مرحله تشکیل شده است :

الف – تهیه یک استراژی تحلیل برای مشخص کردن فعالیتهای تیم پروژه .

ب –جمع آوری نیازمندیها (Requirement gathering) و تهیه مدل تحلیل (Analysis model) که در آن مشخص میشود فعالیتهای تجاری در سیستم آینده به چه صورت خواهند بود .

ج –تهیه گزارش پیشنهاد سیستم (System proposal) که در آن کلیه مفاهیم سیستم و مدلها مشخص میشوند و این گزارش را در اختیار تصمیم گیرندگان سیستم قرار داده تا تصمیم نهایی را برای ساخت سیستم بگیرند .

۳ – طراحی (Design)

در فاز طراحی چگونگی عملکرد سیستم را توسط تعیین کردن مشخصات سخت افزار , نرم افزار , زیرساختهای شبکه ای , واسط کاربری , فرمها و گزارشها و ... مشخص میکنیم .این فاز از چهار مرحله تشکیل شده است :

الف – تهیه استراتژی طراحی که در آن مشخص میشود که سازمان توسط برنامه نویسان خود به ساحت سیستم بپردازد و یا از سازمانهای خارجی برای ساخت سیستم استفاده کند و یا اقدام به خرید بسته نرم افزاری موجود نماید .

ب – تعیین معماری طراحی (Architectural design) که در آن زیرساختهای نرم افزاری , سخت افزاری و شبکه ای مشخص می شود . سپس به طراحی واسط کاربری میپردازیم (Interface design)

ج – ساخت خصوصیات فایلها و پایگاه داده که در طی آن مشخص میکنیم چه داده هایی و در چه زمانی ذخیره میشوند .

د – تهیه برنامه طراحی (Program design) که در طی آن مشخص میکنیم چه برنامه هایی باید نوشته شوند و هر کدام از این برنامه ها چه کاری انجام میدهند .

در نهایت تمامی موارد گفته شده در غالب گزارشی بنام توصیف سیستمها (System specification) به تیم برنامه نویسان و پیاده سازان ارائه میشود .

۴– پیاده سازی (Implementation)

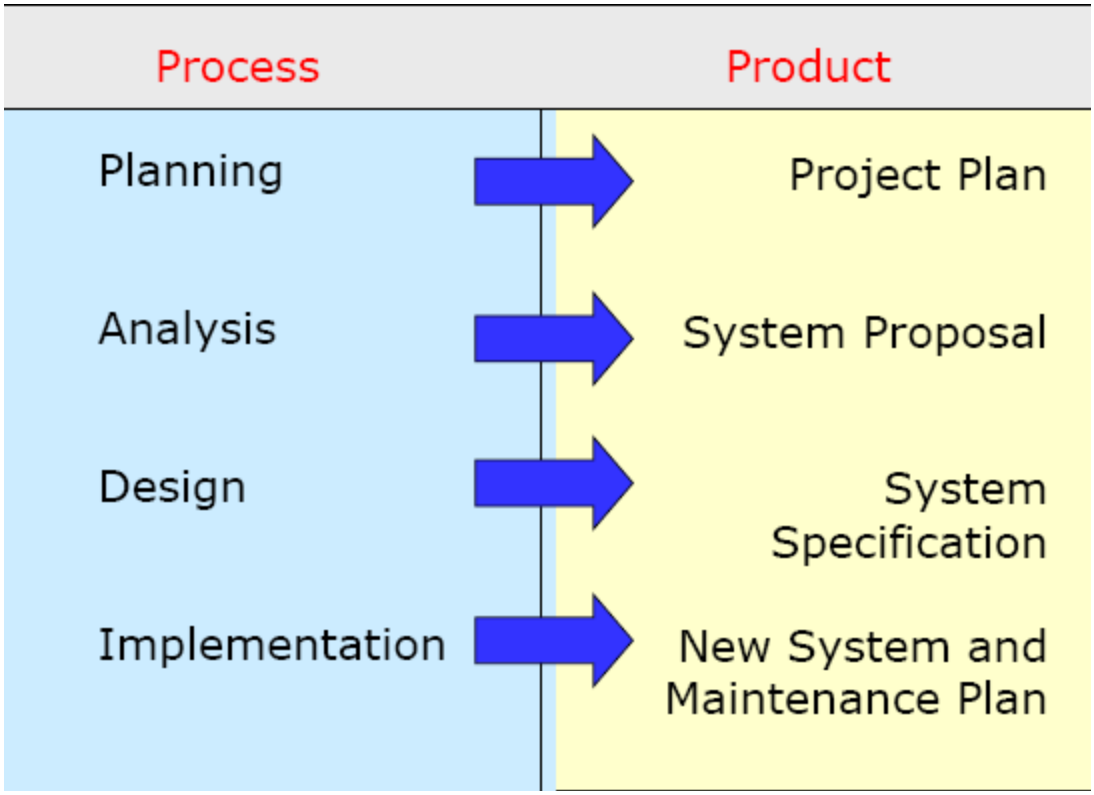
در این فاز سیتم ساخته و نصب میشود . این فاز از سه مرحله تشکیل شده است :

الف – ساخت وتست سیستم (System construction)

ب – نصب سیستم (System installation)

ج – تهیه طرح پشتیبانی (Support plan)

مصنوعات (Deliverable)تولید شده در هر فاز به شرح زیر است :



متدولوژیهای توسعه سیستمها :

متودولوژی چیست ؟

متودولوژی (روش شناسی) مجموعه ای منظم از شیوه ها , ایده ها , استانداردها و راهکارهایی است که برای پیاده سازی SDLC بکار میروند و در آنها ترکیبها و سریهای مختلفی از مراحل SDLC مشخص شده است .

بطور کلی متودولوژیها به سه دسته کلی تقسیم میشوند :

متودولوژیهای ساخت یافته

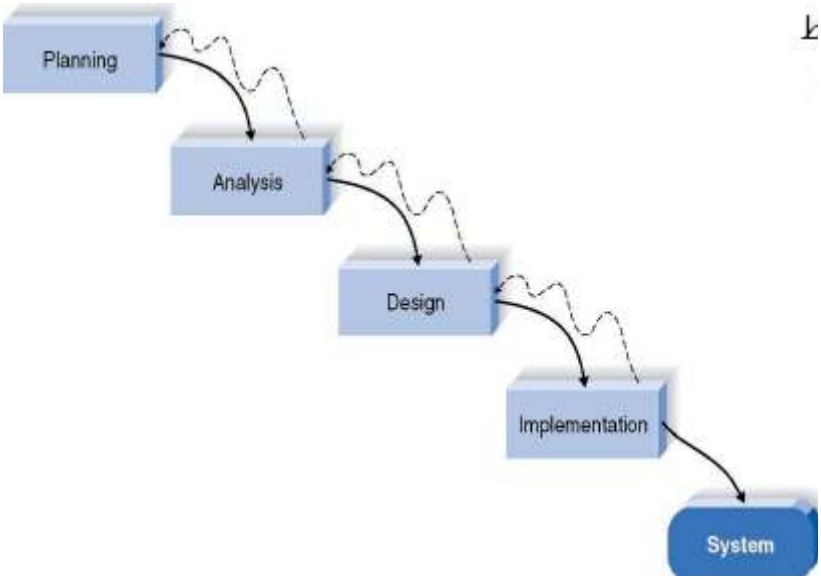
متودولوژیهای RAD

متودولوژی Agile

۱- متودولوژیهای ساخت یافته

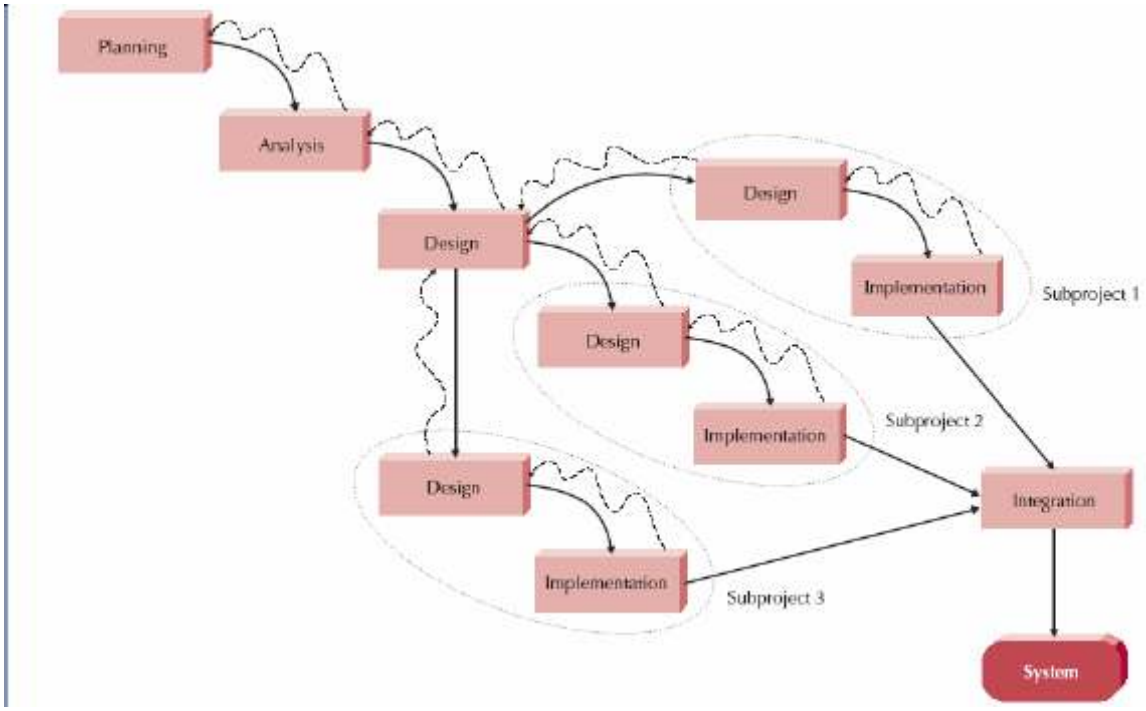
یکی از متودولوژیهای قدیمی است که امروزه در سیستمهای بزرگ عملا غیر قابل استفاده است . در این متودولوژی مراحل مختلف SDLC به ترتیب اجرا میشوند . دو نمونه از این متودولوژی , متودولوژی آبشاری و دیگری متودولوژی موازی است.

الف – متودولوژی آبشاری



ب – متودولوژی موازی

برای رفع فاصله زمانی زیاد مابین فاز تحلیل تا تحویل سیستم این متودولوژی پیشنهاد شد . در این متودولوژی بعد از پایان فاز تحلیل , مرحله طراحی به چند قسمت تقسیم شده که هر قسمت توسط یکسری افراد به طور موازی و هم زمان انجام میشود . بعد از پایان پیاده سازی هر قسمت , یک فاز برای یکپارچه سازی قسمتهای ساخته شده سیستم در نظر گرفته شده است.



۲- متودولوژیهای RAD (Rapid Application Development)

در متودولوژی RAD سعی بر این است که تا حد ممکن نسخه های اولیه سیستم سریعتر به دست کاربران برسند تا کاربر بتواند با استفاده از آنها به فهم بهتری از سیستم دست یابد و خواسته های خود را سریعتر به توسعه دهندگان سیستم اعلام دارد .

سه نمونه مختلف از این متودولوژی به شرح زیرند .

الف – توسعه فازی (Phase development)

ب- پروتوتایپ (Prototyping)

ج- پروتوتایپ دور انداختنی (Throw-Away Prototyping)

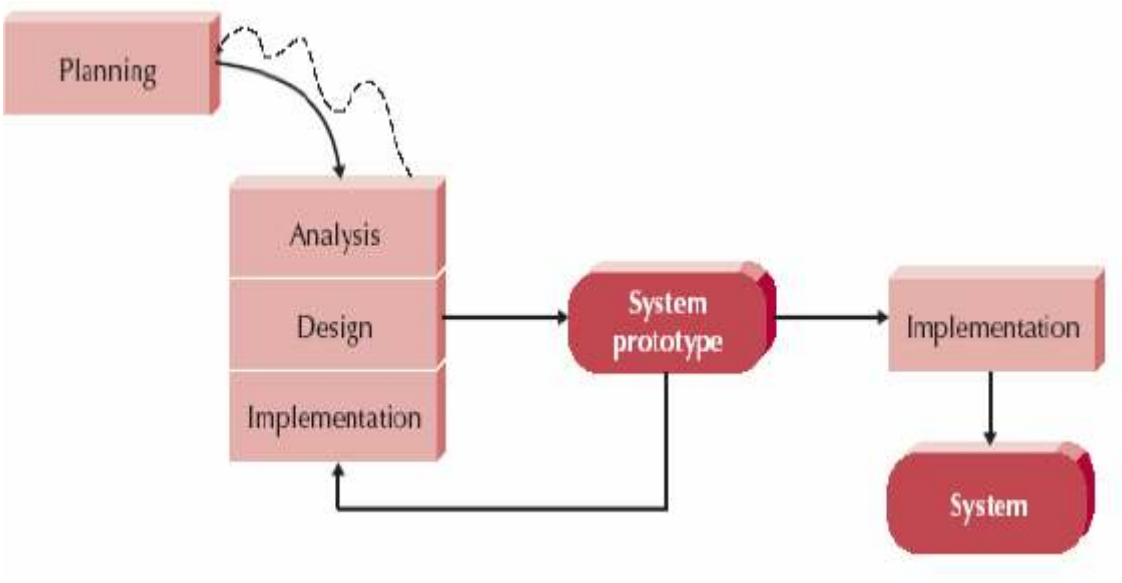
الف – Phase Development

در این متودولوژی کل سیستم را به نسخه های مختلفی تقسیم می کنیم . معمولا نیازمندیهای اصلی و مهم سیستم در نسخه اولیه گنجانده شده و در نسخه های بعدی ویژگیهای دیگر و تغییرات مورد نظر کاربر اعمال میشود . به محض پایان هر نسخه ساخت نسخه بعدی آغاز میشود .

ب – Prototyping

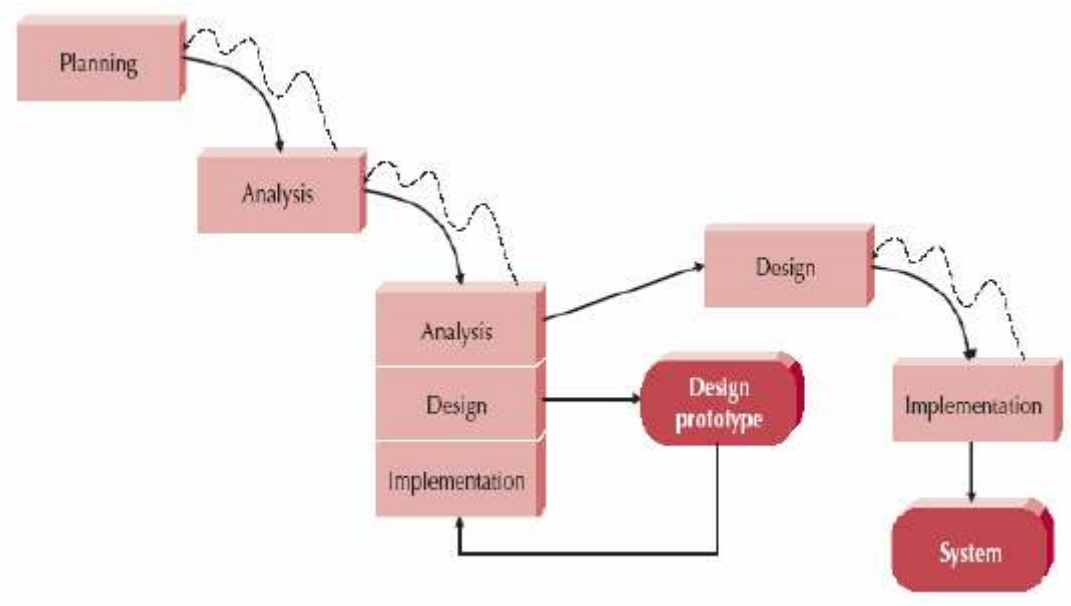
پروتوتایپ به معنی ساخت یک نسخه ازمایشی است .

در این مدل فازهای تحلیل , طراحی و ساخت با هم انجام میشوند و این فازها آنقدر تکرار میشوند تا سیستم کامل شود . در ابتدای کار یک تجزیه و تحلیل سریع و مقدماتی انجام شده و پروتوتایپ اولیه سیستم ساخته میشود . سپس این پروتوتایپ در اختیار کاربر قرار گرفته و مجددا فازهای گفته شده تکرار میشوند. پروتوتایپ در هر مرحله کاملتر میشود تا زمانیکه کل سیستم بر اساس آن پروتوتایپ ساخته شود .



ج –پروتوتایپ دور انداختنی Throw-Away prototype

این متودولوژی شبیه به متودولوژی قبلی میباشد با این تفاوت که در این متودولوژی پروتوتایپ ساخته شده سیستم مورد نظر نیست , بلکه فقط شبیه سازی شده قسمتهای مختلفی از سیستم است که به این منظور ساخته شده اند که کاربر با آنها کارکرده و بدین وسیله تحلیلگر بتواند نیازمندیهای کاربران را بهتر درک کند . در نهایت این پروتوتایپ به دورانداخته شده و سیستم از ابتدا طراحی میشود .



انتخاب یک متودولوژی مناسب

تمامی متودولوژیهای گفته شده مزایا و معایبی دارند و هیچ متودولوژی به عنوان بهترین وجود ندارد . بنابراین بر اساس یکسری از پارامترها و نوع پروژه باید متودولوژی مناسبی انتخاب شوند که این پارامترها به شرح زیرند:

- میزان وضوح نیازمندیهای کاربران
- آشنایی با تکنولوژی
- پیچیدگی سیستم
- قابلیت اطمینان سیستم
- لزوم تحویل سیستم در یک بازه زمانی کوتاه مدت
- با برنامه پیش رفتن

با توجه به جدول زیر میتوان نتودولوژی مناسب را انتخاب کرد :

Ability to Develop Systems	Structured Methodologies			RAD Methodologies		Agile Methodologies
	Waterfall	Parallel	Phased	Prototyping	Throwaway Prototyping	XP
with Unclear User Requirements	Poor	Poor	Good	Excellent	Excellent	Excellent
with Unfamiliar Technology	Poor	Poor	Good	Poor	Excellent	Poor
that are Complex	Good	Good	Good	Poor	Excellent	Poor
that are Reliable	Good	Good	Good	Poor	Excellent	Good
with a Short Time Schedule	Poor	Good	Excellent	Excellent	Good	Excellent
with Schedule Visibility	Poor	Poor	Excellent	Excellent	Good	Good

مهارتهای تحلیلگر

مهارتهایی که یک تحلیلگر باید داشته باشد به شش دسته تقسیم میشود :

- مهارتهای تکنیکی : تحلیلگر برای فهم تکنیکهای استفاده شده در سازمان و تکنولوژیی که برای ساخت سیستم استفاده خواهد شد و اینکه به چه صورت با استفاده از این تکنیکها و تکنولوژیها راه حل مناسبی برای حل مشکلات سازمان بیاید , احتیاج به مهارتهای تکنیکی دارد .
- مهارتهای تجاری (Business skills) : تحلیلگر برای اینکه بداند چگونه یک سیستم اطلاعاتی میتواند نیازهای تجاری یک سازمان را برآورده کند به مهارتهای تجاری نیازمند است .
- مهارتهای تحلیلی (Analytical skills) : تحلیلگر باید توسط مهارتهای تحلیلی خود بتواند هم در سطح سازمانی و هم در سطح پروژه مشکلات را حل کند.
- مهارتهای شخصی : تحلیلگر باید علاوه بر مهارتهای علمی , مهارتهای شخصی هم مثلا برای بهتر ارتباط برقرار کردن با مسئولین سازمان و برنامه نویسان , داشته باشد.

- مهارتهای مدیریتی : تحلیلگر باید از تکنیکهای مختلف مدیریتی آگاه باشد تا بتواند تیم پروژه را در مسیر مناسب هدایت کند .
- مهارتهای اخلاقی : تحلیلگر باید با حسن اخلاق , شجاعت و راستگویی در کل پروژه باعث افزایش اطمینان مابین افراد درگیر در پروژه شود .

در جدول زیر نقشهایی که تحلیلگر میتواند در SDLC بازی کند ذکر شده است .

Role	Responsibilities
Business analyst	Analyzing the key business aspects of the system Identifying how the system will provide business value Designing the new business processes and policies
Systems analyst	Identifying how technology can improve business processes Designing the new business processes Designing the information system Ensuring that the system conforms to information systems standards
Infrastructure analyst	Ensuring the system conforms to infrastructure standards Identifying infrastructure changes needed to support the system
Change management analyst	Developing and executing a change management plan Developing and executing a user training plan
Project manager	Managing the team of analysts, programmers, technical writers, and other specialists Developing and monitoring the project plan Assigning resources Serving as the primary point of contact for the project

Chapter 2: Introduction to Object-Oriented Systems Analysis and Design with the Unified Modeling Language, Version 2.0

سیستمهای شیئ گرا

(Object Oriented Systems)

تمرکز سیستمهای شیئ گرا در دسته بندی کردن ساختارها و رفتارهای سیستم اطلاعاتی در غالب اشیاء (Object) میباشد . در این فصل ما به توصیف مفاهیم و ویژگیهای اصلی سیستمهای شیئ گرا میپردازیم و مفاهیم زیر را تعریف میکنیم :

- کلاس (Class)
- اشیاء (Object)
- رویه (Method)
- پیام (Message)
- بسته بندی (Encapsulation)
- پنهان سازی اطلاعات (Information Hiding)
- ارث بری (Inheritance)
- چند ریختی (Polymorphism)

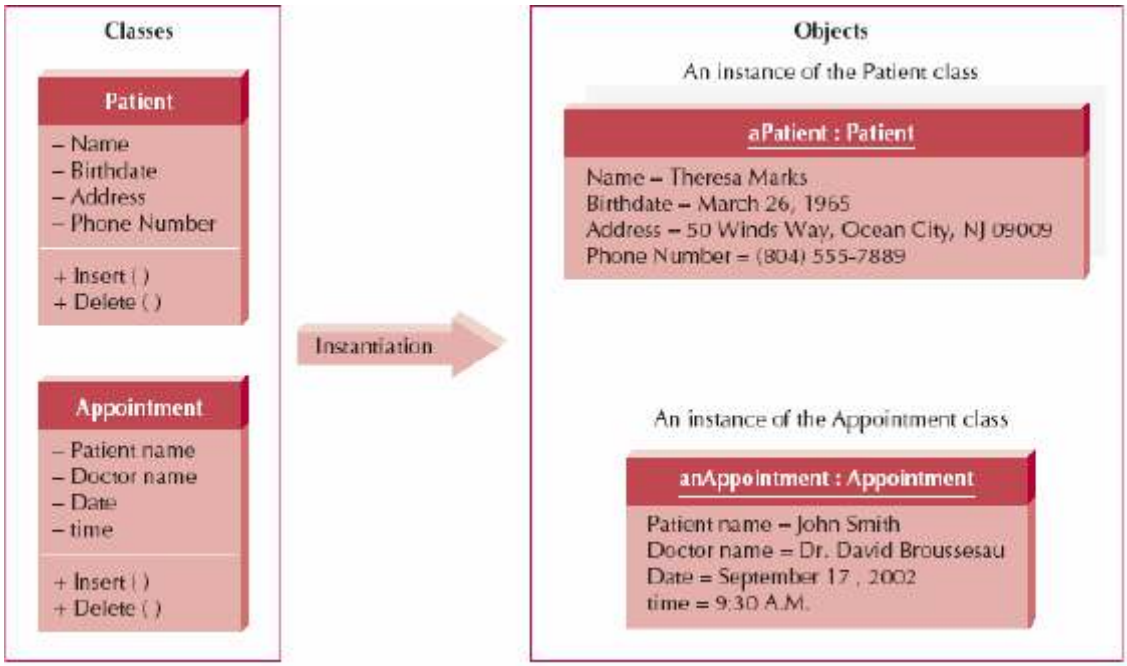
کلاسها و اشیاء

کلاسها یک غالب کلی برای تعریف و یا ساخت اشیاء میباشند . به عنوان مثال میتوانیم کلاسی تعریف کنیم که شامل تمامی اطلاعات لازم در مورد یک بیمار باشد (نام , سن و ...) و همینطور تمامی متدهاو رفتارهای آن بیمار (گرفتن نوبت , معاینه شدن و ...) .

اشیاء نمونه هایی از کلاسها میباشند که مقادیر ویژگیهای آنها با هم متفاوت است . مثلا ممکن است یک نمونه از کلاس بیمار بنام علی و نمونه ای دیگر بنام رضا داشته باشیم که علی و رضا دو شیئ از کلاس بیمار میباشند . پس اشیاء شامل اشخاص , مکانها , رخدادها و یا هر چیز دیگری میباشند که میخواهیم درباره آن اطلاعات جمع آوری و یا نگهداری کنیم .

هرشیئ دارای یکسری ویژگیها (Attribute) میباشد که توصیف کننده اطلاعاتی در مورد آن شیئ است. مثلا نام بیمار , تاریخ تولد , سن و ... . وضعیت هر شیئ توسط مقادیر ویژگیهای آن شیئ و روابطش با دیگر اشیاء در یک زمان خاص مشخص میشود .

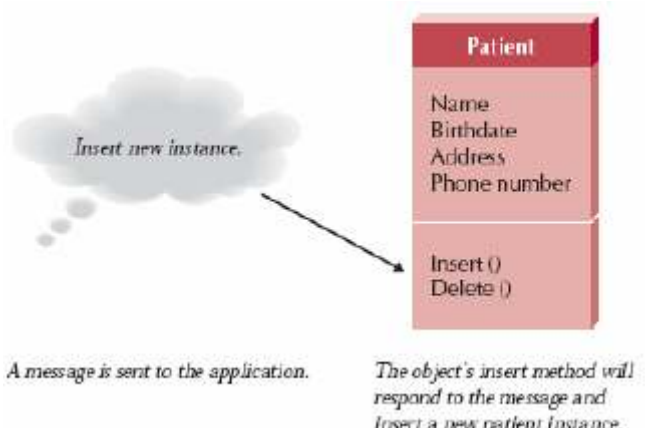
همچنین هر شیئ شامل رفتارهای (Behavior) خاصی میباشد که این رفتارها نشان دهنده کارهایی هستند که آن شیئ میتواند انجام دهد .



متدها و پیامها

متدها همان رفتارهای اشیاء هستند . یعنی فعالیتها و رویه هایی که یک شیئ میتواند انجام دهد .

پیامها , اطلاعاتی میباشند که برای فعال سازی یک متد به اشیاء ارسال میشوند (Trigger) . مانند فراخوانی تابعی از یک شیئ که توسط شیئی دیگری انجام شود .



بسته بندی و پنهان سازی اطلاعات

بسته بندی زمانی است که ما ویژگیها و متدهای مربوط به یک چیز را در غالب یک شیئ ارائه کنیم . یا به عبارت دیگر , مجموعه فرآیندها و داده های یک موجودیت را در غالب یک شیئ قرار دهیم .

پنهان سازی اطلاعات بدین معنی است که فقط اطلاعاتی از یک شیئ را در اختیار اشیاء دیگر قرار دهیم که برای آن شیئ قابل استفاده است نه بیشتر . یا به عبارت دیگر , یک شیئ از اینکه چگونه اعمال مختلف درون اشیاء دیگر انجام میشود و یا متدها در آن اشیاء چگونه اجرا میشوند, باخبر نباشد . فقط در صورت امکان میتوان آن متدها را فراخوانی کرد و از نتایج تولید شده توسط آن استفاده کرد . یعنی استفاده از اشیاء فقط توسط فراخوانی متدهای آن امکان پذیر است .

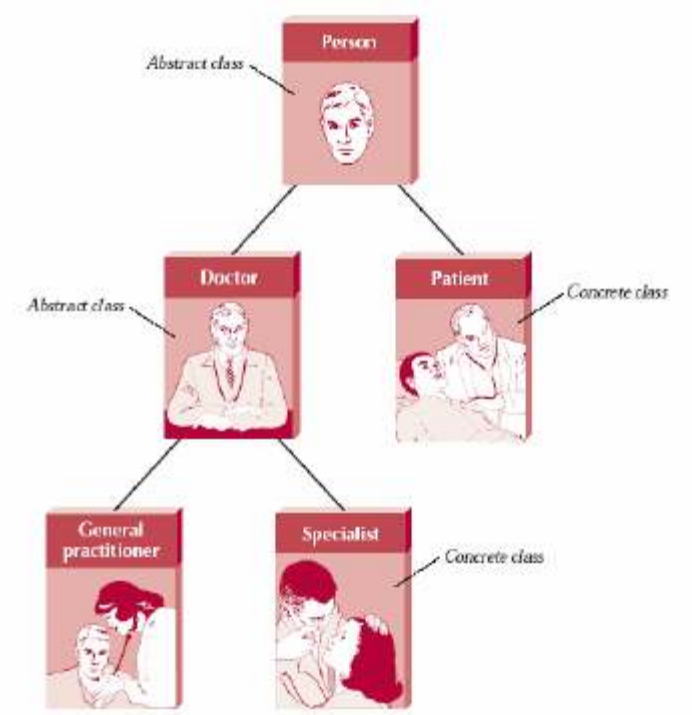
ارث بری

در خیلی از مواقع کلاسهایی وجود دارند که دریکسری از ویژگیها و متدها مشترکند و در برخی دیگر متفاوت . ما از ویژگیها و متدهای مشترک یک سوپر کلاس ایجاد کرده و برای دیگر موارد متفاوت زیر کلاسهایی ایجاد میکنیم که این زیرکلاسهای متفاوت علاوه بر خصوصیات مختص خویش , خصوصیات سوپرکلاسها را نیز به ارث میبرند .

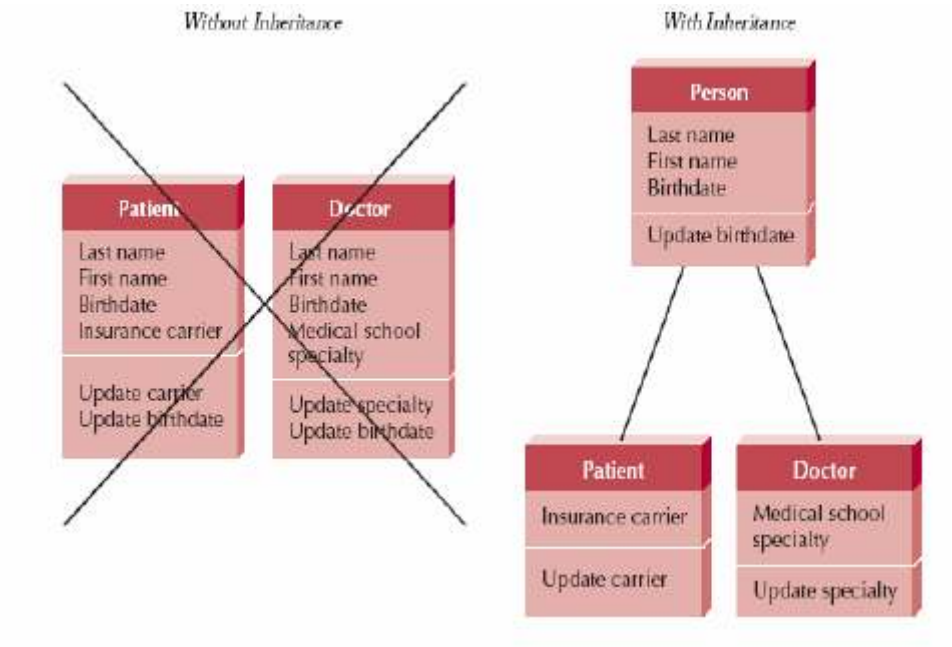
رابطه ای که مابین زیرکلاسها و سوپرکلاسها به وجود میاید , رابطه " نوعی از (a kind of) " میباشد .

کلاسهایی را که میتوان از آنها اشیائی را مشتق کرد Concrete class و آنهایی را که از آنها بیشتر به عنوان غالبی برای زیرکلاسها استفاده میشود کلاسهای انتزاع (Abstract class) مینامند .

شکل زیر مثالی از سلسله مراتب کلاس نشان داده شده است

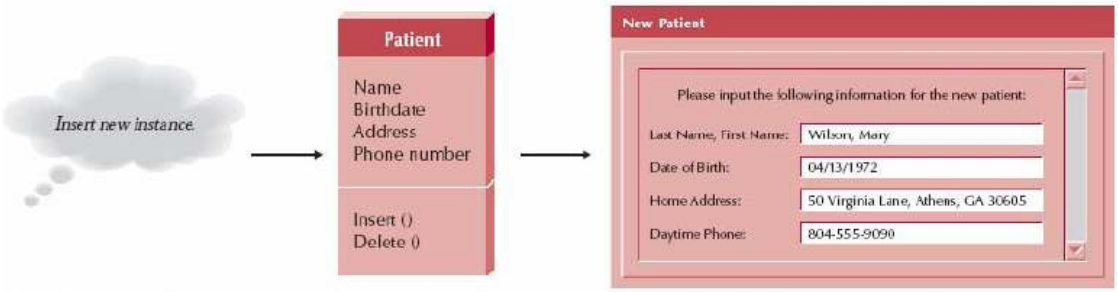


شکل زیر مثالی از ارث بری نشان داده شده است .



چند ریختی

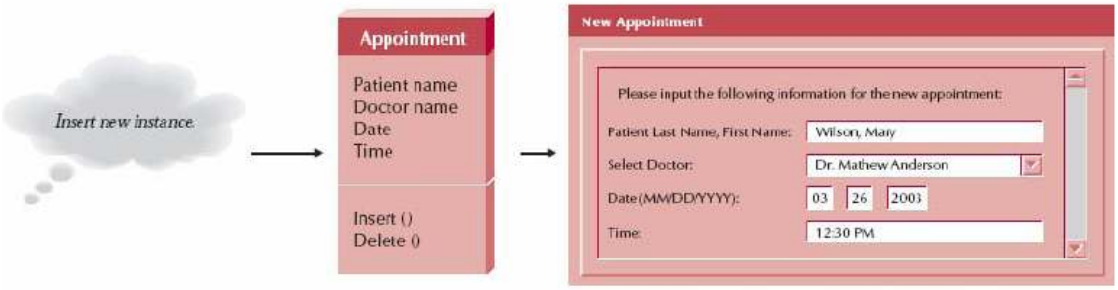
چند ریختی بدین معنی است که ارسال یک پیام به اشیاء مختلف میتواند منجر به رفتارهای متفاوتی در اشیاء مختلف شود . مثلا در صورتیکه پیام "خود را رسم کن" به یک شیئ مستطیل ارسال شود باعث رسم یک مستطیل میشود ؛ در صورتیکه اگر همین پیام را به یک شیئ دایره ارسال کنیم منجر به رسم دایره میشود .



1. An insert message is sent to the patient object.

2. The object's method responds to the message.

3. The application responds appropriately.



1. An insert message is sent to the appointment object.

2. The object's method responds to the message.

3. The application responds appropriately.

The Unified Modeling Language, Version 2.0

UML , تکنیکهای استانداردی برای رسم نمودار است که برای مدلسازی سیستمهای شیئ گرا استفاده میشود و هدف آن ارائه یک لغت نامه و تکنیکهای نموداری مشترک از مفاهیم شیئ گرایی است تا بتواند از مرحله تجزیه و تحلیل تا مرحله پیاده سازی کاربرد داشته باشد . آخرین نسخه آن , نسخه ۲ میباشد که دارای چهارده تکنیک رسم نمودار برای مدلسازی سیستم میباشد . تعدادی از نمودارها برای مدلسازی ساختار سیستم (جنبه ایستای سیستم) و تعدادی دیگر برای مدلسازی رفتارهای سیستم (جنبه دینامیکی سیستم) استفاده میشوند .

نمودارهای ساختاری (Structure Diagrams)

نمودارهای ساختاری برای نشان دادن داده ها و روابطی از سیستم استفاده میشوند که در چرخه حیاتی سیستم غیر قابل تغییرند (اصطلاحا ایستا هستند) و شامل نمودارهای زیرند :

- Class diagram
- Object diagram
- Package diagram
- Deployment diagram
- Component diagram
- Composite Structure diagram

۱- دیاگرام کلاس (Class diagram) :

هدف اصلی این نمودار ارائه لغتنامه ای است که هم توسط تحلیلگر و هم توسط کاربر مورد استفاده قرار بگیرد . این نمودار برای نشان دادن کلاسها و روابط مابین آنها در سیستم اطلاعاتی , استفاده میشود .

۲ – نمودار اشیاء (Object diagram):

در این نمودار تمامی اشیاء و روابط مابین آنها مدلسازی میشوند و جزئیات بیشتر کلاسها هم در این نمودار نشان داده میشوند .

۳ – نمودار بسته ها (Package diagram) :

از این نمودار برای گروه بندی عناصر مختلف UML در غالب یک پکیج استفاه میشود بدین منظور که بتوان عناصر را از سطح بالاتری مورد بررسی قرار داد . به عنوان مثال در سیستم مطب پزشک میتوان کلاس بیمار و بیماری را در غالب یک پکیج نشان داد .

۴ – دیاگرام گسترش (Deployment diagram):

از این نمودار برای نشان دادن روابط ما بین عناصر سخت افزاری که در زیر ساختهای فیزیکی سیستم اطلاعاتی وجود دارند استفاده میشوند .

۵ –نمودار مولفه ها Component diagram :

برای نشان دادن روابط فیزیکی مابین مولفه های نرم افزاری سیستم اطلاعاتی مورد استفاده قرار میگیرد .

۶ – نمودارهای ساختاری مرکب (Composite Structure diagram) :

در صورتیکه ساختار داخلی یک کلاس پیچیده باشد از این نمودارها استفاده میشود و برای نشان دادن روابط قسمتهای مختلف درون کلاس بکار میرود .

نمودارهای رفتاری (Behavioral diagram)

از این نمودار برای نشان دادن وجهی از سیستم استفاده میشود که در طول حیات سیستم متغیرند (اصطلاحا پویا یا دینامیک هستند) و شامل نمودارهای زیر میباشد :

1 - Activity diagram (دیاگرام فعالیت)

2 - Interaction diagram (دیاگرام تعامل)

که خود شامل دیاگرامهای زیر میباشد :

الف – دیاگرام توالی (Sequence diagram)

ب – دیاگرام ارتباطات (Communication diagram)

3 – State machine diagram

که خود شامل دیاگرامهای زیر میباشند :

الف –Behavioral State Machine diagram

ب – Protocol State Machine diagram

4 – Use case diagram

۱ – دیاگرامهای فعالیت :

این دیاگرامها شباهت زیادی به فلوچارتها دارند و برای مدلسازی فرآیندهای سیستم اطلاعاتی استفاده میشوند و مشخص میکنند که گردش کار در سیستم اطلاعاتی به چه صورت میباشد .

۲ – دیاگرامهای تعامل :

از این نمودارها برای به تصویر کشیدن تعاملات مابین اشیاء سیستم اطلاعاتی استفاده میشود و شامل نمودارهای زیر است :

الف – دیاگرام توالی : این نمودار برای نشان دادن تعاملات پویای اشیاء در سیستم اطلاعاتی استفاده میشود و تاکید ان بر وجه زمانی این تعاملات میباشد .

ب – دیاگرامهای ارتباطات : همانند نمودار توالی است با این تفاوت که بجای تاکید بر وجه زمانی , بر پیامهایی که مابین اشیاء ردوبدل میشوند تاکید دارد .

۳ – دیاگرامهای State Machine :

این نمودارها دونوعند :

الف – State machine های رفتاری : این نمودارها برای نشان دادن تغییراتی که در یک شیئ در طول چرخه حیاتش رخ میدهد , بکار میرود . در این نمودار وضعیتها و مجموعه مقادیر مختلف اشیاء مدلسازی میشوند . همچنین رخدادهایی که باعث این تغییر وضعیتها میشوند نیز نشان داده میشوند .

ب –دیاگرامهای Protocol State Machine : از این نمودار برای نشان دادن وابستگیهای مابین عناصر کلاس استفاده میشود . به عنوان مثال گفته میشود " قبل از خواندن یک فایل باید آن فایل باز شود " .

۴ – دیاگرامهای موارد کاری :

از این نمودارها برای نشان دادن تعاملات مابین سیستم اطلاعاتی و محیط پیرامون آن استفاده میشود , که این محیط میتواند شامل کاربران سیستم یا هرگونه سیستم خارجی که به نحوی با سیستم در ارتباط است , باشد . هدف اصلی این نمودار فهم نیازمندیهای سیستم اطلاعاتی است. این نمودار یکی از مهمترین نمودارهای استفاده شده در سیستمهای شیئ گرا است .

Diagram Name	Used to	Primary Phase
Structure Diagrams		
Class	Illustrate the relationships between classes modeled in the system.	Analysis, Design
Object	Illustrate the relationships between objects modeled in the system. Used when actual instances of the classes will better communicate the model.	Analysis, Design
Package	Group other UML elements together to form higher level constructs.	Analysis, Design, Implementation
Deployment	Show the physical architecture of the system. Can also be used to show software components being deployed onto the physical architecture.	Physical Design, Implementation
Component	illustrate the physical relationships among the software components.	Physical Design, Implementation
Composite Structure	Illustrate the internal structure of a class, i.e., the relationships among the parts of a class.	Analysis, Design
Behavioral Diagrams		
Activity	Illustrate business workflows independent of classes, the flow of activities in a use case, or detailed design of a method.	Analysis, Design
Sequence	Model the behavior of objects within a use case. Focuses on the time-based ordering of an activity.	Analysis, Design
Communication	Model the behavior of objects within a use case. Focuses on the communication among a set of collaborating objects of an activity.	Analysis, Design
Interaction Overview	Illustrate an overview of the flow of control of a process.	Analysis, Design
Timing	Illustrate the interaction that takes place among a set of objects and the state changes in which they go through along a time axis.	Analysis, Design
Behavioral State Machine	Examine the behavior of one class.	Analysis, Design
Protocol State Machine	Illustrates the dependencies among the different interfaces of a class.	Analysis, Design
Use-Case	Capture business requirements for the system and to illustrate the interaction between the system and its environment.	Analysis

Interaction diagrams

UML میتواند درتمامی متدولوژیهای گفته شده با یک رهیافت شیئ گرا استفاده شود که البته بیشتر از متودولوژیهای RAD استفاده میشود خصوصا توسعه فازی . ولی هر متودولژی شیئ گرایی که برای ساخت سیستمهای اطلاعاتی استفاده شود باید خصوصیات زیر را داشته باشد :

1 - Use case driven

2 - Architecture centric

3 - Iterative and Incremental

1 - Use case driven

بدین معنی است که use caseها به عنوان اصلی ترین ابزار مدلسازی برای تعریف رفتار سیستم استفاده شوند . یک use case , نحوه تعامل کاربر با سیستم را برای انجام یک کار خاص توصیف میکند و در یک زمان فقط برروی یک فعالیت متمرکز است .

2 – Architecture Centric

بدین معنی است که معماری نرم افزار و سیستم اطلاعاتی به صورت لایه به لایه باشد . یک معماری لایه به لایه حداقل از سه لایه (دید) تشکیل میشود :

الف – دید عملکردی(Functional view): توصیف کننده رفتارهای خارجی سیستم از دید کاربراست .

ب – دید ایستا (Static view) : توصیف کننده ساختار سیستم و کلاسهای آن و روابط مابین کلاسها میباشد .

ج – دید پویا (Dynamic view) : توصیف کننده رفتارهای داخلی سیستم است مانند ارسال پیامها مابین اشیاء .

در جدول زیر منافع رهیافت شئ گرا ذکر شده است :

Concept	Supports	Leads to
Classes, objects, methods, and messages	■ A more realistic way for people to think about their business ■ Highly cohesive units that contain both data and processes	■ Better communication between user and analyst developer ■ Reusable objects ■ Benefits from having a highly cohesive system (see cohesion in Chapter 13)
Encapsulation and information hiding	■ Loosely coupled units	■ Reusable objects ■ Fewer ripple effects from changes within an object or in the system itself ■ Benefits from having a loosely coupled system design (see coupling in Chapter 13)
Inheritance	■ Allows us to use classes as standard templates from which other classes can be built	■ Less redundancy ■ Faster creation of new classes ■ Standards and consistency within and across development efforts ■ Ease in supporting exceptions
Polymorphism and Dynamic Binding	■ Minimal messaging that is interpreted by objects themselves	■ Simpler programming of events ■ Ease in replacing or changing objects in a syste ■ Fewer ripple effects from changes within an object or in the system itself
Use-case driven and use cases	■ Allows users and analysts to focus on how a user will interact with the system to perform a single activity	■ Better understanding and gathering of user neec ■ Better communication between user and analys
Architecture centric and functional, static, and dynamic views	■ Viewing the evolving system from multiple points of view	■ Better understanding and modeling of user neec ■ More complete depiction of information system
Iterative and incremental development	■ Continuous testing and refinement of the evolving system	■ Meeting real needs of users ■ Higher quality systems

تکالیف

تکلیف ۱ : درمورد Iterative and Incremental به عنوان خصوصیت یک متودولوژی تحقیق کنید .

تکلیف ۲ : درمورد Unified Process و یا Rational Unified Process (RUP) تحقیق کنید .

Chapter 3: Project Initiation

مقدمه :

مرحله اول فرآیند SDLC , شناخت پروژه به نحوی است که برای کاربران ارزشمند است . پس از آن باید گزارش درخواست سیستمها , که در آن اطلاعات اولیه درمورد سیستم اطلاعاتی ذکر شده است , را آماده نمود . سپس تحلیلگر اقدام به تحلیل امکان سنجی میکند تا امکان ساخت سیستم را از لحاظ اقتصادی , تکنیکی و سازمانی بررسی کند .

پروژه چیست ؟ به مجموعه ای از فعالیتها که یک نقطه شروع و یک نقطه پایان دارند و منجر به ایجاد سییستمی میشوند که ارزشهایی را برای سازمان به ارمغان می آورد , پروژه گویند .

یک پروژه زمانی شروع میشود که شخص یااشخاصی (به این اشخاص Project Sponsor میگویند) متوجه میشوند که توسط یک سیستم اطلاعاتی میتوانند ارزشهایی تجاری را عاید سازمان کنند . اهداف پروژه توسط گزارشی بنام درخواست سیستمها مشخص میشوند که این گزارش به گروهی بنام کمیته تایید ارسال میشود . این کمیته به بررسی گزارش پرداخته و در صورت تایید آنان , وارد مرحله تحلیل امکان سنجی میشویم .

تحلیل امکان سنجی نقش مهمی را در اجرا و یا عدم اجرای یک پروژه و پیشرفت آن دارد .

توسط تحلیل امکان سنجی مزایا و معایب پروژه را از جنبه های مختلف بررسی میکنیم و جزئیات کافی و دقیقی را در مورد منافع پروژه برای اسپانسور پروژه شرح میدهیم . در اکثر مواقع سفارش دهندگان سیستم همراه با تحلیلگر در تحلیل امکان سنجی همکاری میکنند . در نهایت نتیجه امکان سنجی به کمیته تایید ارسال میشود .

شناسایی ارزشهای و منافع تجاری(IDENTIFYING BUSINESS VALUE)

ایده ارزشهاو منافع تجاری که توسط یک سیستم اطلاعاتی عاید سازمان میشود , در ابتدا معمولا توسط اسپانسور پروژه مطرح میشود . این ارزشها و منافع میتوانند منافع قابل لمس (Tangible value) و یا غیر قابل لمس باشند (intangible value).

ارزشهای قابل لمس منافعی هستند که دارای مقدار و کمیت پذیر باشند (quantified) و معمولا بسادگی قابل سنجش هستند . مثلا کاهش سالانه ۲ درصدی هزینه های عملکردی سازمان

منافعی که غیر قابل مقدار دهی هستند و بسختی سنجیده میشوند , ارزشهای غیر قابل لمس نامیده میشوند . مثلا بهبود سرویس دهی به مشتری .

درخواست سیستمها System Request

درخواست سیستمها گزارشی است که در آن دلایل ساخت یک سیستم و منافعی که با ساخت آن سیستم عاید سازمان میشوند را مشخص میکند . معمولا اسپانسور پروژه خود این گزارش را تهیه میکند .

گزارش درخواست سیستمها از پنج عنصر اصلی تشکیل شده است :

۱- Project Sponsor

۲ – Business needs : دلایل تجاری که برای آغاز شدن ساخت سیستم لازمند . مانند " افزایش فروَش " ویا " بهبود دستیابی به اطلاعات " .

۳ – Business requirement : توانایی های تجاری که با ساخت سیستم به سازمان اضافه میشوند . مانند " دسترسی به اطلاعات از طریق وب " و یا " امکان جستجو در بین محصولات توسط کاربران "

۴ – Business value : منافعی که سیستم اطلاعاتی برای سازمان مهیا میکند . مانند " سه درصد افزایش فروش " و یا " ۱۵۰۰۰تومان کاهش هزینه در هر ماه "

۵ – نکات خاص و محدودیتها : نکات خاص و محدودیتهایی که برای ساخت سیستم با آن مواجه هستیم. مانند " سیستم باید برای تعطیلات نوروزی آماده باشد " ویا " نیاز به امنیت بالا "

System request Template

درخواست سیستمها – پروژه "نام پروژه"

مسئول پروژه :

دلایل و نیازهای تجاری :

دلایل تجاری که برای ساخت سیستم لازمند به شرح زیرند :

- "افزایش فروش محصولات"
- "بهبود دستیابی به اطلاعات"

نیازمندیهای تجاری :

توانایی های تجاری که توسط ساخت سیستم برای سازمان مهیا میشود به شرح زیر میباشد:

- "دسترسی به اطلاعات از طریق وب "
- "توانایی خرید محصولات از طریق شبکه اینترنت"

منافع تجاری :

از منافع قابل لمسی که با ساخت سیستم نصیب سازمان میشود میتوان موارد زیر را نام برد :

- "سه درصد افزایش فروش سالیانه"
- "یک میلیون تومان کاهش هزینه در هر سال"

نکات خاص و دیگر موارد :

- "سیسم باید برای تعطیلات پایان سال آماده استفاده باشد"
- "احتیاج به امنیت بالای سیستم "

در زیر مثالی از گزارش درخواست سیستمها آورده شده است :

System Request—Internet order project

Project sponsor: Margaret Mooney, Vice President of Marketing

Business Need: This project has been initiated to reach new Internet customers and to better serve existing customers using Internet sales support.

Business Requirements:

Using the Web, customers should be able to search for products and identify the brick-and-mortar stores that have them in stock. They should be able to put items on hold at a store location or place an order for items that are not carried or not in stock. The functionality that the system should have is listed below:

- Search through the CD Selections' inventory of products
- Identify the retail stores that have the product in stock
- Put a product on hold at a retail store and schedule a time to pick up the product
- Place an order for products not currently in stock or not carried by CD Selections
- Receive confirmation that an order can be placed and when it will be in stock

Business Value:

We expect that CD Selections will increase sales by reducing lost sales due to out-of-stock or nonstocked items and by reaching out to new customers through its Internet presence. We expect the improved services will reduce customer complaints, primarily because 50 percent of all customer complaints stem from out of stocks or nonstocked items. Also, CD Selections should benefit from improved customer satisfaction and increased brand recognition due to its Internet presence.

Conservative estimates of tangible value to the company includes:

- $750,000 in sales from new customers
- $1,875,000 in sales from existing customers
- $50,000 yearly reduction in customer service calls

Special Issues or Constraints:

- The Marketing Department views this as a strategic system. This Internet system will add value to our current business model, and it also will serve as a proof of concept for future Internet endeavors. For example, in the future, CD Selections may want to sell products directly over the Internet.
- The system should be in place for the holiday shopping season next year.

تحلیل امکان سنجی (Feasibility Analysis)

پس از آنکه دلایل نیاز به سیستم اطلاعاتی و نیازمندیهای تجاری مشخص شدند , زمان آن فرا رسیده است که به بررسی محدودیتهایی که در ساخت سیستم با آن مواجه هستیم بپردازیم و توسط تحلیل امکان سنجی امکان ادامه ساخت سیستم را بررسی کنیم و ر یسکهای پر اهمیت سیستم را مشخص کنیم. تحلیل امکان سنجی توسط سه دسته امکان سنجی انجام می پذیرد:

۱ – امکان سنجی تکنیکی (Technical feasibility)

۲ – امکان سنجی اقتصادی (Economic feasibility)

۳ – امکان سنجی سازمانی (Organizational feasibility)

نتایج تحلیل امکان سنجی در غالب گزارشی بنام مطالعه امکان سنجی (Feasibility study) به کمیته تایید ارائه میشود . البته مفاد این گزارش میتوانند در هر مرحله ای از SDLC تغییر کنند .

۱ – امکان سنجی تکنیکی (Technical feasibility)

Can We Build It?

در امکان سنجی تکنیکی پروژه , بررسی میکنیم که آیا میتوانیم سیستم را با موفقیت طراحی , پیاده سازی و نصب کنیم یا خیر . به عبارت دیگر ما بدنبال پاسخ به این پرسش که " آیا میتوانیم سیستم را بسازیم ؟ " هستیم . در حقیقت ما بدنبال شناسایی ریسکهای تکنیکی پروژه میباشیم . میتوان با بررسی پارامترهای زیر میزان ریسکهای تکنیکی سیستم را مشخص کرد :

الف : میزان نا آشنایی کاربران و تحلیلگران با محدوده کاربردی تجارت . در صورتیکه کاربران و یا تحلیلگران با حوزه کاربردی سیستم تجاری آشنا نباشند , میزان ریسک افزایس پیدا میکند . معولا میزان ریسک سیستمهایی که تا به حال ساخته نشده اند بالا میباشد .

ب : میزان آشنایی با تکنولوژی : اگر سازندگان سیستم بخواهند که از یک تکنولوژی جدید استفاده کنند ریسک سیستم بالا میرود .

ج : اندازه پروژه : اندازه پروژه توسط تعداد نفرات تیم پروژه , مدت زمان لازم برای ساخت سیستم و یا با تعداد ویژگیهای سیستم مشخص میشود . پروژه های بزرگ مسلما ریسک بالاتری دارند . زیرا مدیریت در آنها مشکل تر است و همچنین احتمال اینکه یک ویژگی مهم سیستم از قلم بیافتد بسیار زیاد است .

د : میزان سازگاری سیستم اطلاعاتی با تکنولوژی که در سازمان در حال استفاده است .

۲ – امکان سنجی اقتصادی (Economic feasibility)

Should We Build It?

امکان سنجی اقتصادی یا تحلیل هزینه – منافع (Cost-Benefit analysis) .

در اینجا ریسکهای اقتصادی سیستم را بررسی میکنیم . به عبارت دیگر بدنبال پاسخ به سئوال " آیا باید سیستم را بسازیم ؟ " هستیم و برای رسیدن به این پاسخ هزینه های سیستم و منافع اقتصادی سیستم را بررسی کرده و Cash Flow (گردش وجوه) را محاسبه میکنیم . یعنی مقدار هزینه سیستم و زمان برگشت سرمایه و مقدار سوددهی سیستم را حساب میکنیم .

نحوه محاسبه Cash Flow

مراحل محاسبه Cash Flow به صورت زیر است :

۱ – شناسایی هزینه ها و منافع : به عنوان اولین قدم باید هزینه ها و منافع سیستم را شناسایی کنیم و آنها را در یک جدول لیست کنیم . در حالت کلی هزینه ها ومنافع را میتوان به چهار دسته کلی تقسیم کرد :

الف : هزینه های ساخت سیستم : مانند حقوق اعضاء تیم , خرید سخت افزارها و نرم افزار ها , هزینه مشاوران , آموزش , هزینه های فضای کاری و تجهیزات .

ب : هزینه های عملکردی **:** هزینه هایی که برای عملکرد سیستم مصرف میشود . مثلا حقوق اپراتورها و کارمندان , هزینه خرید مجوز نرم افزارها , تجهیزات و هزینه های ارتباطی .

ج : منافع قابل لمس : شامل منافعی هستند که با کم شدن مخارج عاید سازمان میشوند . مانند افزایش فروش , کاهش کارمندان و ...

د : منافع و هزینه های غیر قابل لمس : منافع و هزینه هایی که براحتی قابل محاسبه نیستند . مانند بهبود سرویس دهی به مشتری , محصولات با کیفیت بهتر و ...

Development Costs	Operational Costs
Development Team Salaries	Software Upgrades
Consultant Fees	Software Licensing Fees
Development Training	Hardware Repairs
Hardware and Software	Hardware Upgrades
Vendor Installation	Operational Team Salaries
Office Space and Equipment	Communications Charges
Data Conversion Costs	User Training

Tangible Benefits	Intangible Benefits
Increased Sales	Increased Market Share
Reductions in Staff	Increased Brand Recognition
Reductions in Inventory	Higher Quality Products
Reductions in IT Costs	Improved Customer Service
Better Supplier Prices	Better Supplier Relations

۲ – مقداردهی و ارزش دهی به منافع و هزینه ها :

پس از آنکه نوع هزینه ها و منافع مشخص شدند , باید ارزش پولی آنها را مشخص کنیم . البته اینکار بسیار مشکل است چون پیش بینی هزینها و منافع عملی بسیار مشکل است . برای اینکار بهتر است از افرادی که دراین زمینه مهارت دارند استفاده کنیم .

Benefits[a]	
Increased sales	500,000
Improved customer service[b]	70,000
Reduced inventory costs	68,000
Total benefits	**638,000**
Development costs	
2 servers @ $125,000	250,000
Printer	100,000
Software licenses	34,825
Server software	10,945
Development labor	1,236,525
Total development costs	**1,632,295**
Operational costs	
Hardware	54,000
Software	20,000
Operational labor	111,788
Total operational costs	**185,788**
Total costs	**1,818,083**

[a] An important yet intangible benefit will be the ability to offer services that our competitors currently offer.

[b] Customer service numbers have been based on reduced costs for customer complaint phone calls.

۳ – تعیین Cash Flow :

گردش وجوه معمولا در یک بازه زمانی چند ساله (عموما بین ۳ تا ۵ سال) در نظر گرفته میشود و هزینه ها و منافع برای هرسال محاسبه میشود . سپس در پایان بازده زمانی مجموع هزینه ها و منافع در نظر گرفته میشود .

۵ – مشخص کردن NPV :

مشکل Cash Flow اینست که ارزش زمانی پول را در نظر نمیگیرد (ارزش پول امروز با ارزش پول سال دیگر متفاوت است) . به همین علت برای محاسبه ارزش زمانی پول ما پارامترهای دیگری را هم محاسبه میکنیم .

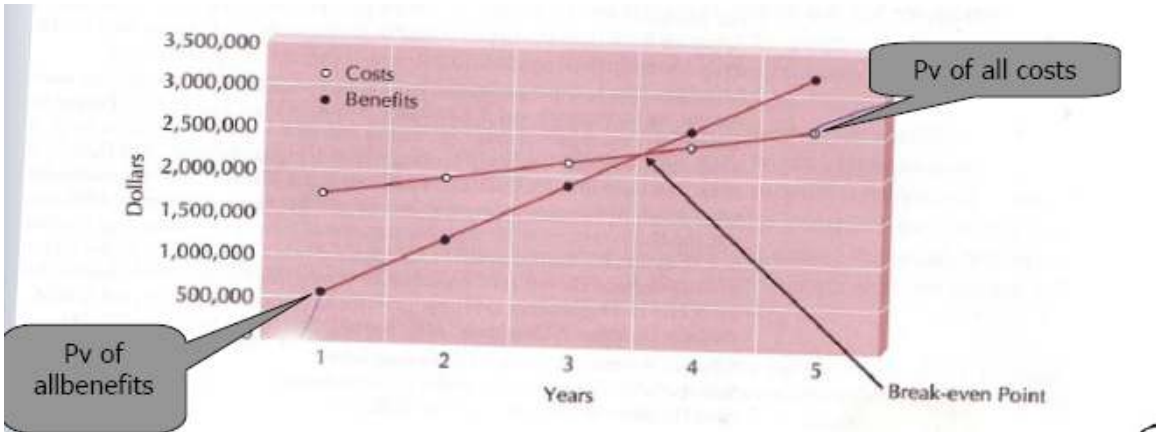
Presented Value (PV) :مقایسه ارزش امروزسرمایه با ارزش سرمایه در سالهای آتی. یا به عبارت دیگر مشخص میکنیم پول امروز در آینده چقدر ارزش دارد .

$$PV = \frac{\text{مقدار پول}}{(1 + \text{نرخ بهره})^{n}}$$

n= تعداد سالهای مورد نظر

Net Presented Value (NPV): کسر ارزش فعلی منافع از ارزش فعلی هزینه ها

Break-Even point : نقطه ای از زمان که هزینه های پروژه با منافعی که از پروژه عاید سازمان میشود , برابر شود .یعنی زمانیکه پس از کارکرد سیستم منافع حاصل شده از سیستم اطلاعاتی , کل هزینه های پروژه را خنثی کند .


3,500,000
3,000,000
2,500,000
2,000,000
1,500,000
1,000,000
500,000
Dollars
Costs
Benefits
1
2
3
4
5
Years
Pv of all costs
Pv of allbenefits
Break-even Point

در زیر مثالی از تحلیل هزینه–منافع آورده شده است :

	2003	2004	2005	2006	2007	Total
ncreased sales	500,000	530,000	561,800	595,508	631,238	
Reduction in customer complaint calls	70,000	70,000	70,000	70,000	70,000	
Reduced inventory costs	68,000	68,000	68,000	68,000	68,000	
Total Benefits:	638,000	668,000	699,800	733,508	769,238	
PV of Benefits:	**619,417**	**629,654**	**640,416**	**651,712**	**663,552**	**3,204,752**
PV of All Benefits:	**619,417**	**1,249,072**	**1,889,488**	**2,541,200**	**3,204,752**	
2 Servers @ $125,000	250,000	0	0	0	0	
Printer	100,000	0	0	0	0	
Software licenses	34,825	0	0	0	0	
Server software	10,945	0	0	0	0	
Development labor	1,236,525	0	0	0	0	
Total Development Costs:	1,632,295	0	0	0	0	
Hardware	54,000	81,261	81,261	81,261	81,261	
Software	20,000	20,000	20,000	20,000	20,000	
Operational labor	111,788	116,260	120,910	125,746	130,776	
Total Operational Costs:	185,788	217,521	222,171	227,007	232,037	
Total Costs:	1,818,083	217,521	222,171	227,007	232,037	
PV of Costs:	**1,765,129**	**205,034**	**203,318**	**201,693**	**200,157**	**2,575,331**
PV of All Costs:	**1,765,129**	**1,970,163**	**2,173,481**	**2,375,174**	**2,575,331**	
Total Project Benefits—Costs:	**(1,180,083)**	**450,479**	**477,629**	**506,501**	**537,201**	
Yearly NPV:	**(1,145,712)**	**424,620**	**437,098**	**450,019**	**463,395**	**629,421**
Cumulative NPV:	**(1,145,712)**	**(721,091)**	**(283,993)**	**166,026**	**629,421**	
Return on Investment:	**24.44%**	(629,421/2,575,331)				
Break-even Point:	**3.63 years**	(break-even occurs in year 4; [450,019 – 166,026] / 450,019 = 0.63)				
Intangible Benefits:	This service is currently provided by competitors Improved customer satisfaction					

۳ – امکان سنجی سازمانی (Organizational feasibility)

If we build it, will they come?

در این مرحله از امکان سنجی ما بررسی میکنیم که به چه نحوی سیستم مورد قبول کاربران قرار میگیرد و چگونه میتواند در عملکردهای سازمان دخیل شود . برای اینکار باید خیلی از فاکتورهای سازمانی را در نظر بگیریم . در کل باید به این سئوال پاسخ دهیم که " آیا اگر سیستم ساخته شود , مورد استفاده قرار خواهد گرفت ؟ " .

اولین راه برای امکان سنجی سازمانی پروژه , تطبیق اهداف پروژه با اهداف تجاری سازمان است . اصطلاحا به این کار , Strategic alignment گفته میشود .

راه دوم Stakeholder analysis است . تمامی اشخاص , گروه ها و یا سازمانهایی که سیستم اطلاعاتی به نحوی بر آنها تاثیر گذار یا تاثیر پذیر است Stakeholder ها نامیده میشوند .

Stakeholderها را میتوان به سه دسته تقسیم کرد :

الف – Champion : شخص یا اشخاصی که گزارش در خواست سیستمها را تهیه کرده اند و پشتیبانی و تهیه منابع پروژه را عهده دارند .

ب – مدیریت سازمان : این مدیران هستند که کاربران را وادار به استفاده از سیستم و قبول تغییرات میکنند .

ج – کاربران سیستم : اشخاصی که به استفاده از سیستم میپردازند .

در نهایت باید نتایج بدست آمده در گزارشی بنام مطالعه امکان سنجی جمع آ وری شوند .

Feasibility Study Template

گزارش مطالعه امکان سنجی پروژه (نام پروژه)

امکان سنجی تکنیکی :

میزان ریسک پروژه در مورد آشنایی با کاربرد پروژه (کم – متوسط – زیاد) است

ریسکهای شناخته شده در این مورد

میزان ریسک پروژه در مورد آشنایی با تکنولوژی (کم – متوسط- زیاد) است

ریسکهای شناخته شده در این مورد

میزان ریسک پروژه در ارتباط با اندازه پروژه (کم – متوسط – زیاد) است

ریسکهای شناخته شده در این مورد .

میزان سازگاری سیستم اطلاعاتی با سازمان (کم – متوسط – زیاد) است

ریسکهای شناخته شده در این مورد

امکان سنجی تکنیکی

گزارش تحلیل هزینه – منافع

امکان سنجی سازمانی

میزان ریسک پروژه از دید سازمانی (کم – متوسط – زیاد) است.

توضیحات مربوطه

نکات دیگر

گزارشی از مطالعه امکان سنجی در زیر آورده شده است .

Internet Order Feasibility Analysis Executive Summary

Margaret Mooney and Alec Adams created the following feasibility analysis for the CD Selections Internet Order System Project. The System Proposal is attached, along with the detailed feasibility study. The highlights of the feasibility analysis are:

Technical Feasibility

The Internet Order System is feasible technically, although there is some risk.

CD Selections' risk regarding familiarity with Internet order applications is high

- The Marketing Department has little experience with Internet-based marketing and sales.
- The IT Department has strong knowledge of the company's existing order systems; however, it has not worked with Web-enabled order systems.
- Hundreds of retailers that have Internet Order applications exist in the marketplace.

CD Selections' risk regarding familiarity with the technology is medium

- The IT Department has relied on external consultants and an Information Service Provider to develop its existing Web environment.
- The IT Department has gradually learned about Web systems by maintaining the current Web site.
- Development tools and products for commercial Web application development are available in the marketplace, although the IT department has little experience with them.
- Consultants are readily available to provide help in this area.

The project size is considered medium risk

- The project team likely will include less than ten people.
- Business user involvement will be required.
- The project timeframe cannot exceed a year because of the Christmas holiday season implementation deadline, and it should be much shorter.

The compatibility with CD Selections' existing technical infrastructure should be good

- The current Order System is a client-server system built using open standards. An interface with the Web should be possible.
- Retail stores already place and maintain orders electronically.
- An Internet infrastructure already is in place at retail stores and at the corporate headquarters.
- The ISP should be able to scale their services to include a new Order System.

Economic Feasibility

A cost–benefit analysis was performed; see attached spreadsheet for details. A conservative approach shows that the Internet Order System has a good chance of adding to the bottom line of the company significantly.

ROI over 3 years: 229 percent
Total benefit after three years: $3.5 million (adjusted for present value)
Break-even occurs: after 1.7 years

Intangible Costs and Benefits

- Improved customer satisfaction
- Greater brand recognition

Organizational Feasibility

From an organizational perspective, this project has low risk. The objective of the system, which is to increase sales, is aligned well with the senior management's goal of increasing sales for the company. The move to the Internet also aligns with Marketing's goal to become more savvy in Internet marketing and sales.

The project has a project champion, Margaret Mooney, Vice President of Marketing. Margaret is well positioned to sponsor this project and to educate the rest of the senior management team when necessary. To date, much of senior management is aware of and supports the initiative.

The users of the system, Internet consumers, are expected to appreciate the benefits of CD Selections' Web presence. And, management in the retail stores should be willing to accept the system, given the possibility of increased sales at the store level.

Additional Comments:

- The Marketing Department views this as a strategic system. This Internet system will add value to our current business model, and it also will serve as a proof of concept for future Internet endeavors.
- We should consider hiring a consultant with expertise in similar applications to assist with the project.
- We will need to hire new staff to operate the new system, from both the technical and business operations aspects.

مطالعه بیشتر

- For an excellent source of information on IT management see:
- www.cio.com

Chapter 5: Requirements Determination

جمع آوری نیازمندیها (Requirements Gathering)

نیازمندیها (Requirement)

نیازمندیها عبارت است از کارهایی که قرار است سیستم انجام دهد و یا ویژگیهایی که سیستم باید داشته باشد . در این مرحله مشخص میشود که سیستم چه کاری باید انجام دهد .

نیازمندیها را از دید کاربران سیستم (Business user) تعریف میکنیم که به ان Business requirement گفته میشود . ولی در مرحله طراحی نیازمندیها از دید توسعه دهندگان سیستم بیان میشود که به آن System requirement گفته میشود .

در نهایت تمامی نیازمندیها جمع آوری شده و در غالب ***تعریف نیازمندیها*** ارائه میشود .

انواع نیازمندیها

۱ – **نیازمندیهای اساسی** (Functional requirement) **:** شامل فرایندی یا اطلاعاتی است که سیستم باید داشته باشد و در حقیقت مشخص کننده عملکرد سیستم است

۲ – **نیازمندیهای غیر اساسی** (nonfunctional requirement) **:** شامل خصوصیات رفتاری است که سیستم باید داشته باشد . مانند کارایی بالا و امکان دسترسی از طریق اینترنت . در حقیقت ویژگیهای مختلف سیستم مانند ویژگیهای عملیاتی , کیفی , امنیتی , فرهنگی و سیاسی را از طریق این نیازمندیها مشخص میکنیم .

۱ – نیازمندیهای غیر اساسی

- Operational requirement : محیطهای فیزیکی و تکنیکی که سیستم در آن عمل خواهد کرد .
- Performance requirement : سرعت , حجم و قابلیت اطمینان سیستم
- Security requirement : چه افرادی و تحت چه شرایطی توانایی دسترسی به سیستم و اطلاعات را دارا میباشند .
- Cultural and Political requirement : فاکتورهای فرهنگی و سیاسی و قوانینی که بر سیستم تاثیر گذارند .

در نهایت کلیه نیازمندیهای شناخته شده را در غالب گزارش تعریف نیازمندیها ارائه می کنیم.

Requirement Definition Template

نیازمندیهای غیر اساسی شناخته شده در سیستم به شرح زیرند :

۱ – نیازمندیهای عملکردی

الف : "سیستم باید تحت محیط ویندوز عمل کند"

ب :" سیستم باید توانایی خواندن و نوشتن فایلهای Word و Html را داشته باشد "

ج : "سیستم باید توانایی مشاهده تصاویر با فرمت jpeg , gif و BMP را داشته باشد "

۲ – نیازمندیهای کیفی

الف :" زمان پاسخ باید حداکثر ۷ ثانیه باشد "

۳ – نیازمندیهای امنیتی

" مورد امنیتی خاصی در سیستم موجود نیست "

۴ – نیازمندیهای فرهنگی و سیاسی

...

نیازمندیهای اساسی شناخته شده در سیستم به شرح زیرند :

...

مراحل فرآیند تحلیل

فرآیند تحلیل شامل سه مرحله میشود :

- فهم سیستم موجود (as-is system)
- شناسایی نقاط بهبود و پیشرفت سیستم (system improvement)
- تشخیص و توسعه نیازمندیها برای سیستم جدید (to-be system)

در فرآیند تحلیل ما احتیاج به تکنیکهایی برای جمع اوری نیازمندیها (requirement gathering) به منظور جمع آوری اطلاعات , و تکنیکهایی برای تحلیل نیازمندیها (requirement analyzing) برای مشخص کردن اینکه چه نوع اطلاعاتی جمع آوری شده است و اینکه چگونه آنرا تحلیل کنیم , داریم .

تکنیکهای تحلیل نیازمندیها

- **اتوماتیک کردن فرآیند تجارت – BPA**

(Business Process Automation)

- **بهبود فرآیند تجارت -- BPI**

(Business Process Improvement)

- **مهندسی مجدد فرآیند تجارت – BPR**

(Business Process Reengineering)

1 – BPA

در این تکنیک ما اقدام به بررسی سیستم نموده و مشخص میکنیم که در کدام قسمتهای سازمان میتوان کارها را کامپیوتری انجام داد . در این مرحله در عملیاتها و فرآیندهای سازمان هیچ تغییری به وجود نمی آوریم . تکنیکهای BPA زمان زیادی را صرف فهم سیستم موجود می نمایند تا مشخص کنند در سیستم آینده چه فرایندهای تجاری در سیستم قابلیت اتومانیک سازی را دارند و مشخص شود کدام قسمتهای سازمان باید توسط سیستمهای اطلاعاتی پیاده سازی شوند .

تکنیکهای BPA

- **تحلیل مشکل (Problem analysis):** در این تکنیک با استفاده از کاربران و مدیران سیستم به تشخیص سیستم جاری (as-is system) می پردازیم و مشخص میکنیم که مشکل را به چه صورت در سیستم آتی (to-be system) حل کنیم . بدین معنی که از کاربران و مدیران سازمان میخواهیم که مشکلات سازمان را برای ما تشریح کرده و به ما بگویند به چه طریقی میتوان این مشکلات را حل نمود . در اینجا تمرکز برروی حل مشکلات است نه خود مشکل .در پایان این قسمت راه حل مشکلات سازمان پیدا میشود .
- **تحلیل علت ریشه (Root cause analysis) :** در اینجا تمرکز برروی خود مشکل است و تحلیلگر سعی در پیدا کردن ریشه مشکل دارد .به عنوان مثال فرض کنید یکی از لامپهای منزل شما مرتبا میسوزد . شما میتوانید هر دفعه که این لامپ سوخت آنرا با یک نمونه سالم تعویض کنید . (مشکل به صورت موقت حل شده است . یعنی

از تکنیک تحلیل مشکل استفاده کرده اید .) و یا میتوانید بدنبال علت سوختن مکرر لامپ بروید .(از تکنیک تحلیل علت ریشه استفاده کنید و با پیدا کردن ریشه مشکلات , مشکل سوختن لامپ را برای همیشه حل کنید .)

برای اینکار تحلیلگر ابتدا لیستی از مشکلات سیستم جاری تهیه کرده و سپس آنها را اولویت بندی می کند . سپس به برسی مهم ترین انها پرداخته و سعی در پیدا کردن ریشه مشکل دارد . معمولا با پیدا کردن ریشه یکی از مشکلات , تعدادی دیگر از مشکلات نیز حل میشوند .

2- BPI

در این تکنیک ما اقدام به ایجاد تغییراتی در سازمان نموده تا بتوانیم از منافع تکنولوژیهای جدید استفاده نموده و عملکرد سازمان را بهبود بخشیم . توسط BPI عملکرد سازمان بهتر میشود و میتوانیم بازدهی را توسط انجام دادن درست کارها(doing things right) واثربخشی را توسط انجام کارهای درست(doing the right things), بهبود بخشیم .

تکنیکهای BPI

۱ – تحلیل مدت (duration analysis) :

در این متد ما مدت زمان انجام هر فرآیند را در سیستم جاری بررسی میکنیم . تحلیلگر کار خود را با مشخص کردن کل و میانگین مدت زمانیکه برای انجام یک مجموعه از فرایندهای تجاری که به ازاء یک ورودی خاص اجرا می شوند , شروع میکند . سپس مدت زمان اجرای هر کدام از عناصر این مجموعه از فرآیندها را مشخص میکند . در صورتیکه مدت زمان اجرای هر کدام از فرآیندها تفاوت فاحشی با کل زمان لازم برای اجرای مجموعه فرآیندها داشته باشد , آنگاه این فرایند احتیاج به تغییر دارد . به عنوان مثال در فرآیند پرداخت وام , ممکن است کل فرآیند دو ماه زمان ببرد . پس از تحلیل متوجه میشویم که کارهایی که در این فرایند انجام میشود (مانند بررسی درخواست وام گیرنده , میزان اعتبار وام گیرنده , بررسی موجودی بانک و ...) هرکدام چند ساعت بیشتر طول نمیکشد و مجموع این ساعات مشخص میکند که عملیات پرداخت وام یک هفته بیشتر طول نمیکشد . این اختلاف فاحش مشخص میکند که فرآیند پرداخت وام باید تغییر کند و خیلی از قسمتهای آن میتوانند به صورت موازی انجام شوند . مثلا میتوان بررسی میزان اعتبار وام گیرنده و بررسی موجودی بانک ار همزمان انجام دهیم .

۲ – Activity-base costing : همانند تحلیل مدت فقط به جای محاسبه زمان , به بررسی و محاسبه هزینه های فرآیندها می پردازیم .

۳ – محک زدن اطلاعات

(Information benchmarking) چگونگی انجام کارها در فرآیندهای سازمان مورد تحلیل را با دیگر سازمانهای رغیب بررسی می کنیم

3- BPR

در این تکنیک عملیات سازمان در سیستم حذف شده و روند انجام کارها براساس تکنولوژیهای جدید تغییرات اساسی پیدا خواهند کرد . در این تکنیک زمان کمی صرف سیستم موجود میشود .

باBPR چگونگی انجام کارها کاملا تغییر می یابد و راه فعلی انجام فرآیندها حذف شده و فرآیندهای جدیدی تعریف میشوند .

تکنیکهای BPR

۱ – **تحلیل نتیجه (outcome analysis) :** در این مرحله تمرکز برروی فهم نتایج و خروجیهایی که برای مشتریان ارزشمند هستند میباشد . مثلا برای کسی که به یک شرکت بیمه برای دریافت خسارت ماشین تصادف کرده اش مراجعه میکند , مهمترین نتیجه و خروجی دریافت هرچه سریعتر خسارتش میباشد . تحلیلگران معمولا از مدیران و کاربران میخواهند که خود را جای مشتریان گذاشته و مشخص کنند که مشتری چه سرویسهایی را از سیستم خواستار است .

۲ – **تحلیل تکنولوژی (Technology analysis) :** تحلیلگران و مدیران لیستی از تکنولوژیهای مهم و قابل توجه تهیه کرده و سپس بررسی میکنند که کدام تکنولوژی برای سیستم مورد نطرشان مناسب است

۳ – **حذف فعالیت (Activity elimination):** در این تکنیک نحلیلگر و مدیران همراه هم شروع بکارکرده تا مشخص کنند چه فعالیتهایی را میتوانند در سازمان حذف کنند , به چه صورت با حذف این فعالیت سازمان میتواند به کار خود ادامه دهد و با حذف این فعالیت چه اعمالی دیگر در سازمان انجام نمیشوند و اثرات ناشی از آن به چه میزان میباشند . متلا میتوان فغالیتهای تکراری سازمان را حذف کرد . معمولا مدیران به این مهم تمایلی نشان نمیدهند ولی ما باید آنها را مجبور به همکاری کنیم .

تکنیکهای جمع آوری نیازمندیها

- مصاحبه (Interview)
- JAD
- پرسشنامه (Questionnaires)
- تحلیل اسناد (Document analysis)
- مشاهده (Observation)

۱- مصاحبه

مراحل مصاحبه به شرح زیر میباشد :

الف - انتخاب افراد برای مصاحبه

ب – طراحی سئوالات

ج – آماده شدن برای مصاحبه

د – انجام مصاحبه و هدایت آن

ه – کارهای بعد از اتمام مصاحبه

الف : انتخاب افراد برای مصاحبه

در ابتدای کار باید لیستی از افرادی را که می خواهیم باانها مصاحبه کنیم تهیه کرده , هدف مصاحبه با هر شخص را مشخص کنیم و زمان انرا نیز تعیین کنیم . یعنی مشخص کنیم با چه کسی , در چه زمانی و با چه هدفی باید مصاحبه کنیم . افرادی که باید برای مصاحبه انتخاب شوند بر اساس اینکه تحلیلگر به چه اطلاعاتی نیاز دارد , مشخص میشوند .

در ابتدا باید با مدیران ارشد , سپس با مدیران میانی و در نهایت با کارمندان مصاحبه شود , که این ترتیب میتواند مرتبا تکرار شود . زیرا در ابتدا باید یک درک کلی از فرآیندهای سازمان پیدا شود و بعد با مصاحبه با کارمندانی که این فرآیندها را انجام میدهند از چگونگی مراحل داخل فرآیندها مطلع شد .

ب- طراحی سئوالات

انواع سئوالات :

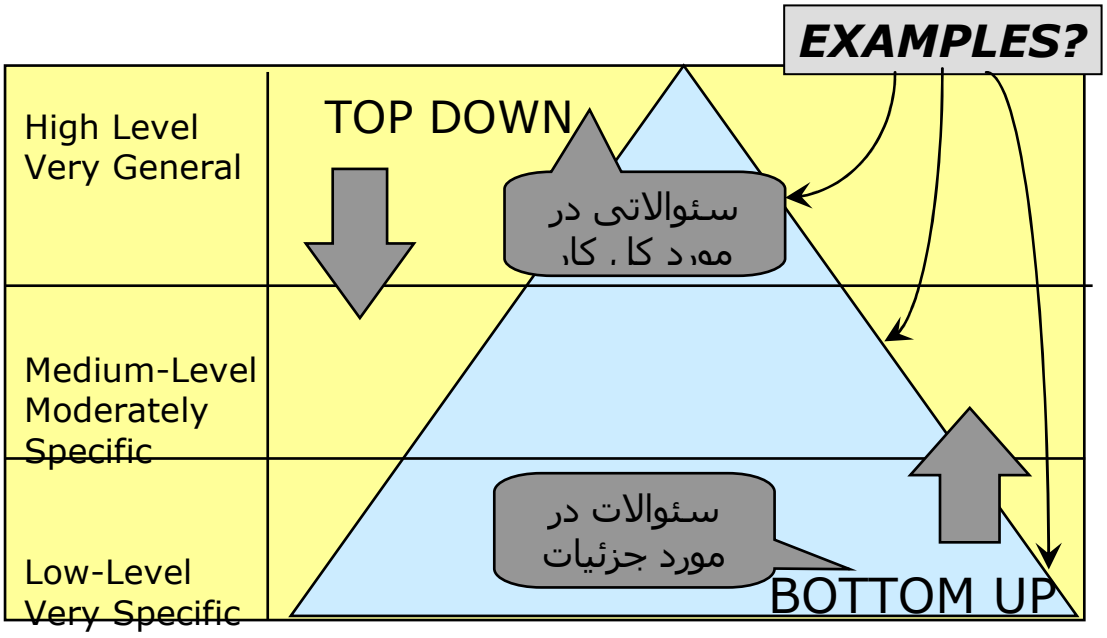
- Closed – End : پرسشهایی هستند که در آنها یک جواب خاص وجود دارد و اطلاعات دقیقی را برای ما مشخص میکند . مثلا بجای اینکه بپرسیم " آیا شما در خواستهای زیادی را در روز پاسخ میدهید ؟ " بپرسیم " چه تعداد درخواستهایی را شما روزانه پاسخ میدهید ؟ "
- Open – End : سئوالاتی هستند که در آنها جواب خاص و دقیقی وجود ندارد مثلا اظهار نظر در مورد یک موضوع و یا اینکه " شما با چه مشکلاتی در سازمان روبرو هستید ؟"
- Probing(کاوشی) : سئوالاتی که در فهم مسئله به ما کمک می کنند و جزئیات بیشتری را برای ما آشکار می کنند و معمولا زمانی پرسیده میشود که قسمتهایی از صحبتهای مصاحبه شونده برای مصاحبه کننده غیر شفاف است . مثلا " آیا ممکن است برای من مثالی بزنید ؟" .

مثال

Types of Questions	Examples
Closed-Ended Questions	* How many telephone orders are received per day? * How do customers place orders? * What additional information would you like the new system to provide?
Open-Ended Questions	* What do you think about the current system? * What are some of the problems you face on a daily basis? * How do you decide what types of marketing campaign to run?
Probing Questions	* Why? * Can you give me an example? * Can you explain that in a bit more detail?

در ابتدای پروژه به علت اینکه مصاحبه کننده هیچ اطلاعی در مورد سیستم ندارد , نمی تواند سئوالات دقیقی را طراحی کند . یعنی مصاحبه کننده از یک مدل بدون ساختار استفاده میکند و سئوالات کلی در مورد سیستم می پرسد . پس از گذشت مدتی با آشنا شدن به سیستم , مصاحبه به شکل ساختار داده –شده ای ادامه پیدا میکند و میتوان برروی موارد خاص تمرکز کرد .

استراتژی های طراحی سئوالات



ج – آماده شدن برای مصاحبه

خیلی از تحلیلگران تازه کار به این علت که تهیه کردن یک ساختار برای مصاحبه و استفاده از سئوالات Close-End عملی دشوار و وقت گیر است , از اینکار اجتناب میکنند . غافل از اینکه در صورتیکه اگر ساختار مصاحبه آماده نباشد کار بسیار دشوارتر شده و مجبور به مصاحبه دوباره میشویم و همچنین اکثر افراد از اینکه زیاد مورد مصاحبه قرار بگیرند ناراضی میباشند . برای آماده شدن برای انجام یک مصاحبه باید اعمال زیر را انجام داد :

۱ – آماده سازی طرح کلی مصاحبه

- تهیه لیست سئوالات
- پیش بینی پاسخها

۲– اولویت بندی سئوالات

۳ – آماده شدن برای مصاحبه

- زمانبندی
- توضیح دلیل مصاحبه
- توضیح محدوده مورد بحث

د – انجام و هدایت مصاحبه

نکاتی که برای هدایت و انجام مصاحبه باید مد نظر داشت :

- در طول مصاحبه بیطرفی و بیغرضی خودراکاملا نشان دهید
- تمامی اطلاعات را ثبت کنید
- از اینکه تمامی نکات و واژگان را درست متوجه شده اید اطمینان حاصل کنید
- حقایق را از نظرات جدا کنید
- زمانی را برای مصاحبه شونده برای طرح سئوالات اختصاص دهید
- ادب را در طول کل مصاحبه کاملا رعایت کنید
- شوخ طبع باشید

- از گفتن اصطلاحات فنی خودداری کنید
- در زمان معین شده به مصاحبه پایان دهید
- نکات کلیدی را جمع بندی کنید
- کوتاه و مختصر سخن بگویید
- راستگو باشید .
- به زبان اشاره و حالات مصاحبه شونده دقت کنید

ه – کارهای پس از مصاحبه

- پس از پایان مصاحبه اعمال زیر را انجام دهید:
- آماده سازی نکات مطرح شده در مصاحبه
- آماده سازی گزارش مصاحبه حداکثر تا ۴۸ ساعت پس از هرمصاحبه
- بررسی مصاحبه انجام شده برای مشخص کردن نکات نامعلوم و پرسشهای جدید
- ارسال گزارش ارسال شده به مصاحبه شونده برای گرفتن تاییدیه .

گزارش مصاحبه

گزارش مصاحبه

مصاحبه توسط ---------- تایید شده است .

مصاحبه شونده : ----------

تاریخ : ---------- **مصاحبه کننده :** ----------

اهداف مصاحبه :

هدف اصلی : ...

بقیه اهداف : ...

خلاصه مصاحبه :

...

مواردی که هنوز مشخص نشده اند یا در مورد آنها ابهاماتی وجود دارد :

...

جزئیات :

2- JOINT APPLICATION DESIGN (JAD)

مقدمه

جلساتی است که مدیر پروژه به همراه توسعه دهندگان و کاربران سیستم تشکیل می دهند و به بحث و گفتگو در مورد فهم نیازمندیهای سیستم می پردازند . در این جلسات , چون تمامی نقشهایی که در ساخت سیستم اثر گذارند , وجود دارند فهم صحیح نیازمندیها میسر میشود .

افراد مهم در JAD

۱. درون جلسات است .این شخص باید از اظهار نظر در مورد سیستم خودداری کند و کاملا در بحث بی اثر باشد . وظایف این شخص به شرح زیر است :

- هدایت جلسات در مسیر تعیین شده
- کمک به گروه برای فهم اصطلاحات تکنیکی
- ثبت ورودیهای گروه
- کمک به حل شبهات موجود

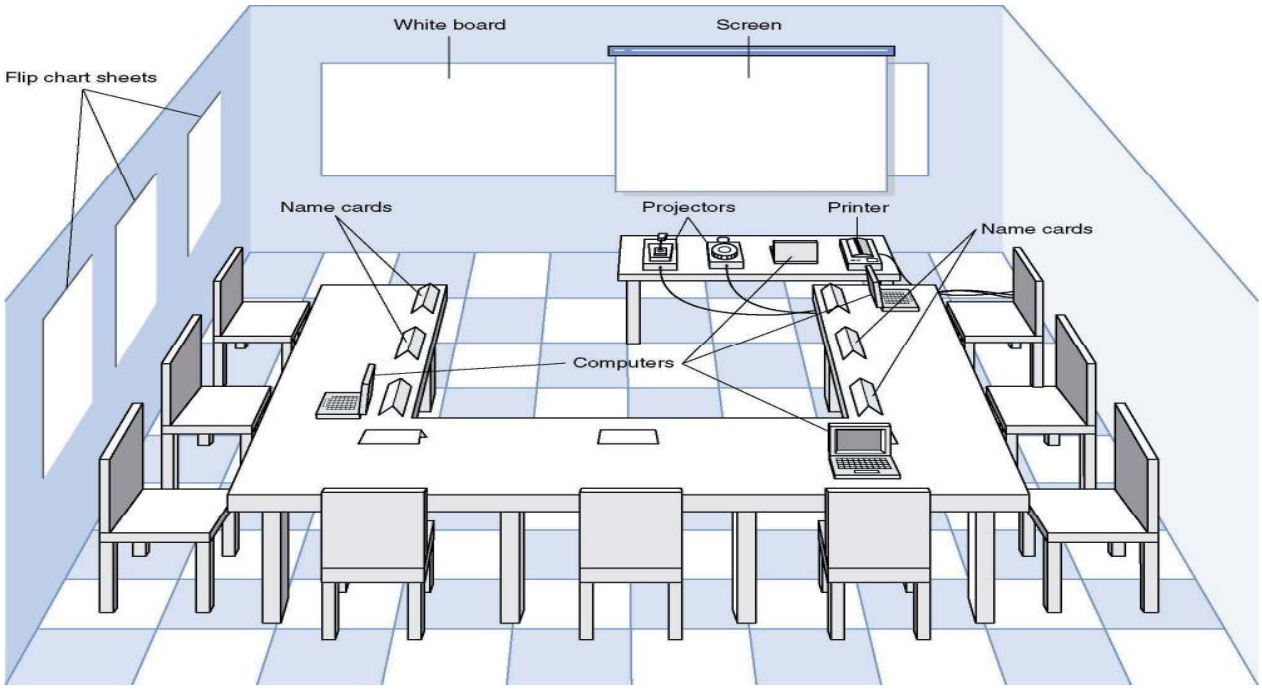
۲. Scribe : یک یا دو منشی هستند که با ثبت نکات , گرفتن کپی و کارهایی از این قبیل به Facilitator کمک میکنند .

Joint Application Design (JAD) Setting

- از یک میز U شکل استفاده می کنیم تا تمامی افراد براحتی همدیگر را ببینند .
- تمام شرکت کنندگان باید حواس خود را کاملا معطوف جلسه کنند و Facilitator باید متوجه این امر باشد .
- از ابزارهای مختلفی مانند Whiteboard , دستگاههای Projection , دستگاههای شبیه ساز و اسلاید برای انجام جلسات استفاده میشود .

- E-JAD : وقتی که از نرم افزارهای خاص و شبکه های کامپیوتری برای ارسال نظرات افراد به همدیگر استفاده کنیم که باعث میشود هر شخص نظر خود را مستقیما به شخص دیگر ارسال کند , بدون اینکه در بحث جاری خللی به وجود آید .

JAD Meeting Room



۳ - پرسشنامه

بیشتر در زمانهایی استفاده میشود که تعداد افراد مورد نظر برای مصاحبه زیاد است . و یا زمانی که میخواهیم یک سیستم را برای خارج از یک سازمان مثلا برای یکسری مشتری طراحی کنیم .

طراحی یک پرسشنامه مناسب

- شروع پرسشنامه باید جلب توجه کننده باشد و از عبارات تهدید آمیز در آن استفاده نشود .
- آیتمهایی از پرسشنامه که در یک ارتباط منطقی با هم میباشند باید گروه بندی شوند .

- نکات مهم نباید در پایان پرسشنامه قرار گیرند .
- یک صفحه را با نکات مختلف شلوغ و درهم نسازید .
- از کوته سازی اجتناب کنید .
- از واژگان پیشنهادی یا غرض دار اجتناب کنید .
- سئوالات را شماره بندی کنید .
- سئوالات طرح شده را امتحان کنید .
- تا حد ممکن سعی شود حویت پاسخ دهنده گمنام بماند .

4- Document Analysis

این تکنیک ابزار مناسبی برای شناخت سیستم جاری میباشد .

اسناد مختلفی در سیستم میتوانند بررسی شوند مانند :

- فرمهای سازمان
- گزارشات سازمان
- سیاستهای سازمانی

در این تکنیک باید اسناد را به دقت بررسی کرد . به عنوان مثال باید به قسمتهایی که کاربران و مشتریان سیستم آنرا کامل نکرده اند و یا اشتباها تکمیل نموده اند توجه کرد .

5 - Observation

خیلی از کاربران و مدیران دقیقا کارهایی را که انجام می دهند به خاطر ندارند . بنابراین مشاهده کارکنان در حین انجام کارشان ابزار مناسبی برای فهم چگونگی روند کار میباشد .

همچنین توسط این تکنیک میتوان صحت اطلاعات بدست آمده در مراحل قبل را بررسی کرد .

مشکل این روش اینست که خیلی از افراد در صورتیکه متوجه شوند شخصی در حال مشاهده و ثبت اعمال انها میباشد , روند کار خودرا تغییر می دهند که این مسئله ممکن است در فهم درست نیازمندیها خلل به وجود آورد .

مطالعه بیشتر

Additional resources regarding Joint Application Development can be found at:

http://www.carolla.com/wp-jad.htm

http://www.utexas.edu/hr/is/pubs/jad.html

Chapter 6: Functional Modeling

مدلسازی تابعی(Functional Modeling)

در فصل قبلی نیازمندیهای سیستم تعیین شدند که توسط آنها کارهایی را که سیستم انجام میدهد را مشخص کردیم . در این فصل نشان می دهیم که اطلاعات جمع آوری شده (نیازمندیها) به چه صورت سازماندهی شده و در غالب نمودارهای فعالیت و مورد کاری (Activity and UseCase diagrams)مدلسازی شوند .

Functional Modeling توصیف کننده فرآیند تجارت و تراکنشهای مابین سیستم اطلاعاتی و محیط اطراف میباشد . در Functional Modeling دو مدل اساسی برای توصیف عملکرد سیستم بکار میروند

- دیاگرامهای فعالیت (Activity diagram)
- دیاگرام مورد کاری (Usecase diagram)

دیاگرامهای فعالیت برای مدلسازی منطق فرآیند تجارت و جریانهای کاری استفاده میشوند و usecaseها هم برای توصیف عملکرد های اصلی سیستم اطلاعاتی استفاده میشوند و سیستم را از دید کاربر نشان میدهند .

مدلسازی یک فرآیند (Process Modeling) نشان دهنده چگونگی عملکرد سیستمهای تجاری میباشد . یعنی مشخص میکند که فرآیندها و فعالیتها چگونه انجام میشوند و اشیاء و داده ها چگونه مابین انها منتقل میشوند .

یک usecase یک ابزار رسمی برای نشان دادن اینکه چگونه یک سیستم تجاری با محیط اطراف خود در تعامل است , میباشد و نشان دهنده فعالیتهایی هستند که توسط کاربران سیستم انجام میشوند . Usecase ها فرآیندهای سیستم را از دید کاربران نشان میدهند ولی در مورد مکانیزمهای داخلی فرآیند (ساختار داخل فرآیندها و نحوه انجام فرآیندها) اطلاعاتی را در اختیار ما نمی گذارند .

به طور کلی تمامی متدهای مدلسازی به دو دسته کلی تقسیم میشوند:

- Logical Modeling : حوزه فعالیتهای تجاری را بدون در نظر گرفتن چگونگی رفتار آنها مشخص میکند . بدین معنی که مشخص میکند چه کارهای در سیستم انجام میشود نه اینکه چگونه اینکارهادرسیستم انجام میشوند .
- Physical Modeling : چگونگی انجام کارها را در سیستم بحث میکند و شامل اطلاعاتی است که برای پیاده سازی کارهای مشخص شده لازمند .

مراحل مدلسازی تابعی

پس از جمع اوری نیازمندیها و مشخص شدن انها , تحلیلگر شروع به ایجاد مدل منطقی میکند . بدین صورت که بر اساس اطلاعات جمع آوری شده در مرحله نیازمندیها , کل فرآیند تجارت توسط دیاگرامهای فعالیت مدلسازی میشود . سپس بر اساس فعالیتهای شناخته شده , usecaseها شناسایی شده , توصیفگر usecase و دیاگرام usecase رسم میشوند . Usecase ها فعالیتهای جداازهمی هستند که کاربران انجام میدهند مانند خرید CD" و یا "سفارش محصول" . بعد از آن تحلیلگر شروع به ساخت مدل ساختاری سیستم (Structural modeling) که در فصل بعد معرفی شده است , میکند . بعد از آنکه نمودارهای فعالیت و توصیفگر موارد کاری تعیین شدن , تحلیلگر آنها را تبدیل به نمودار مورد کاری میکند که در آن رفتارهای خارجی فرآیندهای تجاری به تصویر کشیده میشوند .

در ادامه به توصیف دیاگرامهای فعالیت و دیاگرامهای usecase میپردازیم .

Business Process Modeling with Activity Diagrams

مدلسازی فرآیند تجارت (BPM)

در BPM فعالیتهایی را توصیف میکنیم که مجموع این فعالیتها , فرآیندهای تجاری خاصی را در سیستم اجرا میکند . BPM ابزار بسیار قدرتمندی است برای اینکه تحلیلگرمطمئن شود که نیازمندیها را به درستی درک کرده است . BPM توصیف فرآیندها را توسط نمودار فعالیت انجام میدهد .

از نمودار فعالیت برای مدلسازی رفتار فرایندها مستقل از اشیاء استفاده میشود . در حقیقت نمودارفعالیت , فعالیتها و روابط مابین آنها را در داخل یک فرآیند به تصویر میکشد .

عناصر تشکیل دهنده دیاگرام فعالیت

- Actions and Activities : کارهایی هستند که به دلیل خاصی انجام میشوند . نام آنها با یک فعل شروع و با یک اسم پایان می یابد . مانند " گرفتن قرار ملاقات " . تفاوت مابین Action ها و activity ها در اینست که Activity ها خود از Action ها یا activity ها ی دیگر تشکیل شده اند , در حالیکه Action ها نشان دهنده یک عمل ساده و غیر قابل تجزیه میباشد .
- Object Node : Actions and Activities ها معمولا برروی اشیاء تغییراتی ایجاد میکنند که این اشیاء توسط Object Node ها نمایش داده میشوند .
- Control flows and Object flows : به طور کلی دونوع جریان در نمودار فعالیت وجود دارد : جریانهای Object و Control . جریانهای کنترلی برای مدلسازی مسیر اجرای فرآیندها استفاده میشود . جریانهای کنترلی فقط به Action ها یا Activity ها متصل میشوند . جریانهای اشیاء برای نشان دادن مسیر تغییراتی که بر روی اشیاء در طول اجرای فرآیندها رخ میدهد کاربرد دارد .
- Control Nodes : در نمودار فعالیت , نودهای کنترلی مختلفی وجود دارد مانند : Initial , Final-activity, Final-Flow, Decision, Merge, Fork و Join . برای توضیحات بیشتر درباره این نودها به اسلاید بعد مراجعه کنید .
- Swimlanes : برای دسته بندی فعالیتهایی که یک کار خاص را انجام میدهند .

An Action: ■ Is a simple, non-decomposable piece of behavior ■ Is labeled by its name	Action
An Activity: ■ Is used to represent a set of actions ■ Is labeled by its name	Activity
An Object Node: ■ Is used to represent an object that is connected to a set of Object Flows ■ Is labeled by its class name	Class Name
A Control Flow: ■ Shows the sequence of execution	
An Object Flow: ■ Shows the flow of an object from one activity (or action) to another activity (or action)	
An Initial Node: ■ Portrays the beginning of a set of actions or activities	
A Final-Activity Node: ■ Is used to stop all control flows and object flows in an activity (or action)	
A Final-Flow Node: ■ Is used to stop a specific control flow or object flow	
A Decision Node: ■ Is used to represent a test condition to ensure that the control flow or object flow only goes down one path ■ Is labeled with the decision criteria to continue down the specific path	[Decision Criteria] [Decision Criteria]
A Merge Node: ■ Is used to bring back together different decision paths that were created using a decision-node	
A Fork Node: ■ Is used to split behavior into a set of parallel or concurrent flows of activities (or actions)	
A Join Node: ■ Is used to bring back together a set of parallel or concurrent flows of activities (or actions)	
A Swimlane: ■ Is used to break up an activity diagram into rows and columns to assign the individual activities (or actions) to the individuals or objects that are responsible for executing the activity (or action) ■ Is labeled with the name of the individual or object responsible	Name

FIGURE 6-1 Syntax for an Activity Diagram

نکاتی برای ایجاد دیاگرام فعالیت

- از دیاگرام فعالیت میتوان برای مدلسازی هر نوع فعالیتی استفاده کرد . فقط باید محدوده و محتوای فعالیت مورد نظر را مشخص کنیم و با نامی مناسب نمودار را نامگذاری کنیم .
- باید فعالیتها , جریانهای کنترلی و جریانهای اشیائی را که مابین فعالیتها رخ میدهند شناسایی کنیم .
- باید تمامی تصمیم گیریهایی را که در حین اجرای فرآیند با آن روبرو هستیم , شناسایی کنیم .
- باید امکان هرگونه توازی را در سیستم شناسایی کنیم .
- باید شروع و پایان هر فعالیت را مشخص کنیم .
- در رسم نمودار باید تا حد ممکن ار برخورد خطوط کنترلی با هم جلوگیری کنیم .

نمونه ای از دیاگرام فعالیت گرفتن نوبت

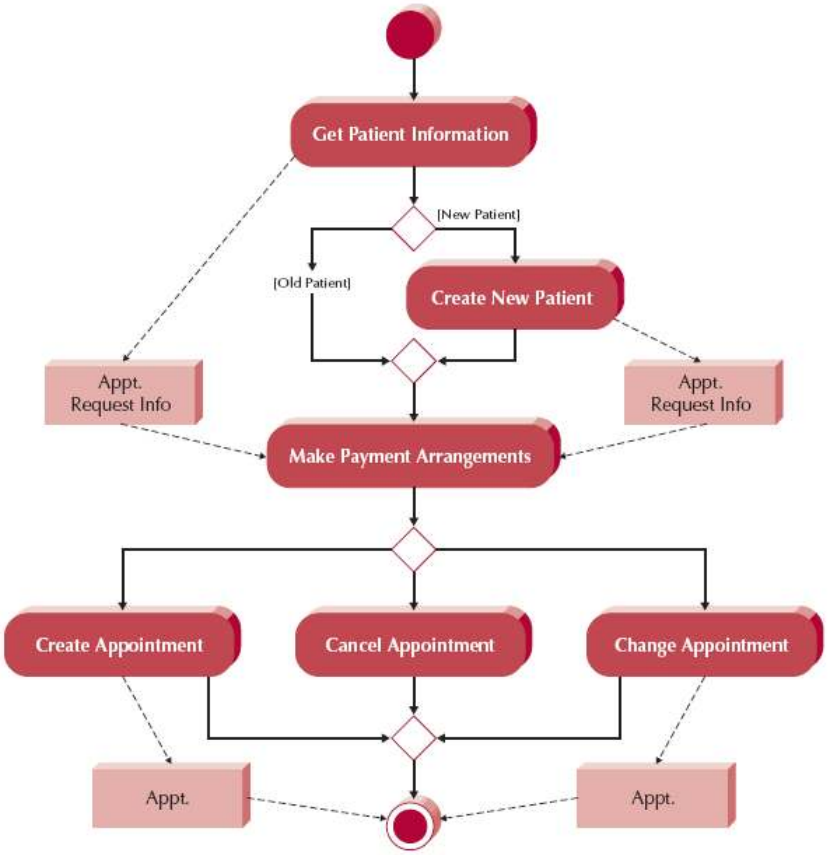
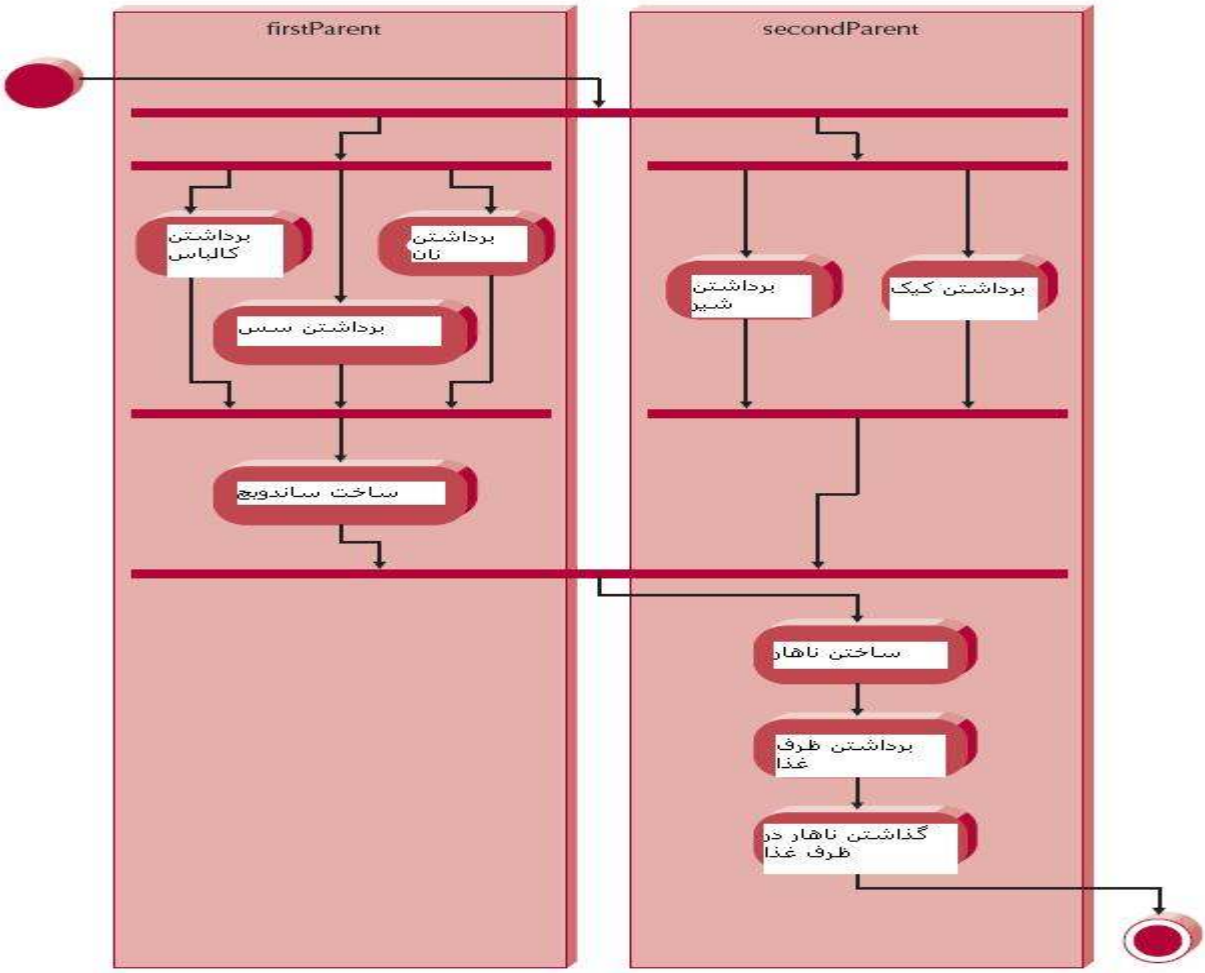


FIGURE 6-2
Activity Diagram for Appointment System

دیاگرام فعالیت برای تهیه غذای یک دانش آموز توسط والدین



USE-CASE DESCRIPTIONS

مقدمه

Use case ها توصیف ساده ای از عملیاتهای سیستم از دید کاربر میباشند . دیاگرامهای Use case این عملیاتها را به تصویر میکشند و مشخص میکنند که چه کارهایی را کاربران میتوانند انجام دهند و سیستم به چه صورتی به اعمال و درخواستهای کاربر پاسخ میدهد . رسم نمودار Use case طی دو مرحله انجام میپذیرد : تهیه توصیفگر Use case با کمک کاربران و سپس تبدیل نتایج بدست آمده به نمودار Use case و تمامی اینها براساس نیازمندیهای شناخته شده و نمودار فعالیت میباشد . توصیفگر Use case دارای تمامی اطلاعاتی که برای رسم نمودار Use case لازمند میباشد .

Use case ها مهمترین عنصرUML میباشند و تمامی نمودارهای دیگر UML نیز از Use case ها استفاده میکنند . توسط Use case ها تراکنشهایی را که سیستم با کاربران سیستم(کاربران نهایی یا دیگر سیستمها) دارد را مشخص میکنیم که این تراکنشها نشان دهنده دید خارجی یا Functional view سیستم از نقطه نظر کاربران میباشد .

هر Use case فقط و فقط یکی از اعمال سیستم را مشخص میکند(فقط یکی از Function های سیستم را) اگرچه میتوان مسیرهای مختلفی را که کاربر میتواند در طی تراکنش خود با سیستم داشته باشد , نشان دهد . به عنوان مثال اگر یک Use case بنام "جستجوی کتاب در کتاب فروشی" داشته باشیم , کاربر میتواند عمل جستجو را بر اساس موضوع کتاب , براساس نویسنده کتب و یا بر اساس نام کتاب انجام دهد. به هرکدام از مسیرهای داخل Use case یک سناریو گفته میشد .

برای پیدا کردن Use caseها میتوان از تکنیکهای جمع آوری نیازمندیها استفاده کرد . ولی بهترین راه برای تعیین Use caseها استفاده از نمودار فعالیت است . در اکثر مواقع Activity های درون این نمودار , Use case های سیستم میباشند.

نکته مهم دیگر اینکه , هر Use case فقط به یکی از نقشهایی که کاربران میوانند در سیستم بازی کنند انتصاب دارد . زیرا یک کاربر ممکن است نقشهای مختلفی در سیستم بازی کند . مثلا منشی مطب یک پزشک میتواند نقشهای زیر را در سیستم ایفاکند : ایجاد وقت ملاقات , پاسخ به تلفنها , پر کردن پرونده بیماران و ... . هرکدام از این موارد یک نقش در سیستم انجام میدهد که ممکن است تمامی انها را یکنفر بازی کند و یا هر نقش را کاربر مجزایی در سیستم ایفا کند .همچنین ممکن

است چند کاربر یک نقش را بازی کنند . مثلا هم منشی عمل ایجاد وقت ملاقات را انجام دهد هم دستیار پزشک . به هر حال Use caseها توسط کاربران اجرا نمیشود بلکه توسط نقشها در سیستم بازی میشوند .

توصیفگر UseCase

توصیفگر UseCase شامل تمامی اطلاعاتی که برای ساخت نمودار UseCase لازمند میباشد . فقط ما انها را به صورت متنی نه نموداری بیان میکنیم .

عناصر تشکیل دهنده UseCase description به شرح زیرند :

۱ – خلاصه (Overview) : در اینجا ما اطلاعاتی کلی و خلاصه را در مورد UseCase مورد نظر ارائه میکنیم . این اطلاعات شامل :

الف – نام UseCase : نام UseCase باید با یک فعل شروع و با یک اسم پایان یابد . مثلا " ثبت قرار ملاقات " .

ب – شماره UseCase : برای هر UseCase یک شماره خاص و یکتا در نظر میگیریم .

ج – نوع UseCase : هر UseCase میتواند کلی یا جزئی باشد . UseCase های کلی (Overview) کلیات را بیان میکنند و UseCase های جزئی (Detail) که جزویات UseCase را مشخص میکنند . در ابتدای تحلیل فقط UseCaseهای کلی را مشخص میکنیم .

د – آکتور اصلی : انسان یا شیئی که UseCase را آغاز میکند یا اجرا میکند. (Trigger کننده UseCase)

ه – توصیف مختصر (brief description) : توضیح هدف یا کار اصلی UseCase در غالب فقط یک جمله .

ی – سطح اولویت : اولویت ساخت UseCase را مشخص میکنیم. UseCaseهای با اولویت بالاتتر در نسخه های اولیه نرم افزار گنجاده خواهند شد . توسط مشخص کردن سطح اولویت مشخص میکنیم که کدام فرآیندهای تجاری از اهمیت بیشتری برخوردارند .

ک – Stakeholder : لیست stakeholder هایی که این UseCase برای آنها اهمیت دارد . معمولا آکتور اصلی هم در این لیسیت قرار میگیرد .

گ – رویداد فعال کننده : هر UseCase توسط وقوع یک رویداد اجرا میشود یا فعال میشود . رویداد فعال کننده UseCase را باید در اینجا مشخص کنیم . رویدادهای فعال کننده میتوانند داخلی یا خارجی باشند .

۲ – Relationships یا روابط : روابطی را که این UseCase با UseCaseهای دیگر یا کاربران سیستم دارند را در اینجا بیان میکنیم . هر UseCase میتواند چهار نوع رابطه داشته باشد :

الف – رابطه انجمنی (Association relationship) :روابطی هستند که میان UseCase و آکتوری که از UseCase استفاده میکند وجود دارد . (آکتورها همان نقشها(Role)هایی هستند که کاربران در سیستم به عهده دارند)

ب – رابطه توسعه یا تعمیم دادن (Extend relationship) : معمولا UseCaseها برای اجرای خود احتیج به اعمالی دارند که ممکن است در UseCaseدیگری انجام شود . مثلا ممکن است UseCase "گرفتن نوبت" به اجرای UseCase"ایجاد بیمار جدید" منجرشود . در اینحالت ما بین این UseCaseها رابطه Extend وجود دارد . دقت شود که در رابطه Extend همیشه اجرای UseCaseاول به اجرای UseCaseدوم منجر نمیشود . مثلا ممکن است در مثال گفته شده بیماری که میخواهد نوبت بگیرد , بیمار قدیمی بوده و UseCase"ایجاد بیمار جدید" اجرا نشود. پس این نوع از روابط فقط مابین UseCaseها برقرار بوده و زمانی استفاده میشود که " شاید یک UseCase بخواهد از یک UseCase دیگر استفاده کند" .

ج – رابطه شامل بودن (Include relationship) : زمانیکه یک UseCaseبرای اجرای خود باید از UseCaseدیگری استفاده کند . یعنی در این حالت UseCaseاول همیشه به UseCase دوم نیاز دارد .

د – رابطه تعمیم(Generalization relationship) : برای نشان دادن ارث بری از این رابطه استفاده میکنیم .

۳ – جریان رویدادها (Flow of events) : در این مرحله , مراحل انجام فرآیند تجاری UseCase موردنظر بیان میشوند . یعنی نحوه انجام کار UseCaseرا تشریح میکنیم . میتوانیم مراحل انجام کار را به سه دسته تقسیم کنیم :

الف – جریانهای معمولی رویدادها (Normal flow of events) : مراحل عادی انجام کاردر UseCase رابه ترتیب اجرایشان بیان میکنیم .

ب – Sub flows : در بعضی موارد کارهای انجام شده در UseCase ها پیچیده میشوند و بیان آنها در غالب Normal flow مشکل و مبهم است. در این حالت این کارها را به کارهای کوچکتری شکسته و آنها را درغالب Sub flowها بیان میکنیم .معمولا این جریانها براساس جریانهای مشخص شده در نمودار فعالیت بیان میشوند.

ج – جریانهای اختیاری و استثنایی (Alternative or Exception flows): کارهای و جریانهایی که ممکن است اجرا شوند یا خیر .

۴ – ویژگیهای دیگر UseCase : موارد دیگری در مورد UseCase مانند پیچیدگی UseCase , زمان اجرای آن , داده ها و اطلاعاتی که مابین آکتورها و UseCase ردوبدل میشود , وابستگیها و ... .

Use-Case Description Template

Use Case Name: ID: Importance Level:

Primary Actor: Use Case Type:

Stakeholders and Interests:

Brief Description:

Trigger:

Relationships: (Association, Include, Extend, Generalization)

Normal Flow of Events:

Subflows:

Alternate/Exceptional Flows:

Use case description Example

Use Case Name: Make appointment	ID: 2	Importance Level: High
Primary Actor: Patient	Use Case Type: Detail, essential	

Stakeholders and Interests:
Patient - wants to make, change, or cancel an appointment
Doctor - wants to ensure patient's needs are met in a timely manner

Brief Description: This use case describes how we make an appointment as well as changing or canceling an appointment.

Trigger: Patient calls and asks for a new appointment or asks to cancel or change an existing appointment.

Type: External

Relationships:

Association: Patient
Include: Make Payment Arrangements
Extend: Create New Patient
Generalization:

Normal Flow of Events:

1. The Patient contacts the office regarding an appointment.
2. The Patient provides the Receptionist with their name and address.
3. The Receptionist validates that the Patient exists in the Patient database.
4. The Receptionist executes the Make Payment Arrangements use case.
5. The Receptionist asks Patient if he or she would like to make a new appointment, cancel an existing appointment, or change an existing appointment.
   If the patient wants to make a new appointment,
   the S-1: new appointment subflow is performed.
   If the patient wants to cancel an existing appointment,
   the S-2: cancel appointment subflow is performed.
   If the patient wants to change an existing appointment,
   the S-3: change appointment subflow is performed.
6. The Receptionist provides the results of the transaction to the Patient.

Subflows:

S-1: New Appointment
1. The Receptionist asks the Patient for possible appointment times.
2. The Receptionist matches the Patient's desired appointment times with available dates and times and schedules the new appointment.

S-2: Cancel Appointment
1. The Receptionist asks the Patient for the old appointment time.
2. The Receptionist finds the current appointment in the appointment file and cancels it.

S-3: Change Appointment
1. The Receptionist performs the S-2: cancel appointment subflow.
2. The Receptionist performs the S-1: new appointment subflow.

Alternate/Exceptional Flows:

3a: The Receptionist executes the Create New Patient use case.
S-1, 2a1: The Receptionist proposes some alternative appointment times based on what is available in the appointment schedule.
S-1, 2a2: The Patient chooses one of the proposed times or decides not to make an appointment.

FIGURE 6-4 Use Case Description Example

نکاتی برای تهیه Use Case description

- مشخص کنید که چه کسی یا چه چیزی شروع کننده یک Action میباشد و در هر مرحله چه کسی دریافت کننده آن عمل است . به عنوان مثال " بیمار نام و آدرس خود را در اختیار مسئول پذیرش قرار میدهد " که کاملا مشخص است شروع کننده عمل "بیمار" است و دریافت کننده آن "مسئول پذیرش" است .
- مراحل انجام کار را از دید یک بیننده بیطرف مشخص کنید . بدین منظور باید مراحل را هم از دید شروع کننده action و هم از دید دریافت کننده بنویسید . سپس بر اساس هر دو دیدگاه مراحل واقعی را مشخص کنید . مثلا در مثال قبل , عمل گفته شده نه از دیدگاه بیمار و نه از دیدگاه مسئول پذیرش بیان شده است .
- هر کدام از مراحل را در یک سطح انتزاع بنویسید . یعنی اینکه هر کدام از مراحل به یک اندازه در تکمیل Use case مورد نظر نقش دارند . مثلا هرکدام از مراحل ۲۰٪ از اعمال Use case را شرح دهند .
- Use case باید شامل مراحل قابل حسی باشد . هر Use case باید یک تراکنش را نشان دهد . بنابراین هر Use case از چهار بخش زیر تشکیل شده است :

۱ – آکتور اصلی توسط ارسال یک درخواست یا داده به سیستم , اجرای یک Use case را آغاز میکند .

۲– سیتم از معتبر بودن درخواست اطمینان حاصل میکند .

۳– سیستم درخواست را پردازش کرده و در صورت لزوم وضعیت سیستم را تعویض میکند .

۴–سیستم پاسخ پردازش را به آکتور اصلی ارسال میکند . به عنوان مثال درتصویر قبل در قسمت Normal flow of event , ۱– بیمار درخواست یک وقت ملاقات میکند .(مراحل ۱و۲) ۲– مسئول پذیرش بررسی میکند که آیا پرونده بیمار موجود میباشد یا خیر (مرحله۳) ۳ – مسئول پذیرش وقت ملاقات را تنظیم میکند (مرحله ۴و۵) ۴ – مسئول پذیرش زمان و تاریخ وقت ملاقات را در اختیار بیمار قرار میدهد .

a.از رهیافت KISS (Keep It Simple, Stupid) استفاده کنید. درصورتیکه یک Use case پیچیده یا طولانی بود انرا به چند Use case تجزیه کنید . همچنین اگر Normal flow of events پیچیده بودند , آنها را به Sub flowها تجزیه کنید . مثلا مرحله پنجم از تصویر قبل .

USE-CASE DIAGRAMS

USE-CASE DIAGRAMS روش بسیار مناسب و ساده ای برای مشخص کردن اعمالی که سیستم انجام میدهد و کاربرانی که با سیستم در ارتباط هستند , میباشد. عناصر تشکیل دهنده نمودار Use case به شرح زیر است :

۱ – آکتور(Actor)یا بازیگر : یک آکتور به معنی یک کاربر خاص نیست , بلکه آن نقشی است که کاربر در سیستم بازی میکند و در غالب آن نقش با سیستم ارتباط برقرا میکند . همچنین آکتور میتواند سیستمی دیگری باشد که با سیستم ما در تراکنش است . هر آکتور میتواند تهیه کننده ورودیهای سیستم و یا مصرف کننده خروجی سیستم و یا هردو باشد .

۲ – Association Relationship : Use caseها و آکتورها توسط رابطه انجمنی به هم متصل میشوند . درجه رابطه برروی خط رابطه نوشته میشود که این درجه میتواند به صورت زیر باشد :

الف – یک به یک (۱..۱)

ب – یک به چند (*..۱)

ج – چند به یک (۱..*)

د – چند به چند (*..*)

۳ - Use case : Use caseها فرآیندهایی هستند که سیستم انجام میدهد و سرویسهایی را برای آکتورها به ارمغان می آورند . . معمولا ابتدا Use case diagramهای سطح بالای سیستم طراحی میشوند تا سیستم بهتر درک شود (یعنی Use caseهایی که فرآیندهای اصلی سیستم را انجام میدهند) و سپس این Use caseها به موارد کاری جزئی تری شکسته میشوند و Use case diagram های جزئی تر ترسیم میشوند .

Use caseها میتوانند برای انجام کارهای خود از Use caseهای دیگر هم استفاده کنند و روابط include , extend و generalization را با آنها برقرار کنند . به عنوان مثال , فرض کنید که هرزمان که بیمار در خواست گرفتن نوبت دارد , ممکن است همان موقع بخواهد هزینه اش را پرداخت کند و یا اینکه بعدا آنرا پرداخت کند .بنابراین ما use caseای رابنام "Make Payment Arrangements" به وجود می آوریم که برای محاسبه هزینه هایی که باید توسط بیمار پرداخت شودو همچنین برای نگهداری حساب بیماران استفاده میشود که این usecase , extend شده usecase "Make appointment" میباشد . زیرا ممکن است در برخی ملاقاتها بیمار بخواهد وضعیت پرداختیهایش را مشخص کند و در برخی دیگر نخواهد .

همچنین در برخی موارد بعضی از usecase ها دارای یکسری توابع مشترک هستند که این توابع در usecase دیگری انجام میشوند . به عنوان مثال , usecaseای در نظر بگیرید که روتینهایی را برای نگهداری ومدیریت وضعیت نوبتها در مطب

پزشک و مرتب سازی آنها انجام میدهد "Manage Schedule" . همچنین usecaseهای دیگری , یکی برای مشخص کردن وضعیت زمانبندی پزشکان و پیدا کردن زمان آزاد انها ("Record Availability") و دیگری برای تولید اطلاعاتی در مورد هر کدام از نوبتهای داده شده که در اختیار مدیر مطب گذاشته میشود("Produce Schedule information") وجود دارد که هر دوی این usecaseها از روتینهایی که در "manage appointment" وجود دارد , استفاده میکنند . بنابرای این دو usecase میتوانند آن usecaseای را که دارای روتینهای مورد نیازشان میباشد "include" کنند .

همچنین مواقعی وجود دارد که ما میتوانیم توسط روابط تعمیم و تخصیص(generalization/specialization) ارث بری را نمایش دهیم . به عنوان مثال میتوان "Make appointment" را برای بیماران جدید و بیماران قدیمی اختصاصی تر کرد و دو usecase , "Make old patient Appt" و "Make new patient Appt" را از آن به وجود آورد که این دو usecase خصوصیات "make appointment" را به ارث میبرند . زیرا ایجاد نوبت برای بیماران جدید ملزم ایجاد پرونده بیمار جدید نیز هست . یعنی برای بیماران جدید باید یک usecase دیگر هم داشته باشیم که پرونده ای برای بیمار جدید ایجاد کند . ولی بیماران قدیمی به این usecase نیازی ندارند .

۵ – Subject Boundary : میتوان یکسری از usecaseها را که برای انجام یک فرآیند خاص در سیستم استفاده میشوند را درون یک بسته گروه بندی کرد تا محدوده کاری هر کدام از usecase ها بهتر مشخص شود . با اینکار میتوانیم مشخص کنیم که کدام قسمتهای سیستم درون این usecaseها و کدامیک از آنها در خارج از محدوده قرار میگیرند .همچنین میتوانی مرز سیستم نرم افزاری و محیط اطرافش را مشخص کنیم , یا زیرسیستمهای مختلف را از یکدیگر جدا کنیم .

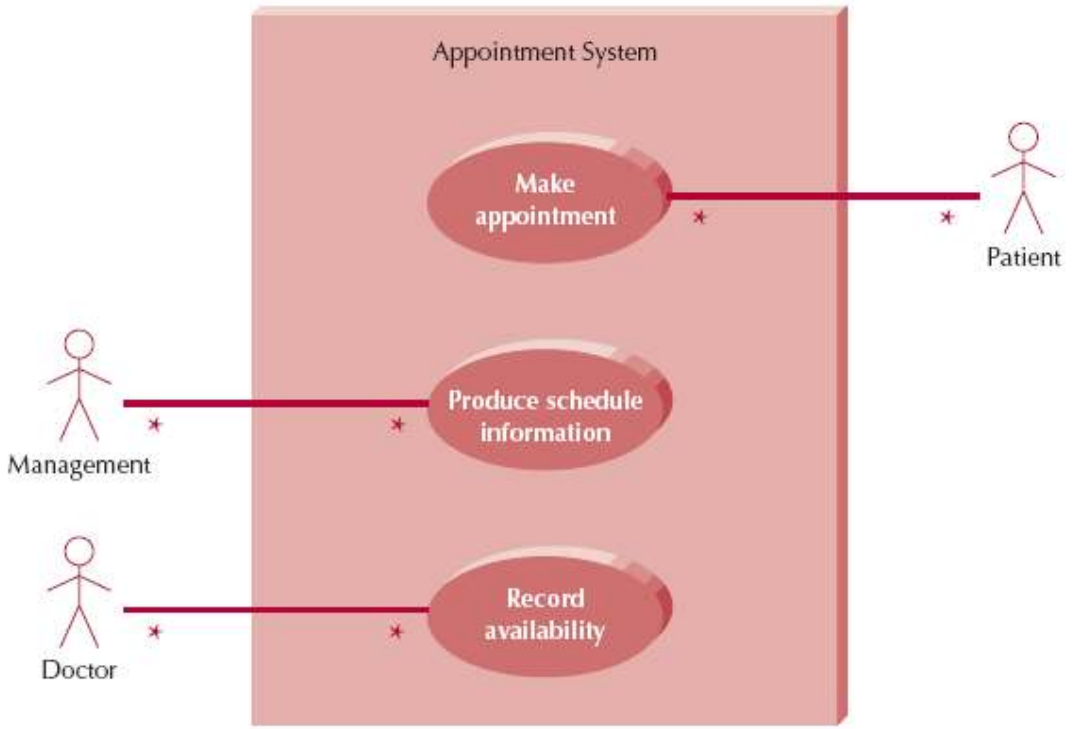
همچنین آکتورهای سیستم می توانند در داخل محدوده و یا در خارج از محدوده قرار بگبرند . اگر بخواهیم نقشها را در غالب آکتورها نشان دهیم , باید آنها را در خارج از محدوده قرار دهیم . در غیر این صورت باید آکتورها را جزئی از یکی از usecaseها در نظر بگیریم . به عنوان مثال ما مسئول پذیرش را جزئی از "Make appointment" تشخیص دادیم و انرا به عنوان آکتور داخلی این usecase در نظر گرفتیم . زیرا مسئول پذیرش قسمتی از کارهای درون این usecaseرا انجام میدهد

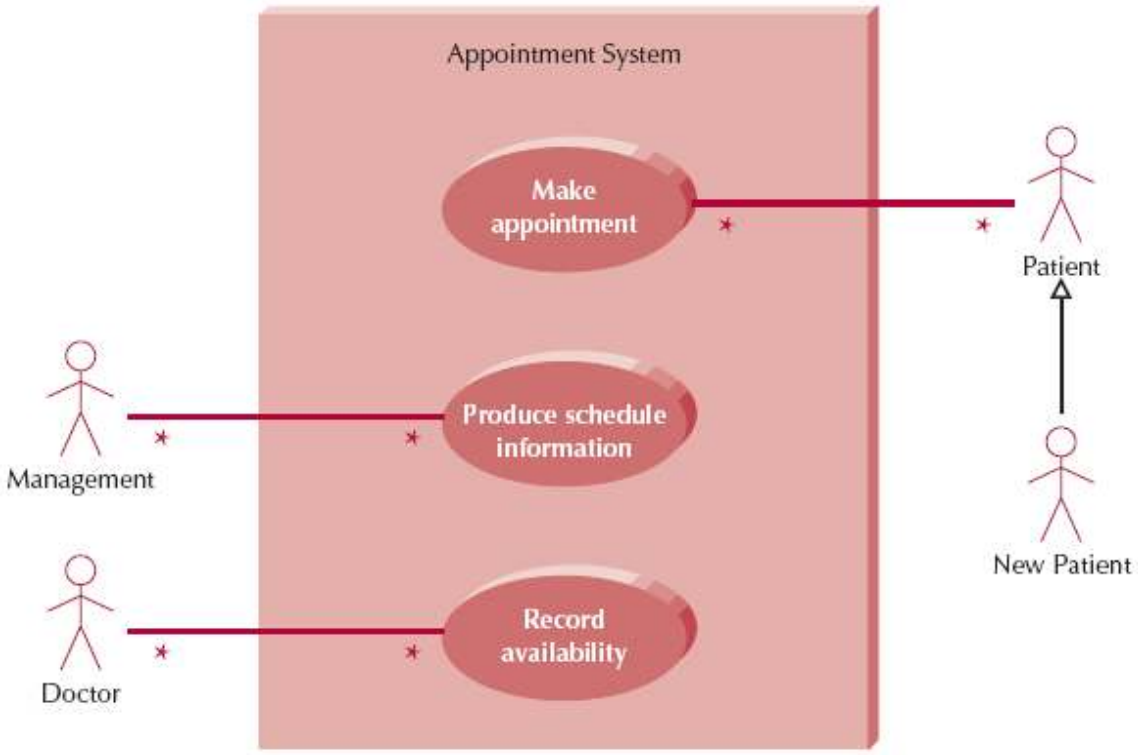
(مثلا اطلاعات بیمار را وارد میکند) . پس در سیتم ما آکتوری بنام مسئول پذیرش وجود ندارد .

An Actor: ■ Is a person or system that derives benefit from and is external to the subject ■ Is depicted as either a stick figure (default) or if a non-human actor is involved, as a rectangle with <<actor>> in it (alternative) ■ Is labeled with its role ■ Can be associated with other actors using a specialization/superclass association, denoted by an arrow with a hollow arrowhead ■ Are placed outside the subject boundary	Actor/Role <<actor>> Actor/Role
A Use Case: ■ Represents a major piece of system functionality ■ Can extend another use case ■ Can include another use case ■ Is placed inside the system boundary ■ Is labeled with a descriptive verb-noun phrase	Use Case
A Subject Boundary: ■ Includes the name of the subject inside or on top ■ Represents the scope of the subject, e.g., a system or an individual business process	Subject
An Association Relationship: ■ Links an actor with the use case(s) with which it interacts	* *
An Include Relationship: ■ Represents the inclusion of the functionality of one use case within another ■ The arrow is drawn from the base use case to the included use case	<<include>> ←----------
An Extend Relationship: ■ Represents the extension of the use case to include optional behavior ■ The arrow is drawn from the extension use case to the base use case	<<extend>> ----------→
A Generalization Relationship: ■ Represents a specialized use case to a more generalized one ■ The arrow is drawn from the specialized use case to the base use case	←

FIGURE 6-6 Syntax for Use Case Diagram

در صفحات بعد ,دیاگرام use case سیستم مطب پزشک بر اساس مطالب گفته شده نشان داده شده است .


Appointment System
Make appointment
Patient
Management
Produce schedule information
Doctor
Record availability


Appointment System
Make appointment
Patient
Management
Produce schedule information
New Patient
Doctor
Record availability

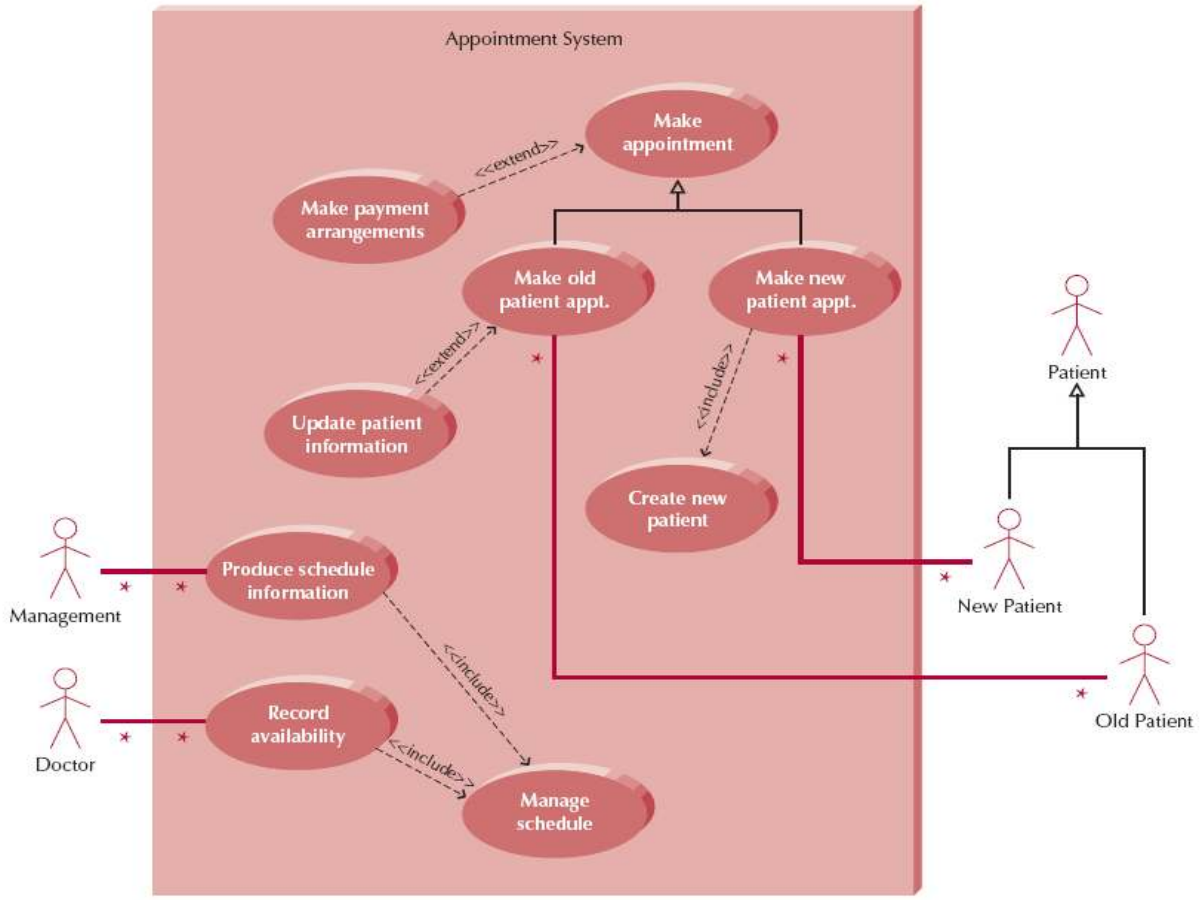


FIGURE 6-9 Extend and Include Relationships

Alan Dennis, Barbara Wixom, and David Tegarden
© 2005
John Wiley & Sons, Inc.

تهیه کننده : سید رضا شاه امیری

Email Address 1: RShahamiri@yahoo.com
Email Address 2 : Reza@RSH.ir
Web site : www.RSH.ir
Group Home page : http://groups.yahoo.com/group/SADU2
Group Email address : SADU2@yahoogroups.com

limooee.Com
لیموی
ناشر کتب الکترونیک