بسمه تعالی

دانشگاه آزاد اسلامی واحد بناب

دانشکده کامپیوتر

مهندسی نرم افزار ۱

نگارنده:

محمد ببردل بناب

سرفصل مهندسی نرم افزار (تجزیه وتحلیل)

بحران نرم افزار،علل نیاز به متدولوژی وفرآیند تولید، چرخه حیات سیستم (مشتمل برتحلیل خواسته ها، طراحی کلی، طراحی جزئی، پیاده سازی ، تبدیل ونگهداری سیستم)

❖ مفاهیم تحلیل سیستم ها

- سیستم های اطلاعاتی ساخت یافته (معرفی برخی روشهای ساخت یافته)
- مدل فیزیکی جریان داده های سیستم موجود
- مدل منطقی جریان داده های سیستم موجود
- مدل منطقی جریان داده های سیستم پیشنهادی
- مدل فیزیکی جریان داده های سیستم پیشنهادی
- مشخصات دقیق خواسته ها (فعالیت ها)
- مشخصات فرهنگ داده

❖ امکان سنجی :

(تکنولوژی نیروی انسانی ومنابع مالی و زمانی)

- تهیه گزارش امکان سنجی،نمونه سازی

❖ طراحی کلی

سیستم شامل طراحی فایل ها یا بانکهای اطلاعاتی ، تهیه فرمهای ورودی وگزارشات نهایی،طراحی واسط کاربر، طراحی ساختمان نرم افزار، تعیین مشخصات پردازش ها یا عملیات سیستم ، تعیین مشخصات فرهنگ داده ها ، تهیه گزارش طراحی کلی سیستم

- معرفی روشهای جمع آوری اطلاعات ، معرفی روشهای تخمین هزینه وبرآورد زمانی جهت انجام هریک ازمراحل سیستم
- معرفی روشهاوابزار مدیریت پروژه
- معرفی ابزار کمک به تحلیل سیستم
- معرفی ابزار کمک به طراحی سیستم
- بخش اول CASE

تاریخچه ی مهندسی نرم افزار

مهندسی نرم افزار حرفه ای است که به یاری دانش رایانه و دیگر فناوری ها و روش ها به تهیه و نگهداری نرم افزار رایانه ای می پردازد.

مسائل اصلی مهندسی نرم افزار تولید نرم افزار بر اساس موارد زیر است:

- الزامات تعیین شده
- در زمان تعیین شده
- در محدوده بودجه پیش بینی شده

مهندسی نرم افزار طراحی، برنامه نویسی، توسعه، مستندسازی و نگهداری نرم افزار با بکارگرفتن روشهای فنی و عملی از علوم کامپیوتر ، مدیریت پروژه ، مهندسی ، محدوده کاربرد، طراحی رابط، مدیریت تجهیزات دیجیتال و سایر زمینه ها است.

کاربردهای مهندسی نرم افزار دارای ارزش های اجتماعی و اقتصادی هستند، زیرا بهره وری مردم را بالا برده، چند و چون زندگی آنان را بهتر می کنند. مردم با بهره گیری از نرم افزار، توانایی انجام کارهایی را دارند که قبل از آن برایشان شدنی نبود. نمونه های از این دست نرم افزارها عبارت اند از: سامانه های توکار، نرم افزار اداری، بازی های رایانه ای، و اینترنت.

فناوری ها و خدمات مهندسی نرم افزار به کاربران برای بهبود بهره وری و کیفیت یاری میرساند. نمونه هایی از زمینه های بهبود: پایگاه داده ها، زبان ها، کتابخانه ها، الگوها، فرآیندها و ابزار.

به واسطه ی تغییرات بسیار سریع و غافل گیرکننده ی فناوری های نوین اطلاعاتی و ارتباطی و به طور خاص نرم افزار، و به موازات آن، تغییر نیازها، خواسته ها، و انتظارات استفاده کنندگان از نرم افزار و قابلیت های آن، طراحی و تولید نرم افزار، بسیار پیچیده می باشد .عوامل دیگری مانند رقابت شدید ، کمبود نیروی متخصص و حرفه ای، عدمِ دسترسی به دانش و تجربه ی موفق دیگران، لزوم تولید سریع، لزوم تولید مقرون به صرفه، نیاز روز افزون به همکاری میان رشته های مختلف، و مهم تر از همه ، عدمِ استفاده ی مناسب از اصول و مبانی مهندسی در طراحی تولید نرم افزار، این صنعت را با چالش های بسیاری روبرو نموده است.

پیشینه مهندسی نرم افزار

سرانجام برای اولین بار، در سال ۱۹۶۸ و در یک کنفرانس که توسط ناتو ۵ در کشور آلمان برگزار شده بود، بر لزوم مهندسی این دستاورد جدید بشر، یعنی نرم افزار، تأکید شد.

اصطلاح مهندسی نرم افزار در این کنفرانس شناخته شد. این اصطلاح طی کنفرانس «مهندسی نرم افزار ناتو ۱۹۶۸» (که در گارمیش آلمان برگزار شد) توسط ریاست کنفرانس F.L. Bauer معرفی شد و از آن پس بطور گسترده مورد استفاده قرار گرفت.

اصطلاح مهندسی نرم افزار عموماً به معانی مختلفی به کار می رود:

- به عنوان یک اصطلاح غیر رسمی امروزی برای محدوده وسیع فعالیت هایی که قبلا برنامه نویسی و تحلیل سیستم ها نامیده می شد.
- به عنوان یک اصطلاح جامع برای تمامی جنبه های عملی برنامه نویسی کامپیوتر، در مقابل تئوری برنامه نویسی کامپیوتر، که علوم کامپیوتر نامیده می شود.
- به عنوان اصطلاح مجسم کننده طرفداری از یک رویکرد خاص نسبت به برنامه نویسی کامپیوتر که اصرار می کند، مهندسی نرم افزار، بجای انکه هنر یا مهارت باشد، باید به عنوان یک رشته عملی مهندسی تلقی شود و از جمع کردن و تدوین روش های عملی توصیه شده به شکل متدولوژی های مهندسی نرم افزار طرفداری می کند.

محدوده مهندسی نرم افزار و تمرکز آن

مهندسی نرم افزار به مفهوم توسعه و بازبینی یک سیستم نرم افزاری مربوط می باشد. این رشته علمی با شناسایی، تعریف، فهمیدن و بازبینی خصوصیات مورد نیاز نرم افزار حاصل سر و کار دارد. این خصوصیات نرم افزاری ممکن است شامل: پاسخگویی به نیازها، اطمینان پذیری، قابلیت نگهداری، در دسترس بودن، آزمون پذیری، استفاده آسان، قابلیت حمل و سایر خصوصیات باشد.

مهندسی نرم افزار ضمن اشاره به خصوصیات فوق، مشخصات معین طراحی و فنی ای را آماده می کند که اگر بدرستی پیاده سازی شود، نرم افزاری را تولید خواهد کرد که می تواند بررسی شود که آیا این نیازمندی ها را تامین می کند یا خیر.

مهندسی نرم افزار همچنین با خصوصیات پروسه توسعه نرم افزاری در ارتباط است. در این رابطه، با خصوصیاتی مانند هزینه توسعه نرم افزار، طول مدت توسعه نرم افزار و ریسک های توسعه نرم افزار درگیر است.

نیاز به مهندسی نرم افزار

نرم افزار عموماً از محصولات و موقعیتهایی شناخته می شود که قابلیت اطمینان زیادی از آن انتظار می رود، حتی در شرایط طاقت فرسا، مانند نظارت و کنترل نیروگاه های انرژی هسته ای، یا هدایت یک هواپیمای مسافربری در هوا، چنین برنامه هایی شامل هزاران خط کد هستند، که از نظر پیچیدگی با پیچیده ترین ماشینهای مدرن قابل مقایسه اند. به عنوان مثال یک هواپیمای مسافربری چند میلیون قطعه فیزیکی دارد (و یک شاتل فضایی حدود ده میلیون بخش دارد)، در حالی که نرم افزار هدایت چنین هواپیمایی می تواند تا ۴ میلیون خط کد داشته باشد.

تکنولوژی ها و روشهای عملی

مهندسین نرم افزار طرفدار تکنولوژی ها و روشهای عملی بسیار متفاوت و مختلفی هستند، که با هم ناسازگارند. این بحث در سالهای دهه ۶۰ میلادی شروع شد و ممکن است برای همیشه ادامه پیدا کند. مهندسین نرم افزار از

تکنولوژی ها و روشهای عملی بسیار متنوعی استفاده می کنند. کسانی که کار عملی می کنند از تکنولوژی های متنوعی استفاده می کنند : کامپایلرها، منابع کد، پردازشگرهای متن. کسانی که کار عملی می کنند از روشهای عملی بسیار متنوعی استفاده می کنند تا تلاشهایشان را اجرا و هماهنگ کنند : برنامه نویسی در دسته های دونفری، بازبینی کد، و جلسات روزانه. هدف هر مهندس نرم افزار بایستی رسیدن به ایده های جدید خارج از مدلهای طراحی شده قبلی باشد، که باید شفاف بوده و بخوبی مستند شده باشد.

با وجود رشد فزاینده اقتصادی و قابلیت تولید فزاینده ای که توسط نرم افزار ایجاد شده ، هنوز هم بحث و جدل های ماندگار درباره کیفیت نرم افزار ادامه دارند.

ماهیت مهندسی نرم افزار

دیوید پارناس گفته است که مهندسی نرم افزار یک شکل از مهندسی است.

استیو مک کانل گفته است که هنوز اینطور نیست، ولی مهندسی نرم افزار باید یک شکل از مهندسی بشود.

دونالد کنوت گفته است که برنامه نویسی یک هنر است.

دیوان فعالیتهای آماری آمریکا مهندسان نرم افزار را به عنوان زیرگروهی از «متخصصین کامپیوتر»، با فرصت های شغلی ای مانند «دانشمند کامپیوتر»، «برنامه نویس» و «مدیر شبکه» دسته بندی کرده است BLS .

تمام مهندسین دیگر این شاخه علمی، که شامل مهندسین سخت افزار کامپیوتر نیز هست، را به عنوان «مهندسین» دسته بندی می کند.

مهندسی نرم افزار

به کلیه روشهاوراه حلهایی که با بکارگیری ابزار وتکنیک های مختلف باعث تولید نرم افزاری با ۳ ویژگی زیرشود:

۱) کیفیت مطلوب

۲)هزینه حداقل

۳) تحویل دروقت مقرر

مهندس نرم افزار چه کارهایی انجام می دهد؟

۱) بررسی مسئله از دیدگاههای متفاوت

۲) تجزیه وتحلیل مسئله با اصول مهندسی نرم افزار

۳)انتخاب بهترین راه حل برای انجام پروژه های نرم افزاری

۴) پیاده سازی نرم افزاری

۵) مدیریت پروژه ونظارت کامل برمراحل اجرای کارپروژه) بایستی بالا بردن کیفیت -به صرفه بودن پروژه و تحویل دروقت مقرردر نظرگرفته شود)

۶)تست پروژه برای اطمینان از نحوه صحیح عملکرد پروژه

۷) پشتیبانی پروژه

هدف مهندسی نرم افزار چیست؟

ارائه روشی جامع جهت تولید نرم افزارمبتنی بر نیازهای واقعی متقاضیان

برای رسیدن به هدف انطباق با نیازمتقاضی چه باید کرد؟

بایستی در اولین مراحل نیازها راشناخت ودرقالب مدل های گویا وبدون ابهام مطرح نمود.

نیاز متقاضیان شامل چه مواردی است؟

- نیاز به رفع مشکلات موجود در سیستم فعلی
- نیاز به اجرای وظایف اضافی
- نیاز به استفاده ازتکنولوژی روز (برتر)

بحران نرم افزاری

درنسل سوم کامپیوترها به دلیل تولید و فروش زیادکامپیوتر های شخصی احساس نیازبه تولید نرم افزار مناسب pc افزایش یافت . تولیدفراوان نرم افزار بدون هیچ گونه کنترل ونظارت باعث ایجاد نارضایتی مصرف کنندگان شد که باعث بوجود آمدن بحران نرم افزاری در دهه ۱۹۶۰ – ۱۹۷۰ گردید.

تعدادی ازدلایل بوجود آمدن بحران نرم افزاری:

۱) هزینه های بالای تولید نرم افزار

۲) عدم تطابق با نیازها ی واقعی کاربران (نارضایتی کاربر)

۳) عدم تحویل درموعدمقرر

۵) کند بودن سرعت نرم افزار درمقایسه با سخت افزار

۵) زیادبودن خطاهای نرم افزاری

۶) محدود بودن امکانات توسعه نرم افزارونگهداری وپشتیبانی محدود

برای **مقابله با بحران باید سراغ تولید نرم افزار بااصول مهندسی مطابق با قوانین و معیارهای خواص** رفت.

انواع نرم افزارها

- نرم افزار سیستمی
- نرم افزار کاربردی

نرم افزارهای سیستمی

نرم افزاری که با سیستم سخت افزاررایانه ارتباط دارد .مهمترین نرم افزار سیستمی سیستم عامل است. سایر موارد عبارتند ازنرم افزارهای فشرده سازی-آنتی ویروس ها و...

نرم افزارهای کاربردی :مثل بازی ها ، نرم افزارهای مالی ،گرافیکی ،آموزشی و...

فرایند تولید نرم افزار

به فعالیت های موردنیازبرای تولیدنرم افزار با کیفیت بالا و توسعه وپشتیبانی ونگهداری آن ،فرایند تولید نرم افزارمی گویند.

قابلیتهای نرم افزار مطا بق با اصول مهندسی :

- قابلیت اعتماد
- قابلیت نگهد اری
- سهولت درکارکردن
- قابلیت حمل (یعنی قابل استفاده با کلیه سیستم ها یا با کلیه سیستم عامل های مختلف باشد.)
- کارایی بالا
- سازگاری باانواع سیستم ها
- قابلیت توسعه
- کمترین هزینه

فرایند مهندسی نرم افزار

شامل فعالیتهای زیراست:

۱- تعریف (چه چیز؟)

مشخص کردن هدف، نیازهای موجودورفتارهای موردانتظارسیستم،جمع آوری اطلاعات ،برنامه ریزی،تجزیه وتحلیل مسئله شرح نیازمندیها ،تهیه مستندات

۲- توسعه (چگونه؟)

تولید DFD (Data flow diagram) سازماندهی داده ها ،تکمیل ارتباطات،چگونگی انجام عمل پردازش ها ،طراحی نرم افزار،کد نویسی ،عملیات تست

۳- نگهداری (تغییرات ؟)

رفع اشکالات موجود در نرم افزار،اصلاح خطاهای موجود توسعه ودر صورت نیاز تولید نر م افزار جدید.

در مهندسی نرم افزار ۱ بیشتر روی تجزیه وتحلیل وطراحی کار می کنیم که یکی از زیباترین شاخه های رشته کامپیوتر است.

تجزیه وتحلیل وطراحی مرحله ایست برای اتصال دانش کامپیوتر با نیازهای کاربردی افراد، سازمانها وبه طورکل دنیای خارج از محیط کامپیوتر.

متدولوژی چیست ؟

روش انجام کار را متدولوژی گویند. (متدولوژی یعنی روش)

هر متدولوژی باید ۲ هدف مهم را تحقق ببخشد:

۱- راه کارهای عملی برای انجام تحلیل به ما بدهد.

۲- برای نمایش نتایج تحلیل یک روش ارائه دهد .(یعنی نتایج تحلیل را چطور نشان بدهیم)

هدف ازمتدولوژی های تحلیل وطراحی

ارائه روشی مدون برای تولید ومستند سازی نرم افزار مطابق با نیازهای متقاضیان است.

چرخه حیات تولید نرم افزار

مراحل تحلیل ،طراحی ،پیاده سازی وپشتیبانی نرم افزار را چرخه حیات تولید نرم افزار می نامند.

تحلیل : در این مرحله ، نیاز کاربررا مشخص شده ، راه حلهای مختلف ارائه وبهترین راه حل جهت رسیدن به سیستم جدید انتخاب می گردد.

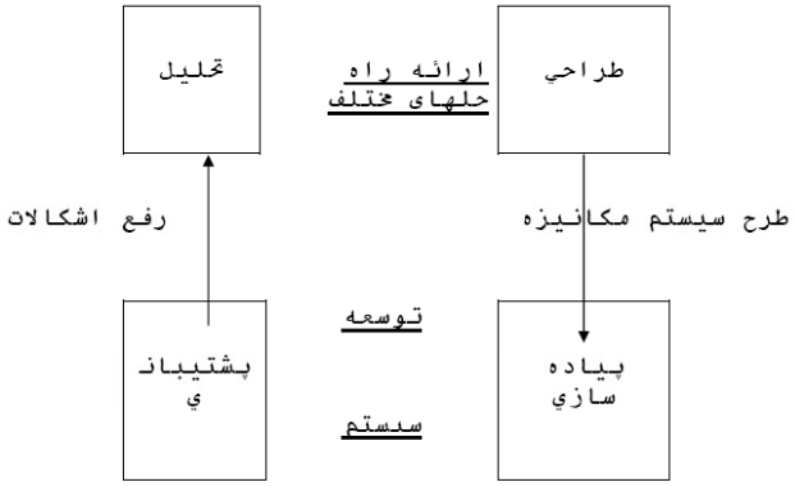
طراحی : طرح سیستم مکانیزه با تعریف نیازمندیها ی مورد نظر جهت خروجی ،ورودی،ذخیره سازی

وپردازش وکنترل های لازم ارائه می گردد.

پیاد ه سازی :کد نویسی ،تست واجرا دراین مرحله صورت می گیرد.

پشتیبانی : بررسی مجدد،رفع اشکالات وترمیم سیستم از عملیات این مرحله می باشد.

شکل زیر نشان دهنده مراحل چرخه حیات تولید نرم افزارمی باشد:



مراحل تولید نرم افزار:

۱- مطالعه امکان سنجی **(Feasebility Study)**

۲- تحلیل سیستم **(System Analysis)**

۳- طراحی سیستم **(System Design)**

۴- پیاده سازی سیستم **(System Implementation)**

۵- آزمون و اصلاح **(System Test & Integration)**

۶- اعتبارسنجی سیستم **(System Validation)**

۷- پشتیبانی فنی **(System Maintance)**

تکنولوژی چیست ؟

هرتکنولوژی یک ایده اصلی دارد و یک زمینه کاربردی. بطور مثال تکنولوژی هسته ای:

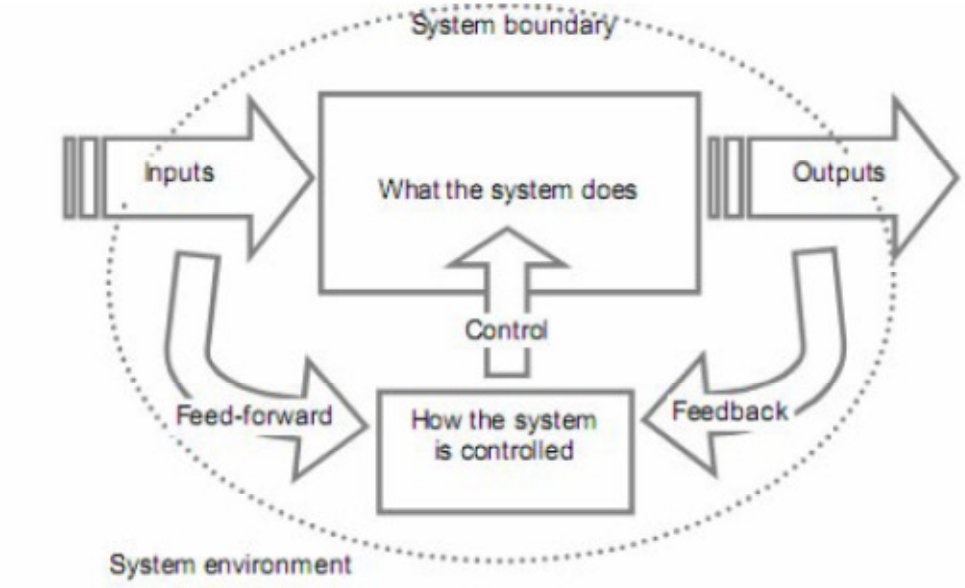
۱- ایده : در هسته اتم , انرژی هست.

۲- کاربرد : در زمینه پزشکی یا کشاورزی و.....

تعاریف سیستم

- هر شی یا پدیده ای که به نوعی دارای چرخه حیات باشد **سیستم** نامیده می شود.

- **سیستم** مجموعه ایست از اجزا به هم وابسته که در جهت رسیدن به هدف های خاص با هم در تعاملند
- **سیستم مجموعه ای ازعناصر یا مولفه های مرتبط با هم با حدود قابل تعریف که برای هدفی خاص باهم کار می کنندو روی هم اثر متقابل دارند.**
- **سیستم** مجموعه ای ازعناصرکه ورودی های مشخصی را دریافت نموده ،روی آنها پردازش انجام داده وخروجیهای معینی راتولید می نماید



- بامقایسه واقعیت هایی که از سیستم گرفته می شود (feedback) وپیش بینی ها واستانداردهای ازقبل تعیین شده ،می توان اصلاحات لازم را انجام داد.
- سیستم مجموعه ای از اجزا که با هم هدف خاصی داشته باشند گفته می شود. در مهندسی نرم افزار با سیستمهای اطلاعاتی روبرو هستیم . سیستمهای اطلاعاتی دارای ۵ مولفه دارند:

۱-مجموعه ای از ورودیها

۲-مجموعه ای از خروجیها

۳-مجموعه ای از فایلهای اطلاعاتی که اطلاعات سیستم در آنجا ذخیره می شود.

۴-مجموعه ای از پروسسها که بیانگر عملیات سیستم می باشد.

۵-هدف

تحلیلگر سیستم باید توانمندیهای زیر را داشته باشد:

۱-آشنایی با روشهای تحلیل.

۲-دانش فنی کامپیوتری داشته باشد.

۳-اطلاعات دامنه ای داشته باشد.

۴-روابط عمومی قوی داشته باشد.

سیستم ها درارتباط بامحیط جانبی:

سیستم باز : بامحیط بیرون ارتباطات پویاوتاثیرات دوجانبه دارد.

سیستم بسته : بدون نیازبه ارتباط با محیط خارج می باشد.

- دریک سیستم بسته اعمال از قبل معین وسیستم وابسته به عناصر داخلی خود است.
- تعامل بین سیستم ومحیط آن به صورت ماده ،انرژی ،اطلاعات است.
- اگرفقط انرژی از سیستم خارج شود،سیستم بسته است.
- درسیستم های اطلاعاتی سیستم بسته نداریم.

محیط سیستم شامل مجموعه سیستم هایی که با آن در ارتباط هستند.

ویژگیهای سیستم:

- هدفمندی
- نظم
- کلیت
- وابستگی اجزا سیستم
- تعامل
- تفکیک پذیری
- هم پایانی

یکی از صاحب نظران علم نرم افزار- **یوردون** - سیر تکاملی روشها ومتولوژیهای تولید سیستم را در نسل ۳ خلاصه می نماید.

روشهای نسل اول

مبتنی برانواع تکنیک های ساخت یافته (با ایده حل مسایل پیچیده با شکستن آنها به مسائل ساده تر) برای تولید نرم افزار می باشند.

این روش ها دودیدگاه برای شناخت دارند:

شناخت وظایف <------ پردازش گرا (Process oriented)

شناخت داده < ------ داده گرا (Data oriented)

روش های ساخت یافته عبارتند از:

الف)برنامه نویسی ساخت یافته : دارای ویژگی های زیرهستند

۱- استفاده کمتراز GO TO

۲- دارای ساختارهای ترتیب،تصمیم گیری وکنترل حلقه

۳- خوانایی بیشتر برنامه وردیابی سریعتر

۴- ابزار طراحی مورد استفاده :

فلوچارت ، شبه دستورالعمل (seudo code)، دیاگرام های فعالیت (action diagram)

اشکال این روش نادیده گرفتن ساختار و ارتباط بین ماژولهای یک برنامه است که طراحی ساخت یافته این اشکال رابرطرف می کند.

ب)طراحی ساخت یافته : مبتنی برطراحی بهینه ماژول های برنامه برتراند مایر ۵ ویژگی را برای طراحی بهینه ماژول های برنامه مطرح می کند:

۱- ماژولها باید فابل بیان توسط زبان برنامه نویسی مورد نظر طراح باشد.

۲- هر ماژول باید کارهای وابسته به هم راانجام دهد (افزایش همبستگی –cohesion)

۳- ارتباط بین ماژولها با ید به حداقل برسد (کاهش جفت شدن –coupling)

۴ - ارتباط بین ماژولها بایدواضح وروشن باشد.

۵ - اطلاعت یک ماژول بایستی منحصر به خود آن ماژول وتنها قابل دسترسی ازطریق همان ماژول باشد واز خارج ماژول نتوان به آنها مستقیما دسترسی داشت.

۶- تعدادپارامترهای رد وبدل شده بین ماژول ها حتی المقدور کم باشد.

اشکال این روش عدم ارائه روالی معین برای شناخت کامل وبدون ابهام نیازهای متقاضیان می باشدکه انالیز ساخت یافته ان را بر طرف می کند.

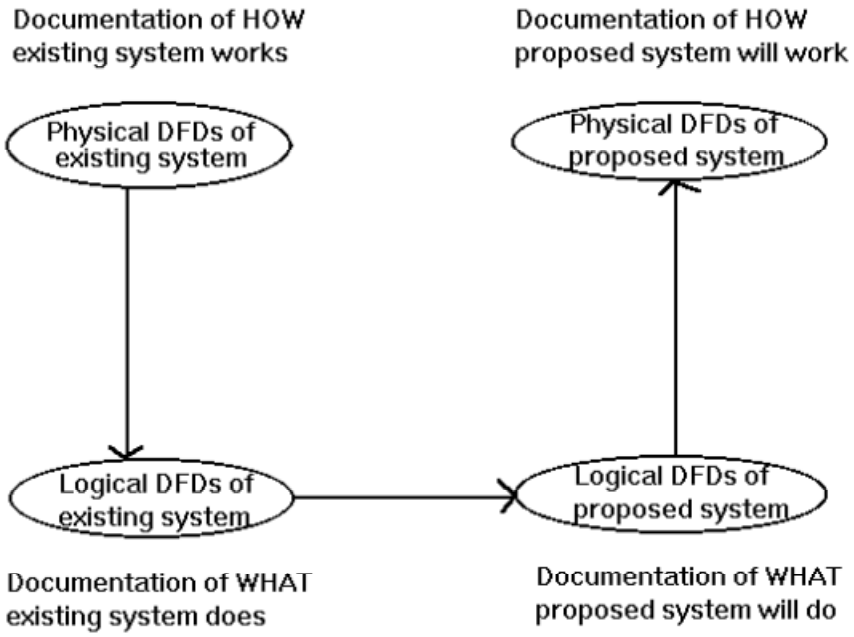
ج)آنالیز ساخت یافته :مبتنی بر شناخت چگونگی گردش داده ها درون سیستم می باشد .این روش پردازش گرا process oriented می باشد.

دراین روش تمرکز روی شناخت نیازهاست .شناخت نیازها معمولااز بالا به پایین (Top-Down) وبر اساس شکستن وتبدبل گام به گام مسائل ونیازهای مبتنی بر سیاست کاری سیستم مورد بحث ،مستقل از سخت افزار ونرم افزار مورد استفاده برای مکانیزه کردن سیستم می باشد .درسیستم های بزرگ شناخت ازپایین به بالا (Down to Up) صورت می گیرد.

د) **مدل سازی داده ها** می توان جریان گردش داده های درون یک سیستم را توسط نمودار هایی به نام نمودار گردش داده (DFD : Data Flow Diagram) به تصویرکشید.

DFD برای مدلسازی چگونگی عملکرد سیستم موج وودیا سیستم مورد نیاز به کار می زود برای هر یک از این سیستم ها می توان دیاگرام های فیزیکی ومنطقی را ترسیم کرد :

- **مدل فیزیکی سیستم موجود** : چگونگی اجرای عملیات سیستم فعلی (به همراه جزئیاتی که در عمل وجود دارد)را نشان می دهد.
- **مدل منطقی سیستم موجود** : با حذف جزئیات از مدل فیزیکی سیستم موجود، مدل منطقی سیستم موجودکه نشانگر تئوری عملیات سیستم است(آنچه که هست) ، به دست می آید.
- **مدل منطقی سیستم پیشنهادی** (جدید یاجایگزین) : با در نظر گرفتن خواسته ها ونیاز ها ولحاظ کردن آنها در مدل منطقی سیستم موجود ، مدل منطقی سیستم جدید(انچه که باید باشد) بدست می آید.
- **مدل فیزیکی سیستم جدید** : با افزودن جزئیات در ارتباط با چگونگی انجام عملیات به مدل منطقی سیستم جدید، مدل فیزیکی سیستم جدید ، به دست می آید.



روشهای نسل دوم : به سیستم از دودیدگاه، هم شناخت داده وهم شناخت وظایف می نگرند ووابسته به ترسیم دیاگرامهای مختلف برای مدلسازی سیستم می باشند.

- وظیفه یک سیستم در ارتباط با چگونگی واکنش سیستم در مقابل اتفاقات موثر برآن سنجیده وشناخته می شود (درابتدا لیستی از عملیات موثر و واکنش های سیستم به آنها تهیه می شود).
- عامل دوم شناخت موجودیت های سیستم وارتباطات آنها با یکدیگر است .(یک موجودیت ، عنصری است که در مورد آن داخل سیستم داده و اطلاعاتی ذخیره می شود.)

مثال : در سیستم خرید ،تقاضای خرید کالایک اتفاق وفرم درخواست خرید یک موجودیت می باشد.

نکته : در این روش برای ترسیم دیاگرام ها ازابزار CASE استفاده می شود. مراحل آنالیز(تجزیه وتحلیل)و طراحی جدا از هم هستند.

روش های نسل سوم : مراحل آنالیزوطراحی در ارتباط با هم می باشند.در این روش مدلها مستقل نبوده وسیستم به صورت مجتمع موردمطالعه وبررسی قرارمی گیرد به طور همزمان وجهت دستیابی به مدل صحیح تکرار می شوند.در این روش نیز از ابزار **CASE** استفاده می شود

چند نمونه **CASE**:

System_architect ، Iconix ، Ecerlertor ، power_builder ، Case Studio ، …

تحقیق ۱ : ابزار مدیریت پروژه *(Project management tools)*

تحقیق ۲ : روش های نسل چهارم؟

ابزار CASE :

نرم افزارهایی هستندکه برای راهنمایی تحلیل گرو ایجاد مستندات مبتنی برترسیم دیاگرامها براساس متدولوژیهای خاص ایجاد شده اند.

سیستمهای اطلاعاتی Information system

سه مولفه اصلی سیستمهای اطلاعاتی عبارتند از:

۱ – داده (Data) و اطلاعات (Information)

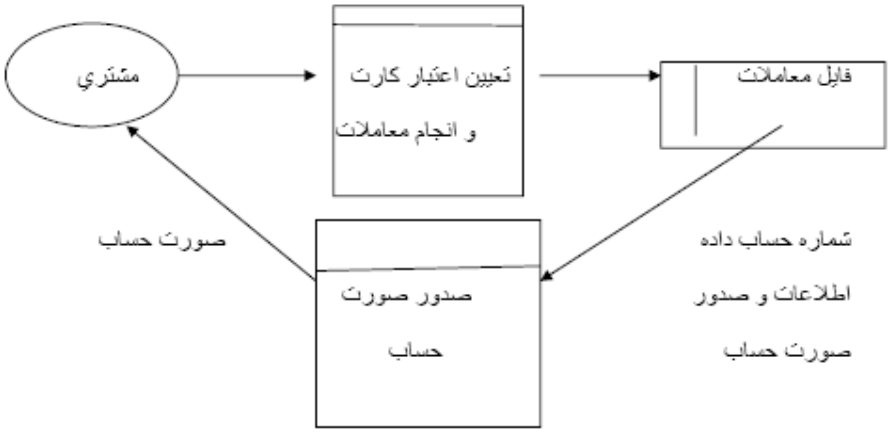
- **داده :**

حقایق خام در مورد افراد ،اشیا و وقایع یک سازمان می باشد.مانند شماره حساب وموجودی انبار

➢ **اطلاعات**

داده هایی که پردازش شده وبه فرم مناسب قابل تفسیر در امده اند.مانند صورت حساب

۲- **جریان داده (Data Flow) :** حرکت داده ها از یک محل به محل دیگر در سیستم ، با مشخص کردن مبدا و مقصد جریان داده،می باشد .به عنوان مثال وقتیکه مشتری کارت اعتباری اش را برای پرداخت چیزی که خریده به کارمی برد،شماره حسابع ش ضبط می شود.سپس شماره حساب بانکی در فابلی ذخیره می شود ویا در موقع لزوم برای تهیه صورت حساب یا آماده کردن آدرس پستی برای خریدهای بعدی مورد استفاده قرار می



گیرد.

۳- **منطق پردازش (Processing Logic)** : مراحلی که طی آن داده ها منتقل می شوند، حرکت می کنند و توصیفی برای وقایعی که برای آنها اتفاق می افتد .مثال:

```
Event::
hours_work=ziro
Event action::
If hours_work > ۴۰ then
Pay=۴۰ *pay_rate +(hours_work -۴۰)*(۱,۵*pay_rate)
Else pay= pay_rate *(hours_work)
End if
```

hours_work ساعت کارکرد ، pay_rate نرخ یک ساعت و pay حقوق دریافتی می باشد.

نمونه هایی ازسیستم ها ی اطلاعاتی موجود وتوسعه آنها

- سیستم پردازش تراکنش معاملات (TPS :Transaction processing system)

این سیستم ها روی داده های تجاری که مربوط به یک فعالیت تجاری است و معاملات ماشین بدون دخالت انسان، کار می کنند .مثل سیستم های فروش حسابداری

هدف از توسعه TPS بهبود بخشیدن پردازش اطلاعات معاملات باموارد زیرمی باشد:

۱ - بالا بردن سرعت

۲ - استفاده از نیروی انسانی کمتر

۳ - توسعه کارایی

۴ – دقت

۵ – ترکیب با سایر سیستم های اطلاعاتی

- سیستم های مدیریت اطلاعات (MIS:Management Information System):

این سیستم ها اطلاعاتی را از سیستم های TPS گرفته، به فرم مناسب مدیران جهت هدایت و رهبری سیستم تبدیل می کنند.

- سیستم های پشتیبانی تصمیم گیری (DSS:Decission Support System):

این سیستم ها اطلاعاتی برای کسانی که تصمیمات سازمانی اخذ می کنند ،طراحی شده ومحیطی تعاملی برای تصمیم گیری ایجاد می کنند.

- سیستم های خبره یا کارشناس (ES:Expert System):

این سیستم ها براساس دانش کارشناسان که به صورت پایگاه دانش موجود است ،کار می کنند.

مثال : سیستم خبره تشخیص عیب اتومبیل،لیست کامل خطاهای یک اتومبیل را دارد.

- سیستم های اتوماسیون اداری(به کار بردن کامپیوتردرکارهای اجرایی سازمان ها وادارات و ....)

این سیستم ها اطلاعاتی به بشر کمک می کنند کلیه کارهای اد رای را با استفاده از سیستم زودتر و راحتتر انجام دهد. به عنوان مثال سیستم دبیرخانه .

- و ...

چرخه عمر توسعه سیستم ها سیستم (SDLC:System Development Life Cycle) :

چرخه عمر توسعه سیستم ها سیستم متدولوژی معمول مورد استفاده در تولید، توسعه، نگهداری و جایگزینی سیستم های اطلاعاتی است که می تواند ۳ تا ۲۰ فاز یامرحله داشته باشد.ویژگی هایی که در چرخه عمر وجود داردعبارتند از:

۱) معمولاً مراحل پشت سرهم به نظر می رسند.

۲) امکان دارد فعالیت های یک مرحله به موازات مرحله دیگر انجام می شود

۳) بعضی وقتها چرخه ی عمر تکراری است

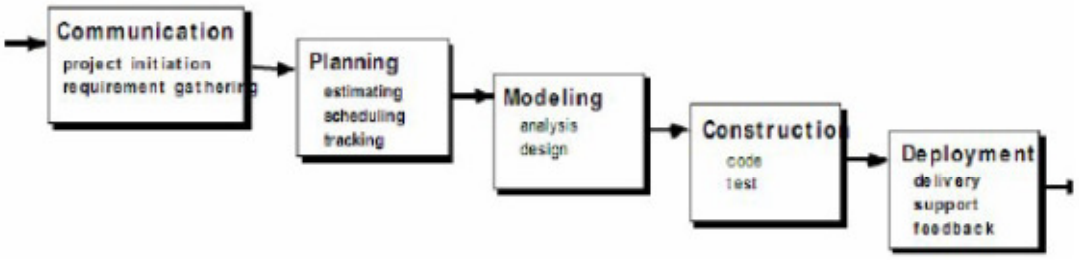
۴) می تواند بعنوان یک پروسه دورانی در نظر گرفته شود که انتهای عمر مفید یک سیستم منجر به آغاز پروژه دیگری می شود

مدل های مختلف چرخه عمر تولید نرم افزار

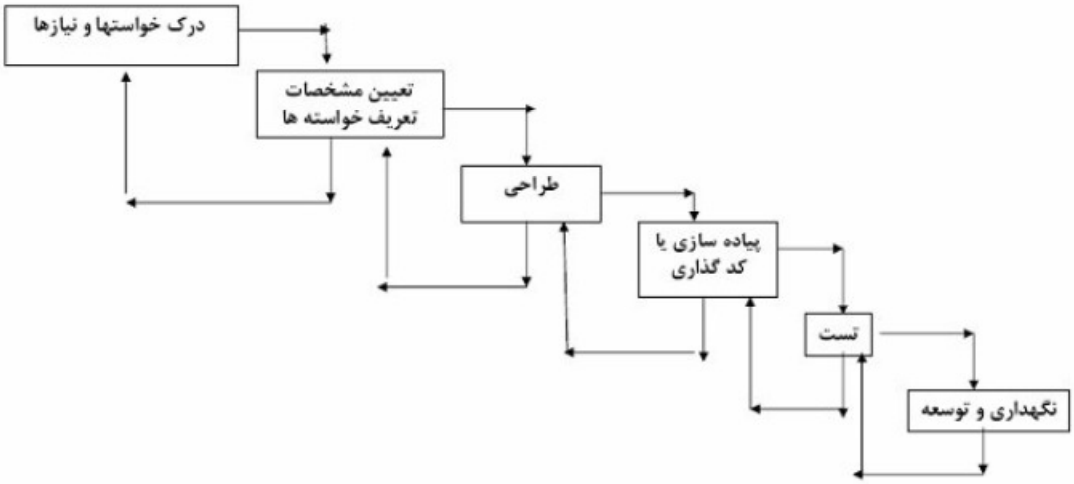
تعدادی از مدل های مختلف چرخه عمر تولید نرم افزارعبارتند از:

۱) مدل خطی (Linear)

تحلیل نیازها ← طراحی نرم افزار ← کد و برنامه نویسی ← تست ← تحویل به مشتری

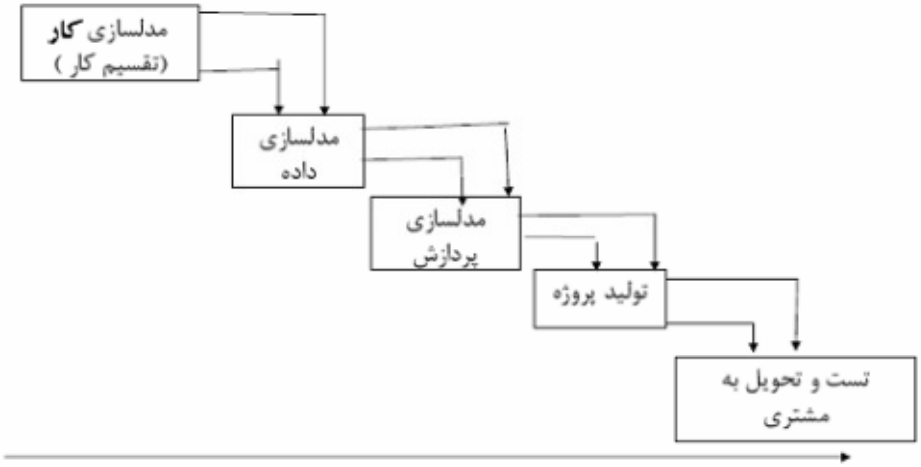


۲) مدل آبشاری (Water fall)



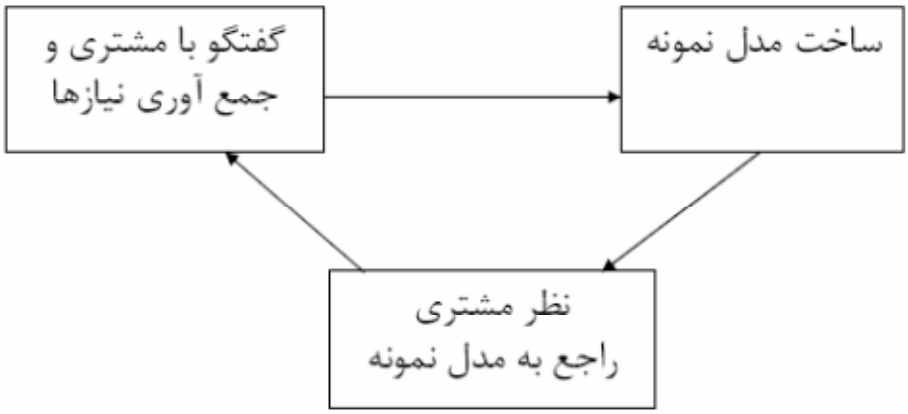
در مدل آبشاری کلاسیک بازگشت تمامی مراحل به مرحله اول بود.

۳) مدل توسعه سریع (RAD: Rapid Application Development)

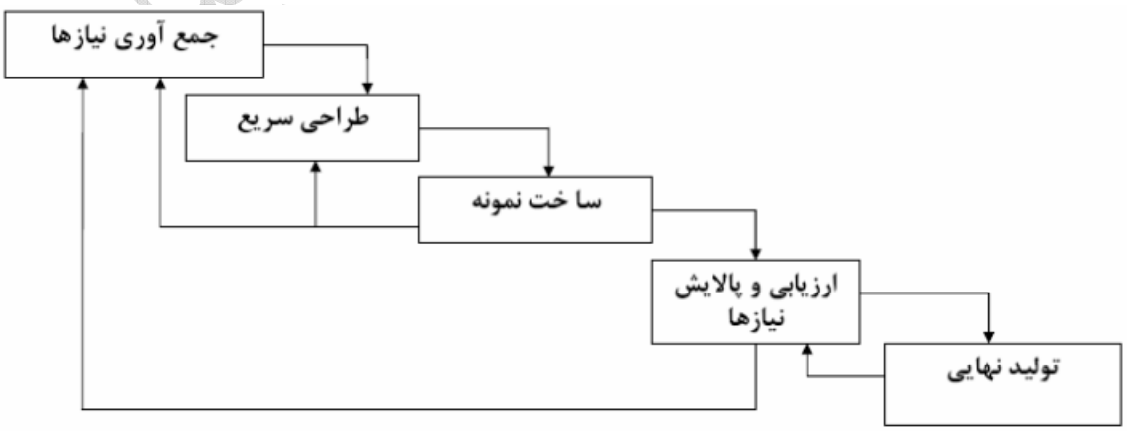


٤) مدل نمونه سازی (Prototyping)

یک مدل عملیاتی از یک یا چند سیستم به منظور ارزیابی مفروضات است که با سرعت و هزینه کم ساخته می شود.



عملکرد مدل نمونه سازی **مزایا:**



مدل نمونه سازی

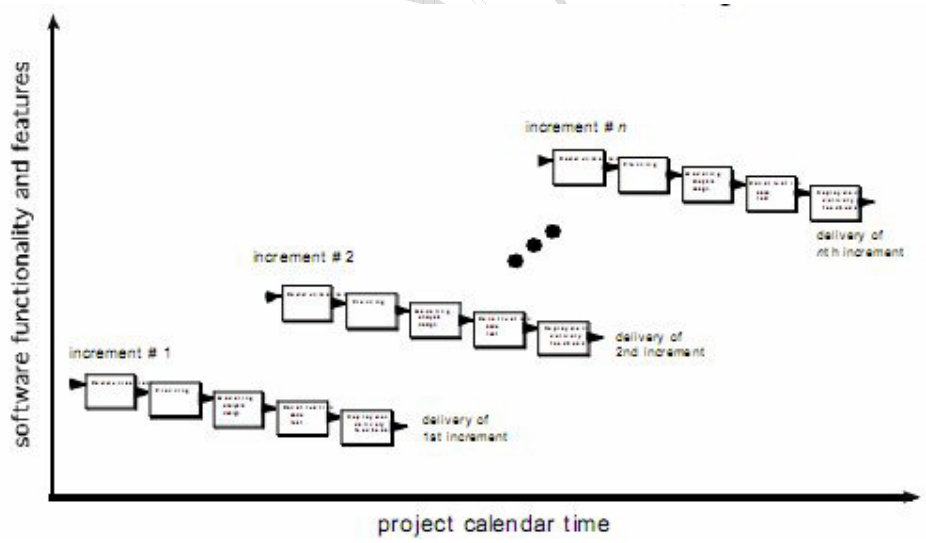
مزایا:

- امکان تغییروجمع آوری نیازها(روشن شدن ابهامات در تشخیص نیازها)
- ارتباط همیشگی بین مشتری و تولید کننده (برقراری ارتباط بهتر)
- آموزش ایجاد نرم افزار
- کاهش مستندات
- کاهش هزینه نگهداری

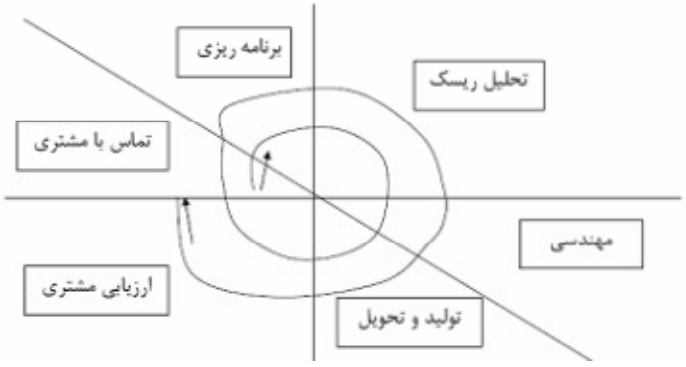
معایب:

- عدم درک کاربران از نقش نمونه سازی (ایجاد تصور غلط از برنامه اولیه برای مشتری)
- ناخوشایند بودن تغییرات پی درپی
- عدم انتخاب سیستم عامل و زبان برنامه نویسی مناسب (به جهت تسریع و تحلیل اراه برنامه)
- محدود به کارایی منابع سخت افزاری
- همجواری کاربر و برنامه نویس

۵) مدل افزایشی (Incremental)



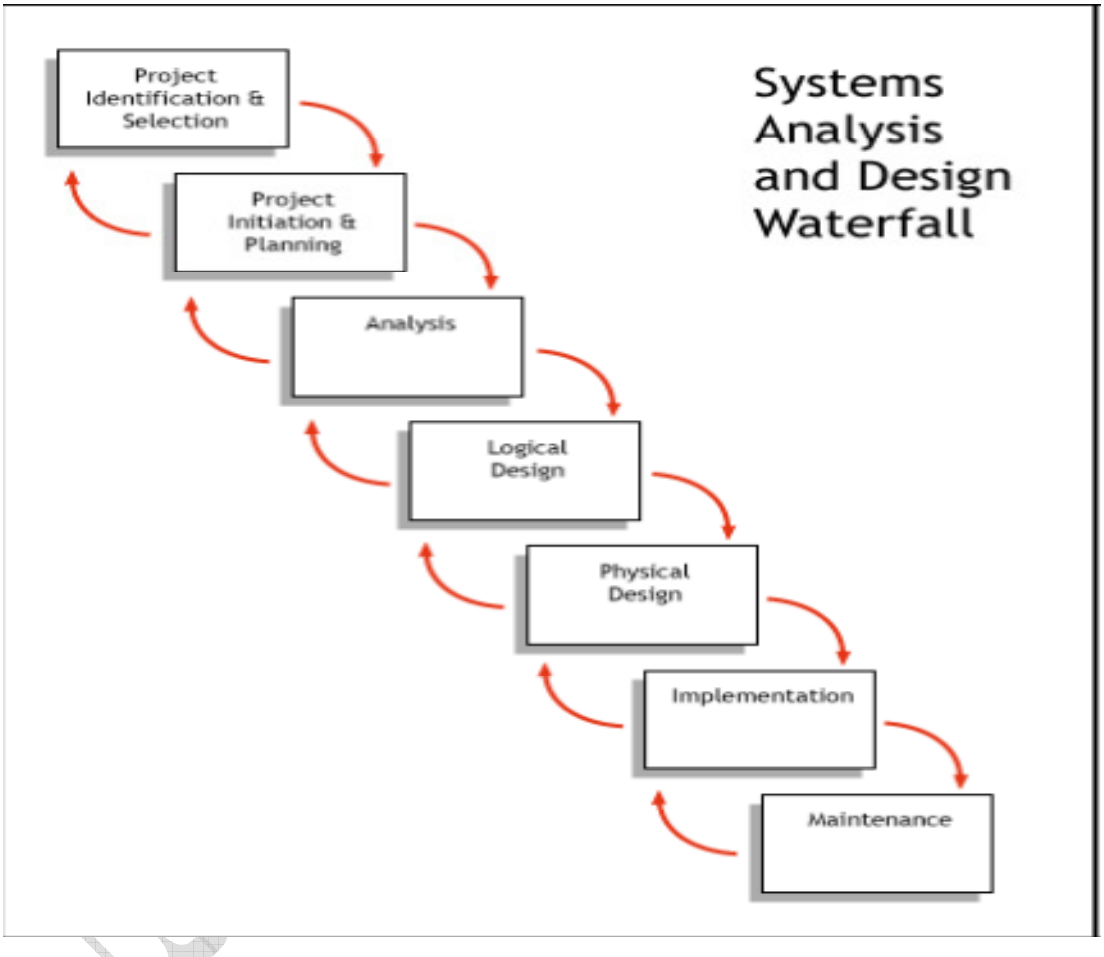
٦) مدل حلزونی (Spiral)



و...

تحقیق ۳ : مدل های چرخه عمر تولید نرم افزار(توضیح ،شکل ، مزایا ومعایب)

مدل پلکانی یا آبشاری بررسی شده Modern system analysis and design به شکل زیر است:



مراحل تولید نرم افزار

مرحله اول انتخاب و تعریف پروژه:

۱) بررسی اطلاعات مورد نیاز یک سیستم بعنوان یک کلیت و تعریف پروژه هایی که بتواند این نیازها را برطرف کند.نیازها می تواند یکی از موارد زیر باشد:

- رفع مشکلات موجود در سیستم فعلی
- اجرای وظایف اضافی
- نیاز به استفاده از تکنولوژی برتر

۲) اولویت بندی نیازها

۳) سپس آماده کردن طرحی شامل زمان بندی و توسعه سیستم های عمده جدید

۴) تعیین هدف سیستم پیشنهادی

مرحله دوم شروع پروژه وطراحی (طرح پروژه):

۱) تحقیق و مطالعه مقدماتی مشکلات سیستم (نقاط ضعف) و نقاط قوت سیستم

۲) ارائه طرح پیشنهادی و مشخص کردن زمان و منابع مورد نیاز(امکان سنجی)

۳) تشکیل تیم پروژه (تیم اولیه)

۴) تهیه طرح پروژه

مرحله سوم تجزیه و تحلیل (مطالعه سیستم موجود و ارائه سیستم های جایگزین):

۱) مطالعه عملیات فعلی سازمان و سیستم های اطلاعاتی مورد استفاده

۲) تعیین نیازمندی ها و تشخیص خواسته کاربران از سیستم (سیستم فعلی چه می کند؟ کاربران دوست دارند سیستم جدید چگونه باشد؟)

۳) مطالعه نیازمندیها و ساختار آنها مطابق با ارتباطات داخلی و حذف افزونگی ها (سازماندهی نیازمندیها ،یعنی درسطح منطقی مستقل ازهرپیاده سازی فیزیکی، جریانات داده ،منطق پردازش و جریان پردازش چگونه است؟)

۴) ایجاد سایر طرحهای اولیه منطبق با نیازمندی ها

۵) مقایسه روش ها و انتخاب بهترین روش) از لحاظ کلیت ،وظایف ،سطوح فنی، مطابقت با خواسته ها و نیازهای کاربران)

درپایان این مرحله راه حل پیشنهادی تیم تجزیه و تحلیل پروژه به صاحب نظران ارائه شده و در صورت تائید آنها امکانات سخت افزاری ونرم افزاری تهیه می گردد.

مرحله چهارم طراحی منطقی (توصیف ویژگی های عملیاتی سیستم انتخابی مستقل از کامپیوتر):

تعیین مدل منطقی : تعیین ویژگی های مشروح و عملیاتی تمام عناصر سیستم از جمله داده هاوپایگاه های داده ، پردازش ها ،ورودی ها ، خروجی ها ، کنترل ها

مرحله پنجم طراحی فیزیکی (تبدیل مدل منطقی به فیزیکی):

برگرداندن دیاگرام ها به واحدهای کوچکتربرای تبدیل به زبان برنامه نویسی (ویژگی های مشروح فنی تمام عناصر سیستم شامل برنامه ها ، فایل ها ، شبکه ها ، نرم افزار سیستم و غیره)

- تعیین زبان برنامه نویسی

- تعیین ساختار فایل ها و پایگاه های داده.
- تعیین سطوح سخت افزاری ، سیستم عامل ، محیط شبکه و غیره

مرحله ششم پیاده سازی

- شامل عملیات کد نویسی ، تست ، نصب ،آموزش کاربران ومیتندسازی می باشد.

مرحله هفتم نگهداری(پشتیبانی)

- رفع اشکالات احتمالی
- انجام تغییرات

(نسخه های جدید نرم افزار، به روز رسانی های مربوط به مستندات، آموزش و پشتیبانی)

تحلیل نرم افزار

تحلیل ساخت یافته :

محصول تحلیل، مشخصه تحلیل نرم افزار است و شامل مراحل زیر می باشد :

- محدوده اطلاعات نرم افزار
- توابع یا عملکرد نرم افزار
- رفتار نرم افزار
- کارایی
- محدودیتها
- DD فرهنگ داده ها
- Help نرم افزار
- اعتبارسنجی

مراحل ایجاد مشخصه تحلیل

- تشخیص مسئله: جمع آوری اطلاعات
- ارزیابی و سنتز راه حل: با ارزیابی هایی که انجام می دهیم راه حلی را انتخاب می کنیم
- مدلسازی: بعد از انتخاب راه حل، از روی آن مدل می سازیم.
- مشخصه: ترکیب مدلها مشخصه را ایجاد می کند.

روشهای جمع آوری اطلاعات :

- پرسشنامه
- کار کردن و حضور در محیط
- مصاحبه
- مشاهده و مرور مستندات و رویه های کاری
- FAST
- QFD
- نمونه سازی

۱- مهارت های تحلیلی:

- فکر سیستمی

- دانش سازمانی (آشنایی با نحوه کار بخش های مختلف یک سازمان ، عملکرد و هدف هر یک ، رابطه بخش ها با یکدیگر ، ارتباط با مشتری و تولید کنندگان)
- تجزیه و تحلیل و حل مسئله

۲- مهارت های فنی - آشنایی با تکنولوژی های متفاوت در مورد:

نحوه کار میکرو کامپیوتر ها ، ایستگاههای کاری ، مینی کامپیوتر ها، کامپیوتر های بزرگ و سوپرکامپیوترها انواع سیستم عامل کامپیوتر تک و شبکه انواع زبانهای برنامه نویسی سیستم های مدیریت فایل و پایگاه های داده استانداردهای ارتباط داده ونرم افزار های لازم برای ارتباط شبکه های محلی و گسترده محیط ها و ابزار توسعه سیستم ها (نظیر تولید کننده های گزارش ها و فرم ها رابط گرافیکی وابزار طراحی) ابزار تحلیل داده و تولید کننده های سیستم های DDS

۳- مهارت های مدیریتی

مدیریت منابع (Resource management) افراد ، اسناد ، ، تکنولوژی ، پول :

شامل توانایی های زیر است:

- پیش بینی برنامه ریزی استفاده از منابع (بودجه بندی)
- پیگیری و حسابرسی منافع
- یادگیری چگونگی استفاده موثر از منابع
- ارزیابی کیفیت منابع استفاده شده
- حفاظت منابع از استفاده نابجا (نا مناسب)
- متوقف کردن استفاده ازمنابع وقتیکه نیازی به آنها نیست.

مسئولیت های تحلیل گر:

۱- امکان سنجی پروژه

۲- تحلیل و شناخت مسائل و راه حل های آنها برای سیستم موجود

۳- تعیین نیازمندیها جهت توسعه و بهبود و یا جایگزینی سیستم موجود

۴- ارزیابی راه حل ها و پیشنهادات متفاوت برای انجام پروژه

۵- تعیین سخت افزار و نرم افزار لازم

۶- طراحی صفحات رابط کاربر و روالهای برنامه نویسی

۷-نظارت بر پیاده سازی و نصب سیستم

تجزیه و تحلیل (Analysis) :

منظور از تجزیه و تحلیل تعیین اطلاعات و سرویس های پردازش اطلاعاتی است که برای حمایت از اهداف انتخاب شده و عملکرد سازمان مورد نیاز است ، در نتیجه تجزیه و تحلیل یک فعالیت هوشمندانه است که در آن تحلیل گران ثبت و سازماندهی اطلاعات را بر عهده دارند . جمع آوری اطلاعات در مورد سیستم های موجود و جایگزین تعیین ملزومات نامیده می شود.

تجزیه وتحلیل شامل فعالیت های زیر است:

- مطالعه عملیات فعلی سازمان و سیستم های اطلاعاتی مورد استفاده
- تعیین ملزومات و تشخیص خواسته کاربران از سیستم
- مطالعه ملزومات و سازماندهی آنها(مطالعه ساختار ملزومات)مطابق با ارتباطات داخلی و حذف افزودنگی جمع آوری داده ها و تجزیه و تحلیل آنها برای اینکه بفهمیم سیستم در چه وضعیتی است ؟ سیستم چه کارهایی باید انجام دهد؟ چه چیزهایی باید به سیستم اضافه یا کم شود؟ نیازهای استفاده کنندگان چیست؟

روش های معمول برای جمع آوری ملزومات(نیازمندیهای سیستم)

- مصاحبه فردی با افراد آگاه از عملیات و طرح های سیستم فعلی و نیازهای فعالیت های سازمانی آتی
- نظر سنجی از افراد مطلع از طریق پرسش نامه برای کشف نظرات و ملزومات
- مصاحبه با گروه هایی از مردم با نیازهای گوناگون برای یافتن همخوانی ها و مغایرت های بین ملزومات سیستم
- مشاهده کارکنان در زمانهای انتخابی برای مشاهده داده هایی که بکار گرفته می شوند و اطلاعاتی که افراد برای انجام کارهایشان نیاز دارند.
- مطالعه اسناد شغلی (یا تجاری) برای بدست آوردن طرح های گزارش شده ، سیاست ها ، قوانین و...

یک نمونه راهنمای مصاحبه:

رئوس مطالب مصاحبه	
مصاحبه شونده :	
مصاحبه کننده :	
محل / وسیله :	تاریخ ملاقات:
ساعت شروع:	ساعت پایان :
اهداف : چه داده هایی جمع آوری می شود روی چه مواردی توافق می شود چه محیط هایی کاووش می شود	**یادآوری :** پس زمینه/تجربیات مصاحبه شونده نظریات شناخته شده مصاحبه شونده
دستور کار	**زمان تقریبی**
مقدمه	1 دقیقه
پس زمینه پروژه	2 دقیقه
مرور مصاحبه: موضوعاتی که باید پوشش داده شود اجازه برای ضبط	1 دقیقه
سوالات موضوع1	5 دقیقه
سوالات موضوع 2	7 دقیقه
خلاصه ای از نکات اصلی	2 دقیقه
سوالاتی از مصاحبه شونده	5 دقیقه
ختم	1 دقیقه
مشاهدات کلی مثلا:مصاحبه شونده مشغول بنظر می رسیده احتمالا نیاز باشد چند روز بعد سوالات دنبال شوند ...	
نظرات مطرح نشده،عناوین پوشش داده نشده(موضوعات پرسش نشده) مثلا:او نیاز به مشاهده اشکال فروش از 1996 داشت . او نظریاتی در مورد اداره کالاهای برگشتی داشت اما فرصت بحث نبود .	

مصاحبه	تاریخ
سوالات	ملاحظات

سوالات 1	پاسخ مشاهدات
سوالات 2	پاسخ مشاهدات

انتخاب سوالات مصاحبه

شاید لازم باشد تصمیم بگیرید که چه ترکیب و ترتیبی از سوالات Open-ended،close – end بکار ببرید.

سوالات مصاحبه و پرسشنامه

Open – end : سوالاتی در مصاحبه و پرسشنامه،که پاسخ از قبل مشخص شده ندارد و در مورد اطلاعاتی بکار میروند که نمی توانید همه پاسخ های ممکن را پیش بینی کنیدو یاوقتی که سوال دقیق و مشخص برای پرسیدن ندارید.

اشکال:طول کشیدن زمان پاسخ یا سختی خلاصه کردن پاسخ سوالات Open – end

Close-end : سوالاتی در مصاحبه و پرسشنامه که پاسخ آنها از بین مجموعه ای از پاسخ های مشخص تعیین می شود(مانند سوالات چهار جوابی)

مزایا : زمان زیادی لازم ندارد.

موضوعات بیشتری را پوشش می دهد.

عیب : اطلاعات مفید در پاسخ ها نمی گنجد

توزیع پرسشنامه ها

مصاحبه وقت گیر و پر هزینه است،درک عمیق می دهد و تعداد سوال محدودی پرسیده می شود و با افراد خاصی مصاحبه صورت می گیردولی پرسشنامه غیر فعال است درک کنترل از مصاحبه بدست میدهد ، خیلی گران نیست ، در زمان کمتر از تعداد افراد بیشتر می توان سوال کرد.

آنهایی که نمونه گیری شان راحت است(در دسترس اند ، تمایل به نظر سنجی دارند،آنهایی که انگیزه پاسخگویی بیشتری دارند)

مشاهده مستقیم کاربران:

با مشاهده کاربران در حین کار در زمانهای خاص می توان اطلاعات خوبی به دست آورد.

پس از مطالعه عملیات فعلی سازمان و سیستم های اطلاعاتی مورد استفاده و تعیین ملزومات و تشخیص خواسته کاربران از سیستم بایستی به مطالعه ملزومات و ساختار آنها مطابق با ارتباطات داخلی و حذف افزونگی ها)پرداخت.

مدل سازی

جهت به تصویر کشیدن سیستم،مدل سازی یک روش ساختار است که به کمک آن می توان سیستم را از بالا به پایین) از کل به جزء) تجزیه و تحلیل نمود.

مدل DFD

DFD (Data flow diagram) نمودار جریان داده:

ابزاری است که به شما اجازه می دهد چگونگی جریان داده را،در یک سیستم اطلاعاتی مدل سازی نمایید. از آنجایی که جریان داده روی جابجایی داده بین پردازش ها تمرکز دارد نمودار پردازش نیز نامیده می شود.

مدل سازی فرآیند (پردازش) :

دربردارنده نمایش گرافیکی عملکردها یا فرآیندهایی است که داده ها را ثبت، دستکاری و ذخیره نموده، بین یک سیستم و محیط آن و بین مولفه های درون سیستم توزیع می نماید، یک فرم معمول از یک مدل فرآیند DFD است که یکی از تکنیکهای تجزیه و تحلیل ساختار یافته می باشد.

در حین سا زماندهی ملزومات شما و اعضای تیم بایستی اطلاعات را به فرم معنی داری برای سیستم اطلاعاتی موجود وسیستم جایگزین سازماندهی نمایید. علاوه بر مدل سازی عناصر پردازش یک سیستم اطلاعاتی و چگونگی جریان داده ها در سیستم بایستی منطق پردازش و زمان وقوع در سیستم و نیز ساخ تار داده های سیستم را نیزمدلسازی نمایید.

مدل فیزیکی:

مدلی است که چگونگی انجام عملیات درون سیستم را نمایش می دهد .یعنی سیستم را به همان نحوی که واقعا رخ می دهد، در مکانی که به وق وع می پیوندد و وسیله یا اشخاصی که پردازشها را انجام می دهند نشان می دهید . این مدل به دنبال به تصویر کشیدن ساختار فیزیکی در درون سیستم می باشد.

مدل منطقی:

مدل ای است که به منطق آنچه در سیستم باید انجام شود می پردازد . این مدل مستقل از تکنولوژی بوده و به وسایلی که پردازشها را انجام می دهد یا ترتیب آنها کاری ندارد.

در تجزیه و تحلیل ساخ ار یافته ا ول ین نتیجه مدلسازی فرآیند مجموعه ای از نمودارهای جریان داده مرتبط و وابسته به هم می باشد .این نمودارها عبارتند از:

۱- نمودارمتن یا دیاگرام مفهومی Context Diagram

۲- DFD های فیزیکی سیستم فعلی

۳- DFD های منطقی سیستم فعلی

۴- DFD های منطقی سیستم جدید

۱- دیاگرام متن Context Diagram

هدف سیستم و عملکرد کلی آن را نشان می دهد و بیانگر این است که کدام عناصر داخل سیستم و کدام عناصر خارج از سیستم هستند.

در سیستم های بزرگ معمولا افراد نمی توانند اطلاعات دقیق برای ترسیم دیاگرام متن در اختیار تحلیل گر بگذارند .بهتر است ابتدا سیستم را به زیر سیستمهای کوچک تقسیم کرد و برای هر یک دیاگرام متن جدا در نظر گرفت و بعد این دیاگرامها را کنار هم گذاشته تا دیاگرام متن کل سیستم حاصل شود.

۲- نمودار های جریان داده فیزیکی سیستم فعلی

مشخص می کند که کدام افراد و فن آوریها در کدام فرآیندها برای جابجایی و انتقال داده، پذیرفتن ورودیها و تولید خروجیها استفاده شده اند. ورودیها با یستی با جزئیات کافی باشد تا:

- درک سیستم فعلی راحتتر گردد .
- چگونگی تبدیل سیستم جاری به سیستم جایگزین مشخص شود.

۳- نمودار جریان داده منطقی (مستقل از تکنولوژ ی) سیستم جاری

نشان می دهد که چه عملیات پردازش داده ای توسط سیستم اطلاعاتی فعلی به اجرا در می آید.

۴- نمودار جریان داده سیستم جا یگزین

جریان یا جابجایی داده ها ،ساختار نیازهای عملیاتی سیستم جدید در نمودارهای منطقی نمایش داده می شوند.

تعاریف و سمبل های مورد استفاده در ترسیم DFD

جریان داده Data Flow :

بعنوان داده در حرکت از یک محل به محل دیگر در سیستم بهتر می تواند شناخته شود . همچنین می تواند بیانگر نتایج یک پرس و جوی پایگاه داده ،محتوای یک گزارش چاپی یا داده روی فرم ورود داده باشد.

مثال :فرم سفارشی مشتری، یک چک حقوقی

یک جریان داده، یک داده در حال حرکت است، بنابراین می تواند ترکیبی از اجزای داده منحصر بفرد باشد که در یک زمان تولید شده و با یکدیگر به مقصد مشترکی جریان می یابند.

مخزن داده – انبار Data Store :

محل استقرار داده است ، می تواند یکی از چند محل فیزیکی ذخیره داده باشد. بعنوان مثال : یک پوشه فایل، یک یا چند فایل کامپیوتری، یا یک دفتر (بعدا در برنامه نویسی بعنوان پایگاه های داده مورد ستفاده قرار می گیرند.)

مثال :یک data store می تواند شامل اطلاعاتی راجع به مشتریان، دانشجویان، سفارش دهندگان ، فاکتورها و... باشد.

فرایند یا پردازش Process:

عملی است که روی داده انجام می شود تا تبدیل یابد، ذخیره شود یا توزیع گردد . هنگام مدلسازی پردازش داده ها در یک سیستم، مهم نیست که پردازش دستی باشد یا کامپیوتری.

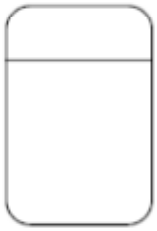
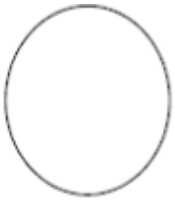
: Source / Sink

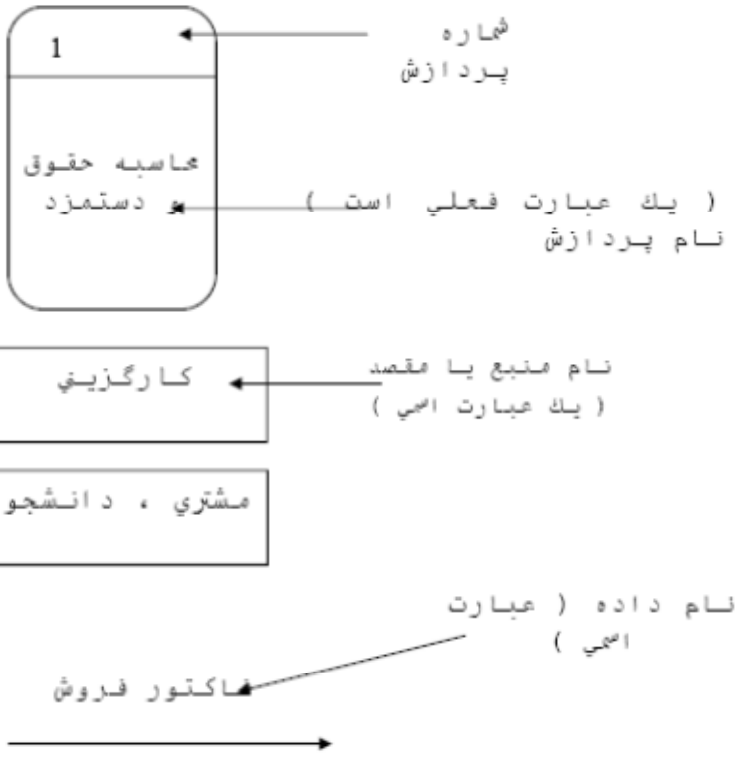
مبدا / مقصد داده هاست . بعضی وقتها به منابع موجودیت خارجی یا نهاد خارجی نیز اطلاق می گردد، چرا که خارج از سیستم هستند . وقتیکه پردازش صورت گرفت داده ها یا اطلاعات سیستم را ترک کرده به جای دیگر می روند . از آنجایی که مبدا و مقصد ها خارج از سیستم مورد مطالعه ما هستند مشخصه های آنها خیلی برای ما جالب توجه نیست زیرا :

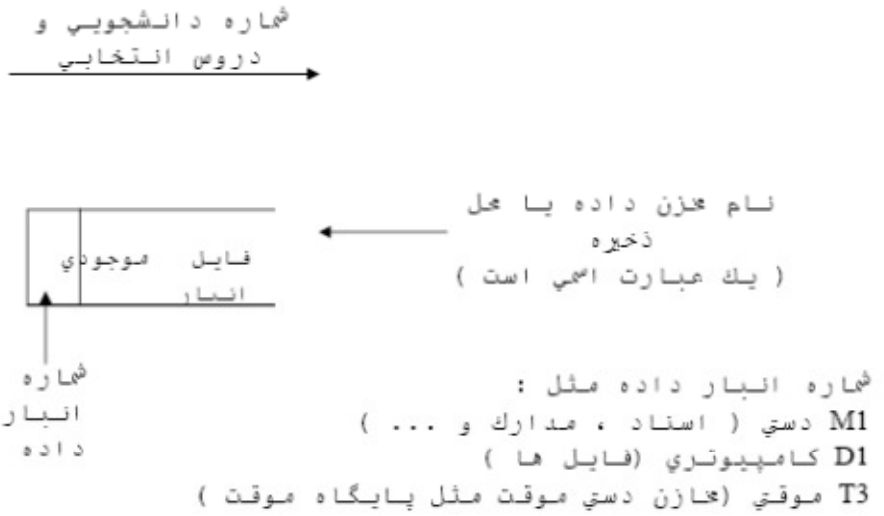
- فعل و انفعالاتی که بین مبدا و مقصد صورت می گیرد.
- مبدا و مقصد چه کاری با داده انجام می دهند؟ چگونه اینکار را می کنند؟
  (Source / Sink همانند یک جعبه سیاه هستند.)
- چگونگی کنترل یا طراحی مجدد مبدا و مقصد . (از نقطه نظر سیستم مورد مطالعه ما داده هایی که منبع می فرستد و داده هایی که مقصد دریافت می کند ثابت هستند)
- چگونگی دسترسی مستقیم مبدا / مقصد به داده ها

سمبل های مورد استفاده در سیستم DFD

دو مجموعه استاندارد مختلف برای ترسیم سمبل های DFD دارد دارد ولی هر دو از چهار سمبل برای نمایش Data store، Data flow ، Source / Sink ،progress استفاده می کنند.در ادامه بحث ما از سمبل های Gane &Sarsone استفاده می کنیم.

سمبل های gane & sarson		سمبل های demacro&yourdon
	پردازش progress	
	انبار داده data store	
	منبع / مقصد Source/sink	
	جریان داده Data flow	





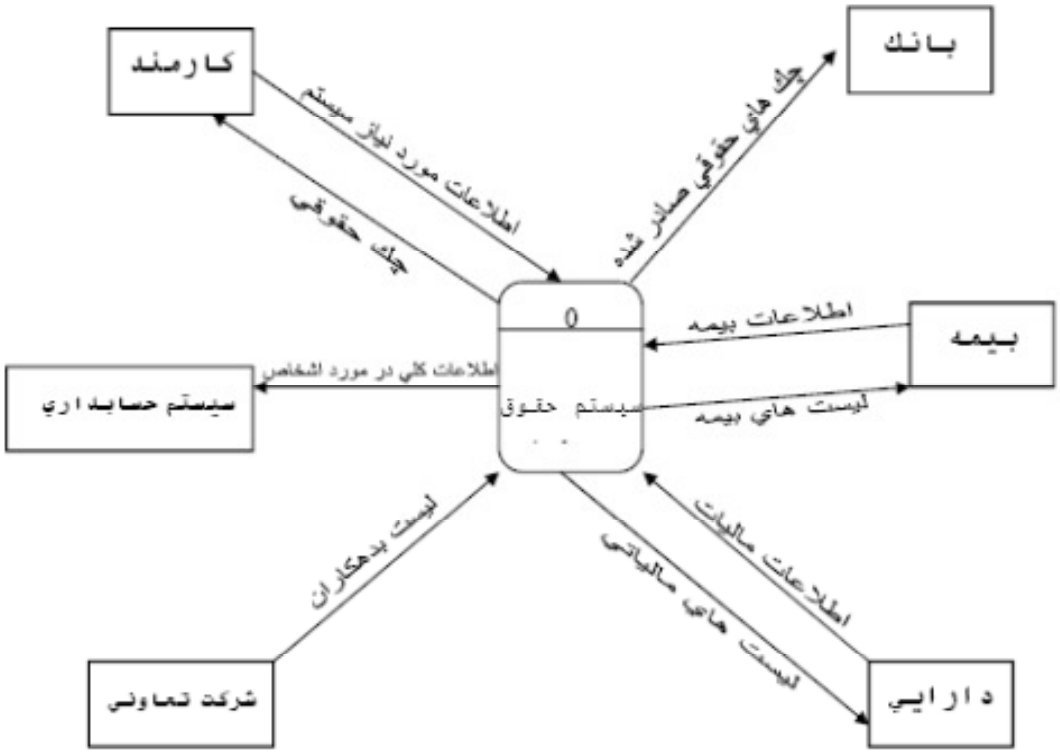
البته در سایر کتابها پردازش را با سمبل های دیگری نظیر:

و **منبع** (مبدا / مقصد) را هم با علامت **بیضی** نمایش می داده اند.

همانطور که گفتیم مبدا ه ا و مقصد ها همیشه خارج از سیستم اطلاعاتی هستند و مرزهای سیستم را مشخص می کنند.

داده ها از از یک یا چند منبع خارج از سیستم نشات می گیرد و سیستم بایستی اطلاعات را برای یک یا چند مقصد تولید نماید . هر پردازش داده ای که درون مبدا / مقصد صورت بگیرد، مورد توجه ما نمی باشد ، چرا که این پردازش خارج از سیستم هایی که ما به تصویر کشیده ایم صورت گرفته است.

مثالی از یک Context Diagram :



یک مبدا / مقصد می تواند یکی از موارد زیر باشد:

- یک سازمان دیگر با یک واحد سازمانی دیگر که داده ها را به سیستم مورد نظر شما می فرستد ، یا دریافت می کند(به عنوان مثال یک تولید کننده، به عبارت دیگر این سازمان ó یا یک بخش دانشگاهی خارج از سیستم مورد مطالعه ماست).
- یک شخص داخل یا خارج از واحد تجاری تحت حمایت سیستمی که شما تحلیل می کند و کسی که با سازمان ارتباط متقابل دارد (مثلا یک مشتری ی ا متصدی وام)
- یک سیستم اطلاعاتی دیگر که سازمان مورد مطالعه شما با آن تبادل اطلاعات دارد.

نکات دیگر

۱-در سیستم های بزرگ شاید اطلاعات کافی برای ترسیم دیاگرام متن کلی در اختیار تحلیل گر قرار نگیرد پس بهتر است سیستم را به زیر سیستم های کوچکتر تقسیم و دیاگرام متن هر زیر سیستم را ترسیم نموده و سپس این دیاگرام ها را کنار هم گذاشته تا دیاگرام متنی کل حاصل شود.

۲-تصحیح وظایف : اگر خطوط داده بین دو زیر سیستم زیاد است،شاید با جابجایی و انتقال یکی از وظایف از زیر سیستمی به زیر سیستم دیگر ارتباطات کمتر شود.

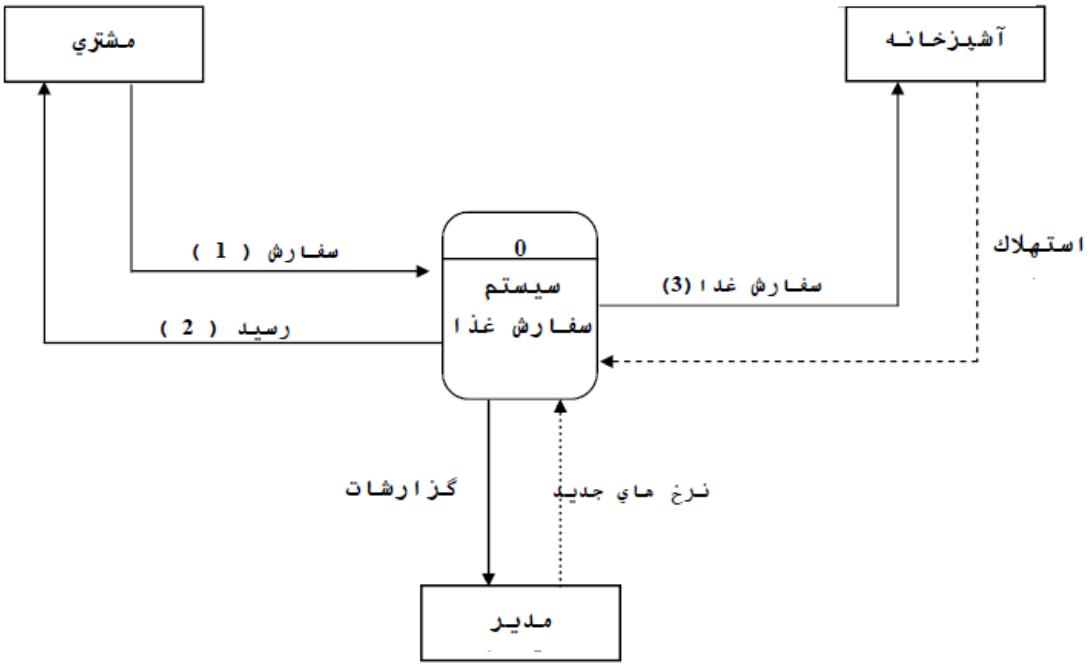
۳-CD باید حتی الامکان ساده، گویا و قابل فهم باشد:

- موجودیت خارجی :حداکثر ۸ موجودیت خارجی ،دسته بندی موجودیتها ، فرم شرح موجودیت ها
- جریان داده : حداکثر ۴ جریان داده برای هرموجودیت ،دسته بندی جریانهای داده فرم شرح خطوط جریان داده

تهیه DFD ها با یک مثال عملی

فرض کنید مدل منطقی یک سیستم سفارش غذا(برگرایندیانا)مد نظر است . بالاترین سطح مشاهده این سیستم، دیاگرام متن (Context Diagram) است که در شکل زیر نشان داده شده است . توجه داشته باشید که این دیاگرام شامل فقط یک پردازش تک با برچسب ۰ برای بیان عملکرد کلی سیستم، بدون محل ذخیره داده و دارای چهار جریان داده و سه مبدا / مقصد است.

در شکل زیر جریانات داده نقطه چین پیشنهاد دانشجویان بوده و در شکل های بعدی ترسیم نشده است:



تعریف Context Diagram :

یک نگرش کلی عملکرد سیستم سازمانی که مرزهای سیستم، موجودیت های خارجی که با سیستم ارتباط متقابل دارند و اکثر جریانهای اطلاعاتی بین این موجودیتهای سیستم را نشان می دهد.

برای ترسیم Context Diagram می توان ابتدا Document flow diagram را که نشان دهنده جریان های داده، موجودیت خارجی و مبدا یا مقصد بودن موجودیت می باشد رسم کرد.

Data flow	Source / Sink	Entity
سفارش مشتري	مبدا	مشتري
رسيد	مقصد	مشتري
سفارش غذا	مقصد	آشپزخانه
گزارشات	مقصد	مدير رستوران

مرحله بعد برای تجزیه تحلیل ،فکر کردن در مورد اینکه این پردازش تک در CD به چه پردازش هایی در سطح بعد تفکیک می شود، می باشد.

یعنی در این مرحله باید پردازشهای اصلی بیانگر عملیات سیستم م شخص گردد(اغلب این عملکرد های اصلی منطبق با انتخاب فعالیتها از منوی اصلی سیستم می باشند.)در مورد مثال قبلی این عملیات عبارتند از:

۱-گرفتن داده ها از منابع متفاوت

۲-تولید و توزیع داده به مقاصد مختلف

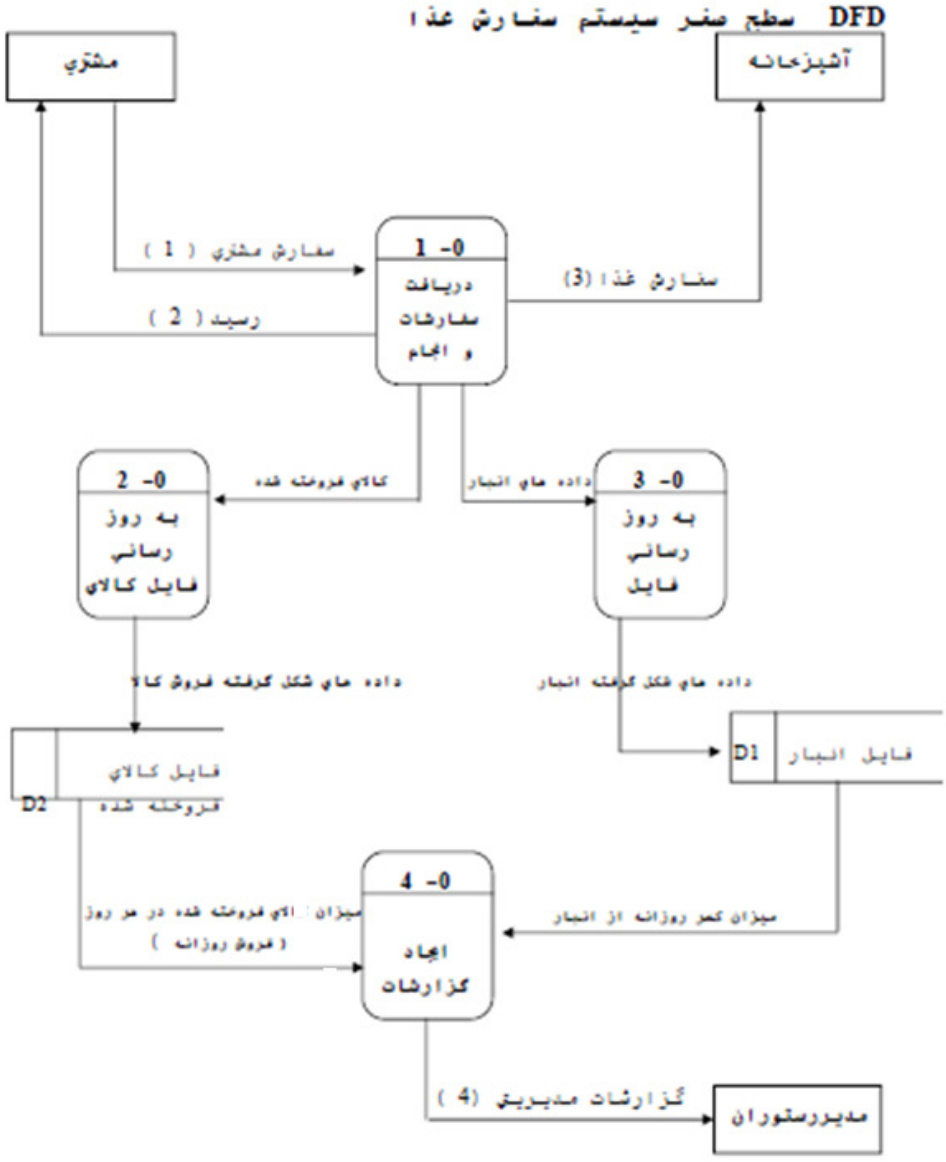
۳-به روز رسانی داده ها

۴-توصیف سطح بالایی از عملیات تبدیل داده

دیاگرام سطح صفر (DFD Level):

یک دیاگرام جریان داده که بیانگر پردازشهای اصلی سیستم، جریان ات داده و مخازن داده در سطح بالایی از جزئیات می باشد.

در مث ال قبلی DFD سطح صفر سیستم سفارش غذا به این صورت می باشد:

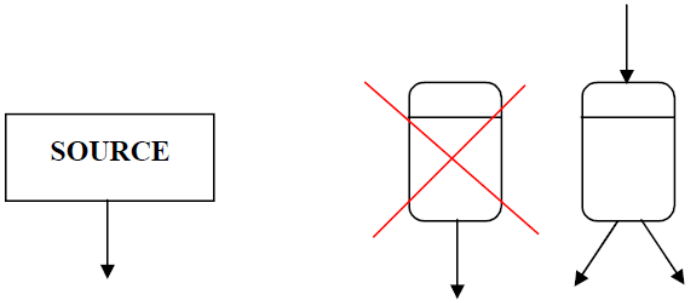


قوانین ترسیم نمودار جریان داده:

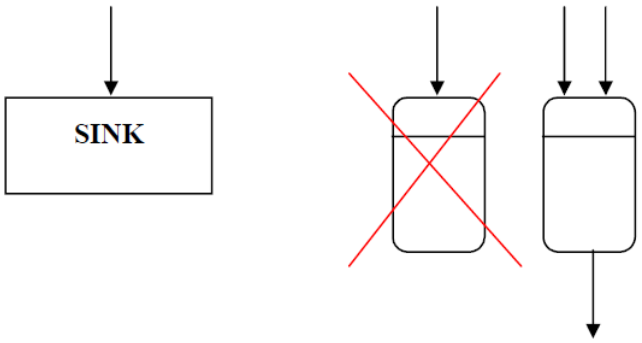
پردازش

A هیچ پردازشی نمی تواند فقط خروجی داشته باشد(ایجاد داده از هیچ،یک معجزه است)

اگر یک شی فقط خروجی داشت بایستی یک منبع باشد



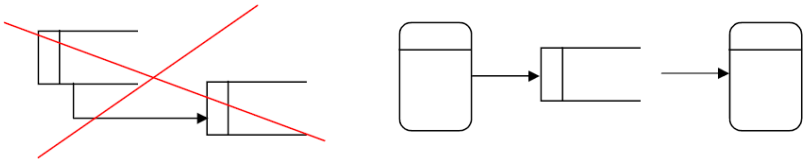
B . هیچ پردازشی نمی تواند فقط ورودی داشته باشد . اگر یک شی فقط ورودی داشت یک مقصد است.



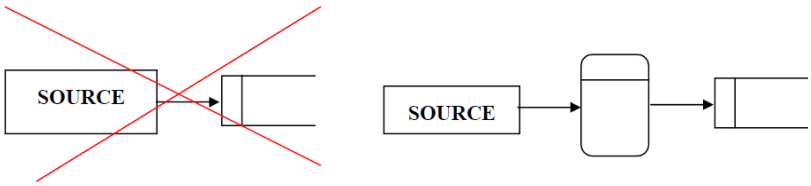
C . یک پردازش برچسب عبارت فعلی دارد.

مخزن داده

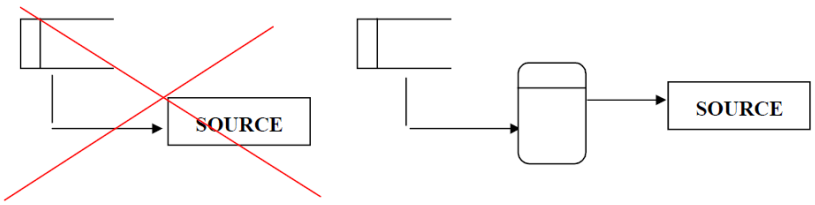
D . داده نمی تواند مستقیما بین مخازن داده جابجا شو د. داده بایستی بین پردازش ها جابجا شود.



E . داده نمی تواند مستقیما از یک منبع خارجی به مخزن داده جابجا شود . داده بایستی از منبع به پردازشی که داده را دریافت می نماید جابجا شود و در مخزن داده قرار گیرد.



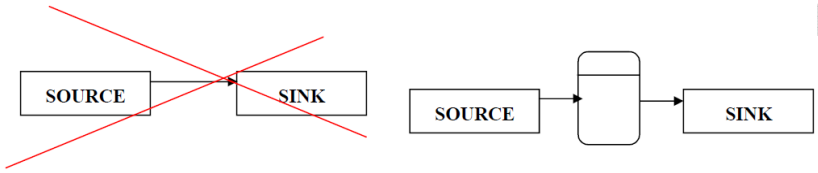
F. داده نمیتواند مستقیما بین مخزن داده و مقصد جابجا شود،این کار بایستی توسط پردازش صورت بگیرد.



G. یک مخزن داده برچسب عبارت اسمی دارد.

مبدا / مقصد

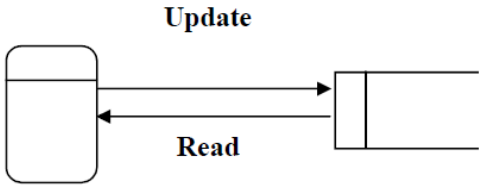
H. داده نمیتواند مستقیما بین مبدا و مقصد جابجا شود، هر داده مرتبط با سیستم م ا بایستی توسط پردازش جابجا شود.جریان داده های غیر مرتبط با سیستم در DFD نمایش داده نمی شوند.



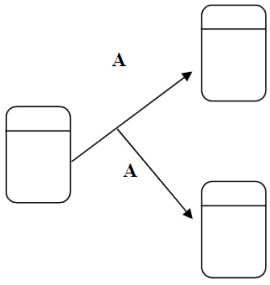
I. هر مبدا /مقصد یک برچسب عبارت اسمی دارد.

جریان داده

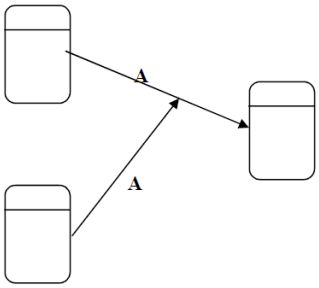
J . جریان داد ه فقط یک جهت دارد .داده می تواند در دو جهت بین پردازش و یک مخزن داده جریان یابد تا عمل خواندن قبل از به روز رسانی را نشان دهد .اگر چه این کار با دو فلش مجزا که در زمانهای متفاوت اتفاق افتاده نشان داده می شود.



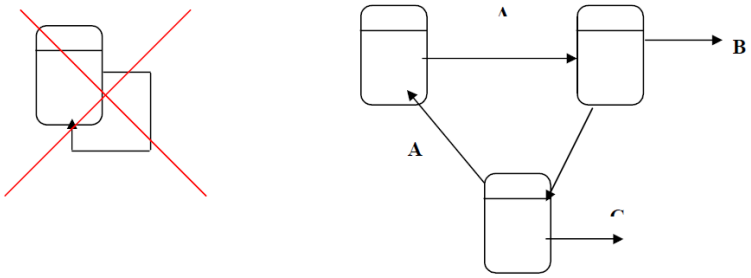
K . یک انشعاب از جریان داده به این معنی است که جریان داده یکسانی از یک محل مشترک به دو یا چند پردازش متفاوت یا مخزن داده رفته است(معمولا بدین معنی است که کپی های مختلفی از داده یکسان به محلهای مختلف میروند.)



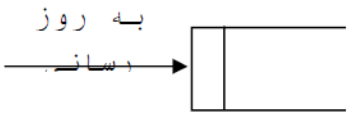
L . یک اتصال در جریان داده بدین معنی است که داده یکسانی از یک یا دو یا تعداد بیشتری از پردازش ها، مخازن داده یا مبدا /مقصد به محل مشترکی آمده است.



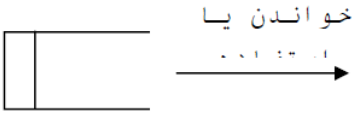
M . جریان داده نمی تواند مستقیما به همان پردازش که از آن خارج شده برگشت نماید . حداقل بایستی یک پردازش دیگر باشد که جریان داده را اداره نماید، جریان دیگری تولید و جریان داده اصلی را به پردازش آغازین برگرداند.



N . جریان داده به یک مخزن داده به معنی به روز رسانی است(حذف یا تغییر)



O . جریان داده از یک مخزن داده به معنی بازیابی یا استفاده است.



P . یک جریان داده برچسب عبارت اسمی دارد . در مدتی کههمه جریانات داده روی یک فلش با هم بعنوان یک بسته جابجا می شوند، این امکان وجود دارد که بیشتر از یک عبارت اسمی روی جریان داده ظاهر شود.

علاوه بر قوانین فوق دو قاعده کلی دیگری برایDFDوجود دارد که اکثر اوقات بکار می رود:

- **ورودی پردازش با خروجی آن متفاوت است :**

بدلیل اینکه آن پردازش یک هدف دارد، مثلا بجای اینکه بطور ساده داده را بدون دستکاری عبور دهد، ورودی ها را تبدیل به خروجی ها می نماید. آنچه اتفاق می افتد اینست که همان ورودی به پردازش وارد و از آن خارج می شود اما پردازش جریانات داده دیگری نیز تولید می کند که نتیجه کار روی ورودی هاست

اشیاء درDFD اسامی منحصر به فرد دارند : هر پردازش یک نام یکتا دارد . دلیلی برای داشتن دو پردازش با نام یکسان وجود ندارد . گرچه برای حفظ نظم و ترتیب **DFD** شما ممکن است مخازن داده و مبدا/ مقصدها را تکرار نماید . وقتیکه د و جریان داده اسم یکسانی دارند، باید دقت نمایید که این دو جریان واقعا

یکسان هستند . نام جریان داده بیانگر یک مجموعه خاص از داده هاست و جریان داده دیگری با حتی یک بخش کمتر یا یک بخش بیشتر داده بایستی نام یک تای مختلف داشته باشند.

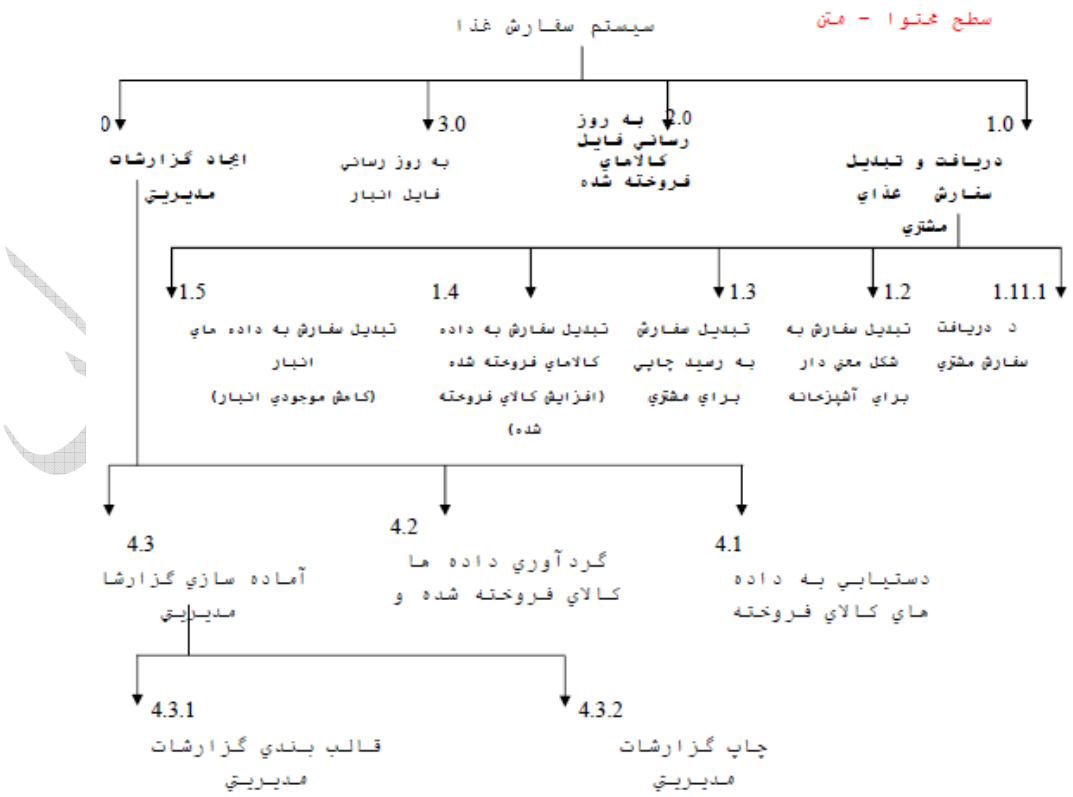
تجزیه DFD

در مثال قبل سیستم سفارش غذا با یک دیاگرام متن سطح بالا شروع کردیم . با فکر کردن بیشتر در مورد سیستم دیدیم که سیستم بزرگتر شامل چهار پروسه می باشد .عمل رفتن از یک سیستم تک به چهار مولفه پردازش ،تجزیه عملیاتی نامیده می شود،در سیستم سفارش غذا سیستم بزرگتر به چهار پروسه تجزیه شد . هر یک از آن فرآیند ها(زیر سیستمها)هم برای تجزیه کاندیدا هستند . هر فرآیند ممکن است شامل چندین زیر سیستم باشد . و هر زیر سیستم نیز ممکن است به واحدهای کوچکتر تقسیم شود . عمل تجزیه تا زمانی ادامه می یابد که به نقطه ای برسید که هیچ زیر سیستمی نتواند بطور منطقی به زیر سیستم های بعدی تقسیم شود.

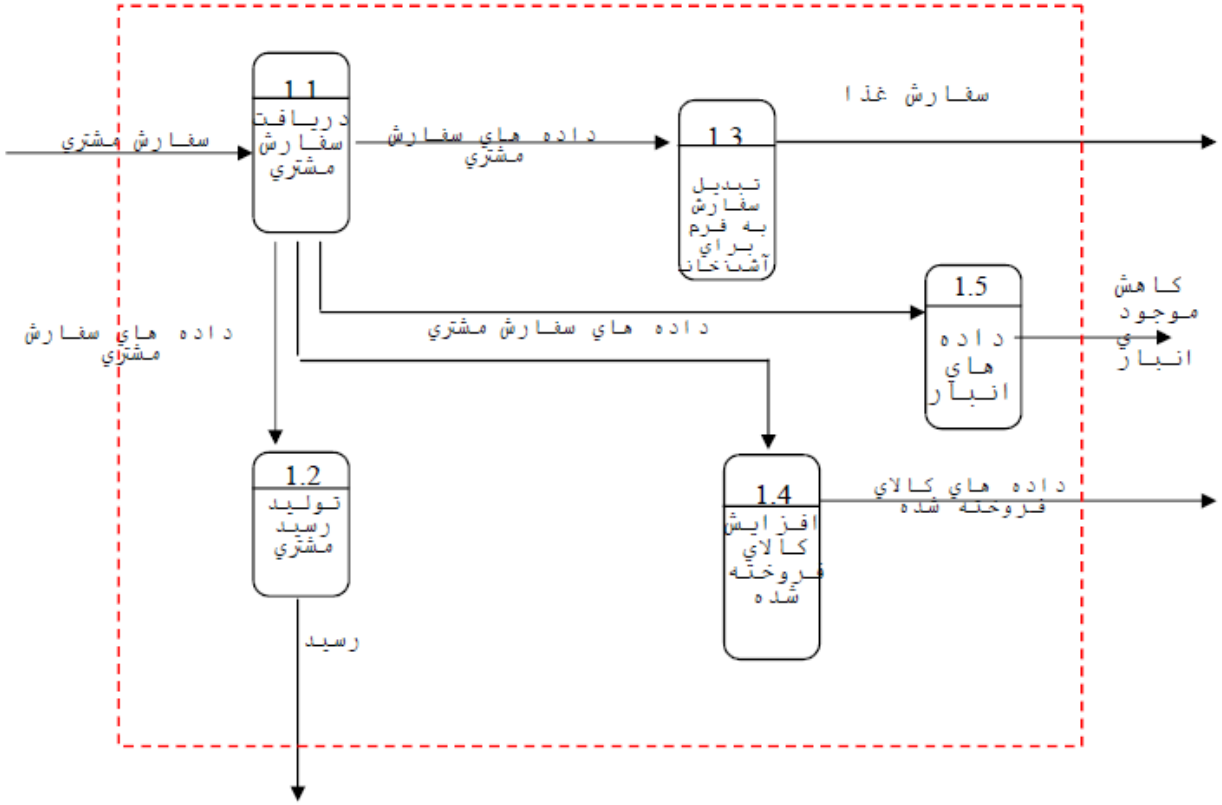
پایین ترین سطح DFD،DFD اولیه (Primitive DFD) نامیده می شود.

تجزیه عملیاتی – Functional Decomposition

یک فرایند تکراری تقسیم توصیف یا پرسپکتیو سیستم به جزئیات کوچکتروکوچکتر است،که مجموعه ای از چارت ها را تولید می نماید،که یک فرایند در یک چارت مشخص با جزییات بیشتری در چارت دیگر توضیح داده می شود. بیایید با مثال سیستم سفارش غذا ادا مه ب دهیم تا ببینیم چطور یک DFD سطح صفر می تواند به DFD بعدی تجزیه شود:



تجزیه پروسه ۱,۰ به DFD های دیگر(دیاگرام سطح۱ - پروسه ۱,۰ سیستم سفارش غذا)در شکل زیر نشان داده شده است:



توجه کنید که هر یک از ۵ پروسه شکل فوق به عنوان زیر پروسه های پروسه ۱,۰ ، پروسه ۱,۱ و پروسه ۱,۲ و نظایر آن نامگذاری شده اند . همچنین توجه نمایید که هریک از پروسه ها و جریانات آن همانند سایر DFD ها نامگذاری شده اند.

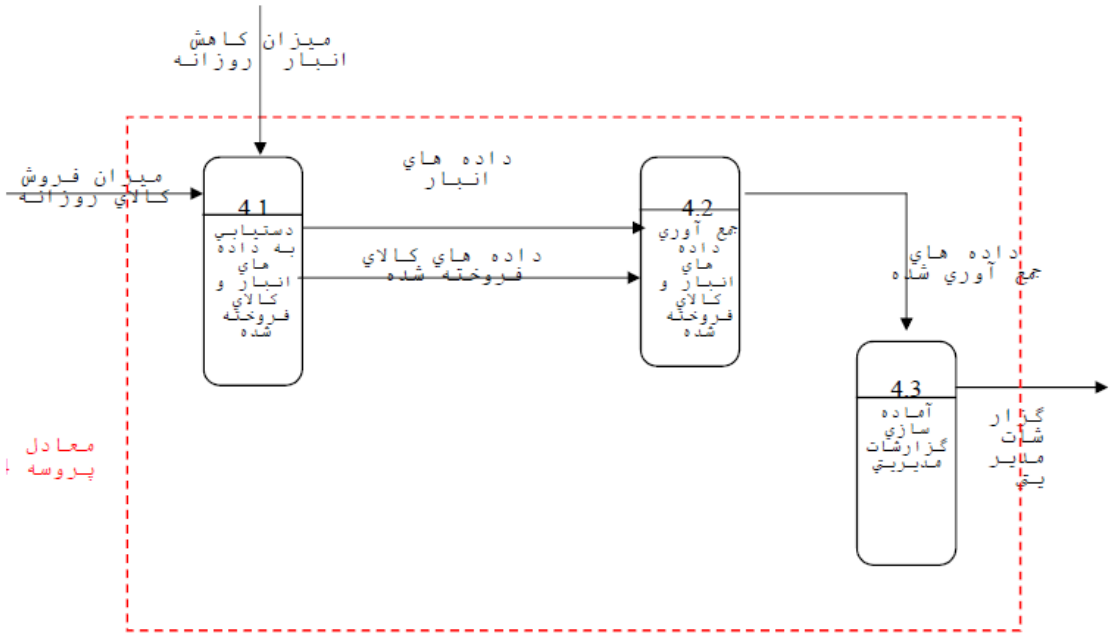
همچنین توجه خواهید کر د که هیچ مبدا یا مقصدی نمایش داده نشده است .اگرچه آوردن مبدا ها و مقصدها ممکن است.

نمودار جریان داده در شکل فوق دیاگرام سطح ۱ نامیده می شود اگر ما تصمیم به تجزیه پروسه های ۲,۵ ، ۳,۵ یا ۴,۵ ، بگیریم DFD های حاصل نیز DFD سطح ۱ خواهند بود .

معمولا یک دیاگرا م سطح n ، DFD ای است که از n تجزیه تودرتوی سطح ۱ ایجاد می شود.

پروسه های ۲,۵ ،۳,۵ عملیات مشابهی انجام می دهند که در آن هر دو داده ورودی را برای به روز رسانی مخزن های داده استفاده می کنند . از آنجا که به روز رسانی مخازن داده یک عملکرد منطقی تک است هیچ یک از این پروسه ها نیاز به تجزیه بعدی ندارد . به عبارت دیگر می توانیم پروسه ۴,۰ تولید گزارشات مدیریتی را به

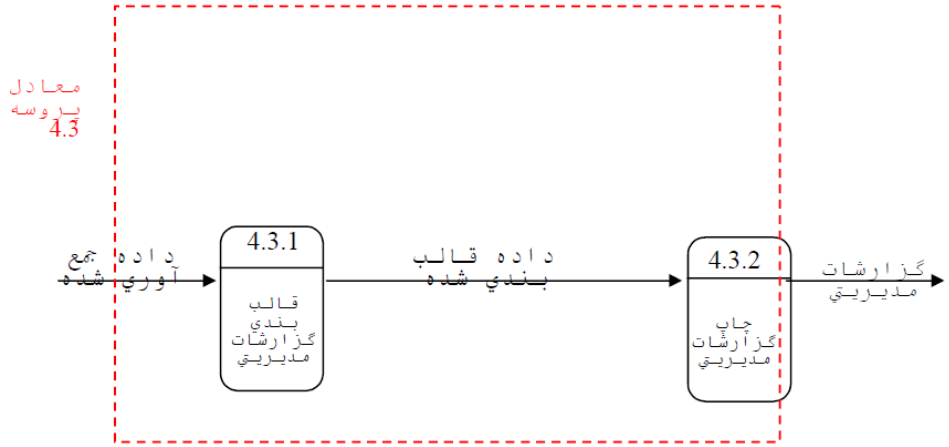
حداقل ۳ زیر پروسه تجزیه کنیم . دستیابی به کالاهای فروخته شده و داده های انبار،جمع آوری فروش کالا و داده های انبار و آماده سازی گزارشات مدیریتی تجزیه پروسه ۴,۰ در دیاگرام سطح۱ نشان داده شده است:



هر DFD سطح n نشان دهنده یک پروسه از DFD سطح ۱ – n است.

هر DFD می تواند در یک صفحه مجزا باشد . به عنوان یک قاعده کلی هیچ DFD ای نباید بیشتر از ۷ پروسه داشته باشد .چرا که پروسه زیاد دیاگرام را خیلی شلوغ و درک آن را مشکل تر می سازد.

DFD Level ۲ Process ۴



متعادل کردن DFD (Balancing)

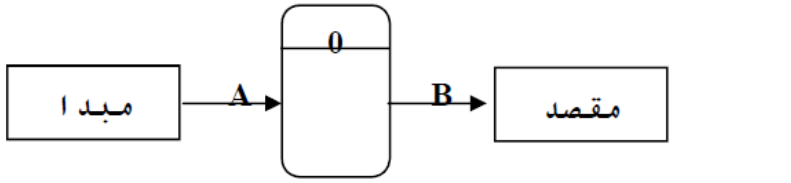
وقتیکه DFD را از سطحی به سطح دیگر تجزیه می کنید،یک اصل حفاظت و نگهداری وجود دارد . شما بایستی ورودی ها وخروجی های یک پروسه را در سطح بعدی تجزیه حفظ نمایید.

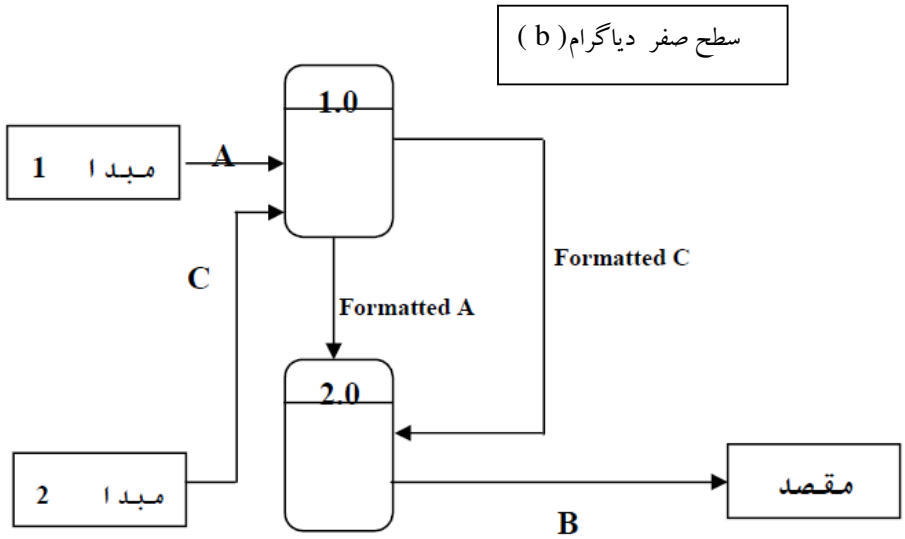
به عبارت دیگر پروسه ۱,۰ که دیاگرام سطح ۰ ظاهر می شود،بایستی هنگام تجزیه به دیاگرام سطح ۱ همان ورودی ها و خروجی ها را داشته باشد .

این حفظ ورودی و خروجی متعادل کردن (balancing) نامیده می شود.

شکل زیر مثالی از یک DFD نامتعادل را نشان می دهد . در اینجا دیاگرام متن دارای یک ورودی (A) و یک خروجی (B) می باشد.در حالیکه در دیاگرام سطح ۰ ورودی اضافی (C) وجود دارد و جریانات A و C از منابع مختلف می آید .این دو DFD متعادل نیستند:

(a) دیاگرام محتوا



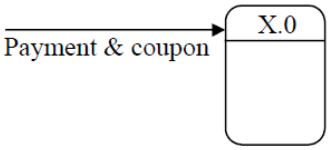


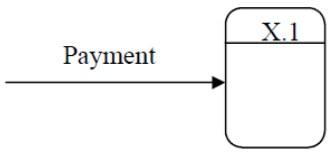
یک جریان داده شامل چند زیر جریان در دیاگرام سطح n می تواند در دیاگرام سطح ۱+ n برای پروسه ای که این جریان داده مرکب را به عنوان ورودی می پذیرد، تقسیم شود.

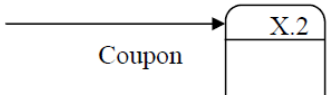
شکل مقابل مثالی از تقسیم جریان داده را نشان می دهد:

(a) جریان داده ترکیبی

(b) جریان داده تفکیک شده







مفهوم حفظ تعادل و هدف نگهداری هر چه ساده تر DFD منجر به چهار قانون پیشرفته اضافه برای ترسیم DFD می گردد .قوانین پیشرفته درزیر آمده اند.

وضع قوانین پیشرفته برای ترسیم DFD

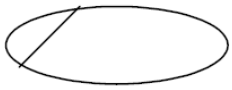
Q. یک جریان داده مرکب از یک سطح می تواند در سطح بعدی به چند مولفه جریان داده تقسیم می شود . اما داده جدید نمی تواند اضافه شود.

R. ورودیهای یک پروسه با در نظر گرفتن داده های موجود در مخازن داده بایستی برای تولید خروجی های پروسه کافی باشند . (ورودیهای اضافه نداشته باشیم که داده های آن بلا استفاده بمانند).

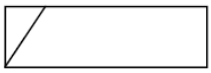
S. در DFD های پایین ترین سطح ممکن است جریان های داده جدید برای نمایش داده هایی که تحت شرایط استثنایی منتقل شده اند اضافه می شوند . این جریانهای داده بیانگر پیامهای خطا(نظیر مشتر ی ناشناخته، آیا می خواهید مشتری جدیدی ایجاد کنید؟ (یا نکات قابل توجه) مثل آیا می خواهید این رکورد را حذف کنید؟) می باشد.

T. برای اجتناب از داشتن جریان های داده متقاطع، ممکن است مخازن داده یا مبدا / مقصد یک DFD را تکرار کنید. برای تعیین یک سمبل تکراری ا ز یک سمبل اضافه نظیر یک خط دوتایی (=) در وسط خط عمودی سمبل مخزن داده یا یک خط قطری در گوشه مربع مبدا / مقصد استفاده کنید.

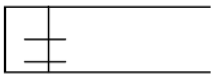
Source / sink



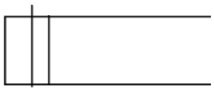
Source / sink



Data store



یا



چهار نوع مختلف DFD

در واقع چهار نوع مختلف DFD در فرآیند توسعه سیستم وجود دارد:

۱-فیزیکی فعلی -۲ منطقی فعلی -۳ منطقی جدید ۴-فیزیکی جدید

هنگامی که در اواخر سال ۱۹۷۵ تجزیه و تحلیل ساختار **یافته** معرفی شد، مورد توافق قرار گرفت که همهٔ چهار نوع DFD به این منظور خاص فراهم شود.

در DFD فیزیکی فعلی، پروسه ها برچسبی شامل نام افراد یا محل آنها یا نام سیستم های کامپیوتر ی که باید کل پروسه های سیستمی را فراهم نمایند، دارند . یعنی برچسب شامل تعریف فن آوری استفاده شده در پردازش

داده ها است .بطور مشابه جریانات داده و مخازن داده اغلب با نام رسانهٔ فیزیکی واقعی که داده روی آنها جریان دارد یا در آن ذخیره می شود نظیر یک پوشهٔ فایل، فایل های
کامپیوتری، فرم های تجاری یا نوارهای کامپیوتری، نام گذاری می گردند.

برای مدل منطقی فعلی، ویژگیهای فیزیکی سیستم حتی الامکان برداشته می شود . آنطورکه سیستم فعلی کاهش پیدا کند تا به اصل برسد ، یعنی به داده ها و پردازش هایی که صرفنظر از اشکال فیزیکی واقع ی آن ها ،داده ها را تبدیل می نمای ند .اگر کاربر کاملا از عملکرد سیستم فعلی راضی باشد، مدل منطقی جدید دقیقاً مشابه مدل منطقی فعلی است،اما در پیاده سازی مشکلاتی دارد . بطور نمونه مدل منطقی جدیدی را تصور کنید که با داشتن عملکردهای جدید با مدل منطقی فعلی متفاوت باشد . عملکردهای از کار افتاده برداشته شده و جریانات ناکارآمد مجدداً سازماندهی شده است.

سرانجام، DFD های فیزیکی سیستم جدید بیانگر پیاده سازی فیزیکی سیستم جدید می باشند DFD های سیستم فیزیکی جدید تصمیمات تجزیه و تحلیل در مورد اینکه کدام یک از عملکردهای سیستم، شامل آنهایی که در مدل منطقی جدید اضافه شده اند، ماشینی و کدامیک دستی خواهند بود، را منعکس می کنند.

برای روشن ساختن تفاوتهای انواع DFD ها با به مثال دیگر از برگرایندیانا می پردازیم.

دیدید که سیستم غذا دو نوع دادهٔ مصرفی برای کالاهای فروخته شده و انبار فراهم می سازد.

در پایان هر روز مدیر (باب ملان کمپ)گزارش انبار را می گیرد ،که مشخص می نماید، چه مقدار از کالای انبار برای هر قلم فروخته شده مصرف شده است . مقادیر نشان داده شده در گزارش انبار در اصل یک ورودی برای سیستم کنترل انبار دستی وسیع ی است که باب هر روزه استف اده می نماید.

مراحل درگیر در سیستم کنترل انبار باب عبارتند از:

۱-ملاقات با کامیونهای حمل قبل از باز کردن رستوران

۲-تخلیه و ذخیرهٔ محموله ها

۳-ثبت فاکتورها و دخیره در فایل کشویی

۴-اضافه کردن دستی مقادیر رسیده به گزارشات انبار

۵- پس از بستن رستوران، چاپ گزارش انبار

۶-شمارش مقادیر فیزیکی انبار

۷-مقایسهٔ حاصل جمع گزارش انبار با حاصل جمع شمارش فیزیکی

۸-مقایسهٔ حاصل جمع شمارش فیزیکی با حداقل مقادیر سفارش،اگر این مقدار کمتر بود سفارش بده و گرنه، هیچ کاری انجام نده.

۹-پرداخت صورتحساب هایی که موعدشان سررسیده و ثبت آنها به عنوان پرداخت شده نمودارهای جریان داده ای که سیستم فیزیکی فعلی را مدلسازی می نمایند در صفحه بعد نشان داده شده است.

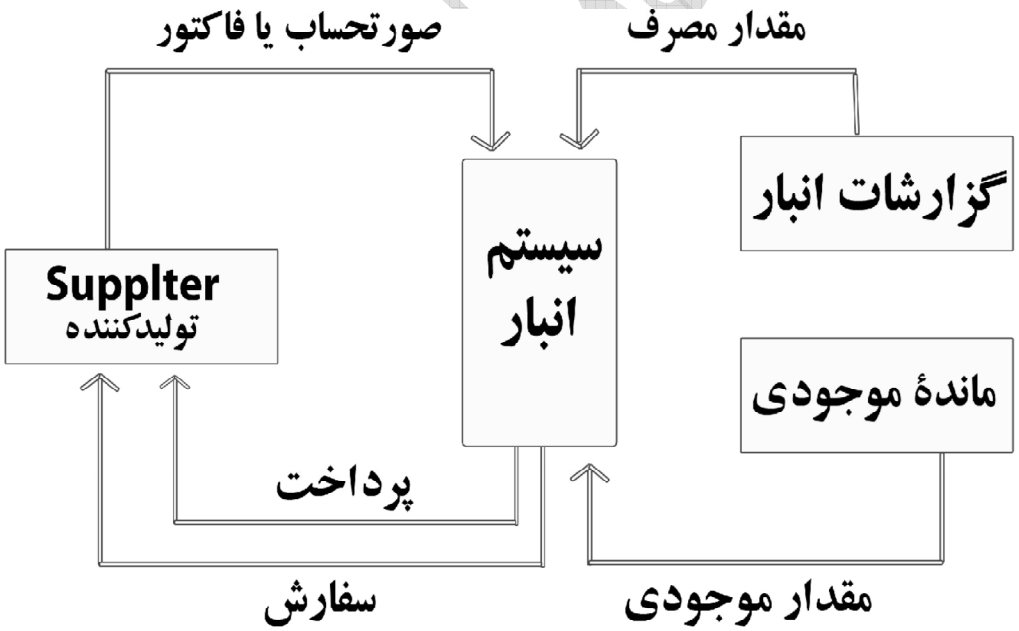
دیاگرام متن نشان داده شده سه منبع دادهٔ خارج از سیستم را نشان می دهد:تولید کنندگان، گزارش انبار سیستم سفارش غذا و ماندهٔ موجودی .تولید کننده ها فاکتورها را به عنوان ورودی فراهم می نمایند و سیستم، پرداخت ها و سفارشات را به عنوان خروجی به تولید کننده ها بر می گرداند.

هر دو مؤلفه گزارش انبار و موجودی ،مقادیر انبار را به عنوان ورودی سیستم فراهم می نمایند.

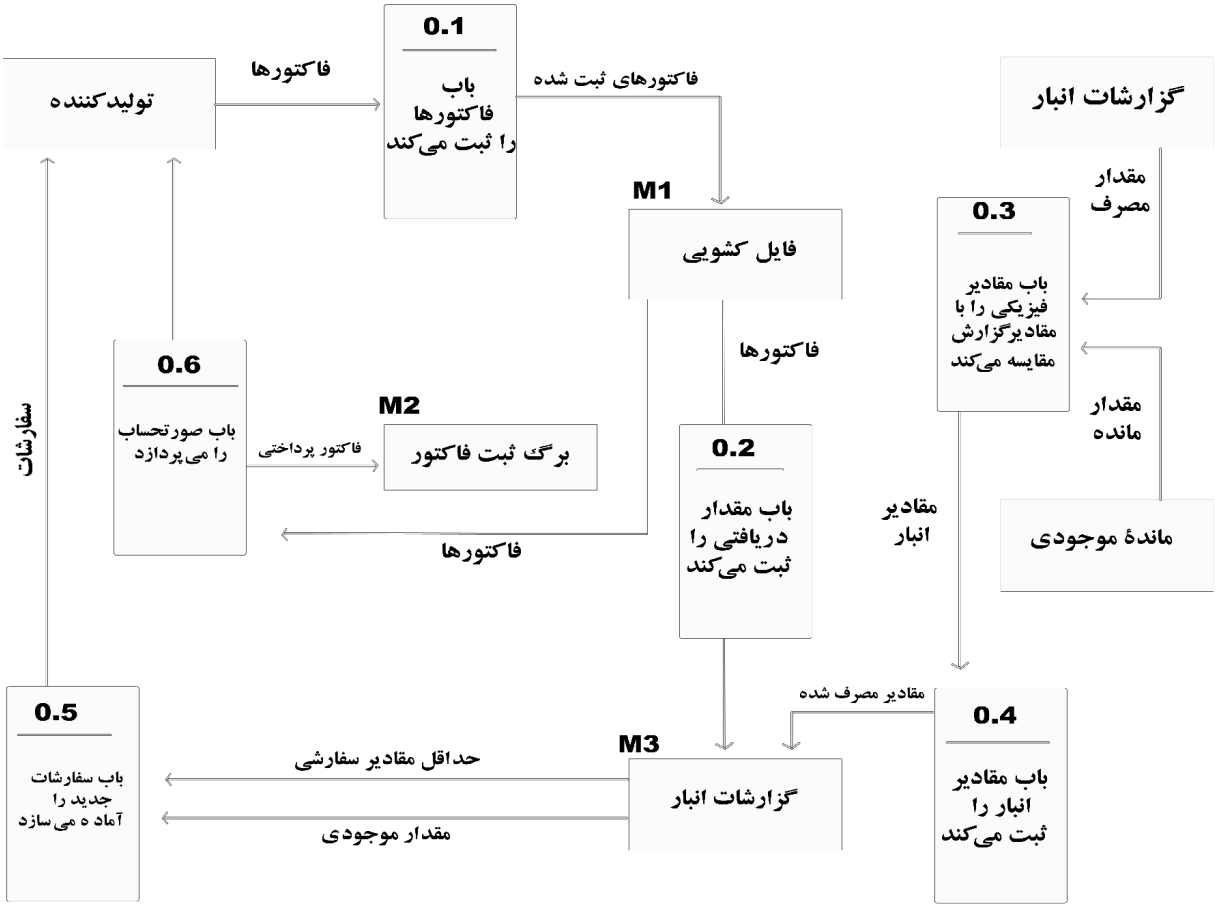
DFD سطح صفر برای سیستم انبار،شش پروسهٔ مختلف را نشان می دهد که اکثر آنها در لیست فعالیت های فهرست شده قبلی نیز ظاهر شده اند .شما می توانید از دیاگرام دریابیدکه وقتی باب فاکتورها را از تولید کننده ها دریافت می نماید، رسید آنها را روی یک برگهٔ ثبت فاکتور ثبت می نماید و فاکتورهای واقعی را در فایل کشویی نگهداری می نماید .با استفاده از فاکتورها باب مقدار

تحویل داده شده به انبار را از گزارشات انبارها که فرم های کاغذی گذاشته شده نزدیک محل ذخیره هر کالای انباراست، ثبت می نماید.

سیستم فیزیکی فعلی کنترل انبار (a) دیاگرام متن(محتوا)



(b) دیاگرام جریان داده سطح صفر (Level -۰)



باب مقادیر تحویل ی راهنگام تحویل روی برگهٔ ثبت می نماید، مقادیر مصرفی وقتیکه باب مقادیر استفاده شده از انبار را مطابق با مقدار فیزیکی و تعداد بدست آمده از گزارش انبار توسط سیستم سفارشی غذا بدست می آورد، وارد می شوند .توجه دا ریم که برگر ایندیانا برای بعضی کالاهای
فاسد شدنی که هر روز استفاده می شوند، نظیر نان، گوشت، سبزیجات؛ سفارش حمل روزانه دارد.
همانگونه که از DFD شکل **(b)** مشاهده می شود باب ازحداقل مقادیر سفارش و مقدار ماندهٔ موجودی برای تعیین اینکه چه سفارش هایی بایستی داده شوند، استفاده می نماید.
و از فاکتورها برای تعیین اینکه کدام صورت حساب ها بایستی پرداخت گردند استفاده می کند و به دقت هر پرداخت را یادداشت می کند.
در ادامه، برای ایجاد مدل منطقی فعلی با استفاده ازDFD ها، باید اساس سیستم انباری که باب وضع کرده را بشناسیم .داده های کلیدی ضروری برای دنبال کردن انبار و پرداخت صورت حساب چه هستند؟ پروسه های کلیدی درگیرکدامند؟ همچنین نیاز است که همهٔ عناصر سیستم فیزیکی، نظیر باب و پوشه های فایل فیزیکی که وی استفاده کرده را حذف نماییم . حداقل چهار پروسهٔ کلیدی وجود دارد که سیم انبار برگرانیدیانا را می سازد:

۱-حسابرسی هر چیزی که به انبار اضافه می شود.

۲-حسابرسی هر چیزی که از انبار خارج می شود.

۳-سفارش دادن

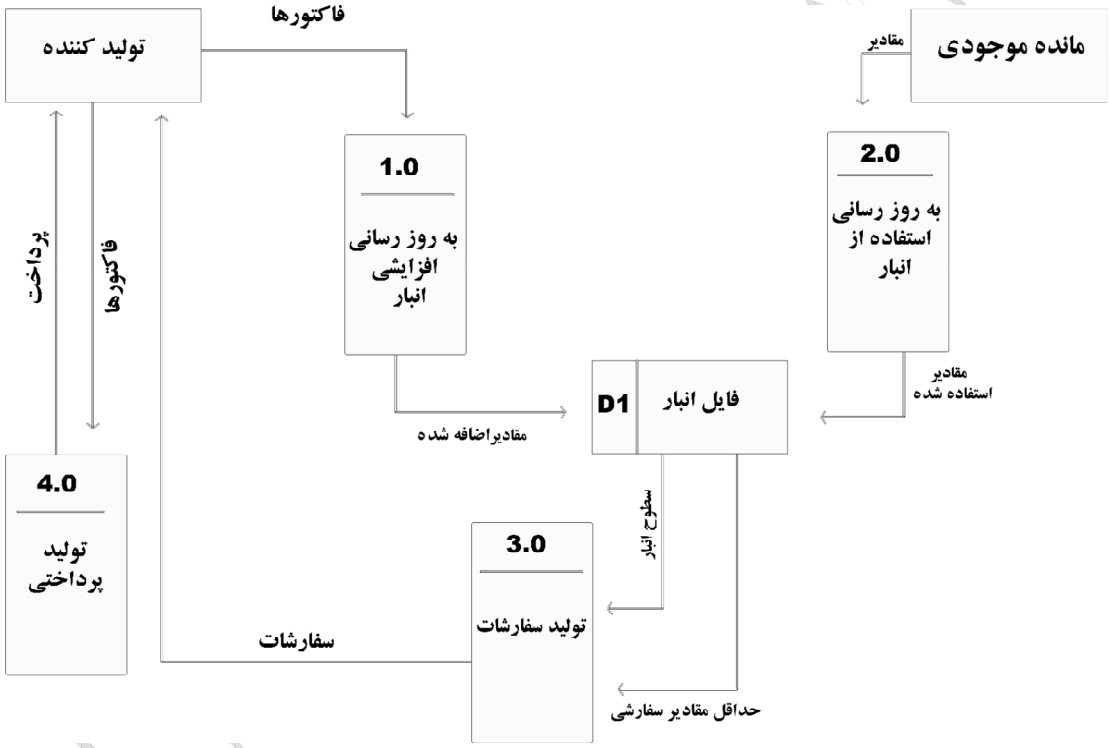
۴-پرداخت صورت حسابها.

داده کلیدی استفاده شده توسط سیستم ، فاکتورها و

مقادیر موجودی انبارند (اگر تعیین شده باشند)،خروجی های اصلی همانند قبل سفارشات و پرداخت ها هستند .

تمرکز روی عناصر اساسی سیستم منجر به DFD سیستم منطقی فعلی نشان داده شده در شکل زیر می شود.

نمودار جریان داده سطح صفر برای سیستم منطقی فعلی کنترل انبار



منظور از DFD منطقی جدید:

۱- نمایش هر عملکرد اضافی ضروری در سیستم جدید.

۲- تعیین مؤلفه های قدیمی که بایستی حذف شوند.

۳- تعریف هر یک از تغییرات جریان منطقی داده ها بین مؤلفه های سیستم شامل مخازن داده مختلف برای سیستم انبار برگرانیدپانا.

باب ملان کمپ تمایل دارد سه عملکرد اضافی داشته باشد:

۱. داده های یک مجموعهٔ جدید از یک سیستم ماش ینی وارد شود، بنابراین از برگه های گزارش موجودی کاغذی اجتناب می شود.

۲. سیستم بطور خودکار تشخیص دهد که آیا سفا رشی جدید بایستی داده شود . سفارش خودکار خیال باب را اینکه در تمامی زمانها بقدر کافی از هر چیزی در انبار موجود است راحت می کند.

۳. باب تمایل دارد در هر زمانی سطوح تقریبی انبار برای همه کالاهای انبار را بداند.

برای بعضی از کالاها نظیر نان همبرگر، باب می تواند با چشم مقادیر انبار را بازرسی نماید و بطور تقریبی تعیین کند که چه مقدار باقی مانده و قبل از اتمام وقت چه مقدار مورد نیاز است اگرچه برای سایر کالاها باب خیلی سریعتر از آنکه بازرسی چشمی انجام دهد نیاز به یک تخمین سخت و سریع آنچه در انبار موجود است، دارد.

نمودار جریان داده سطح صفر برای سیستم منطقی جدید کنترل انبار

توجه داشته باشید که DFD منطقی جدید تقریباً با DFD فعلی یکسان است . تنها تفاوت پروسهٔ جدید ۵,۰ است که امکان پرس و جو از داده های انبار برای بدست آوردن تخمین از مقدار کالایی که در انبار موجوداست، را فراهم می نماید.

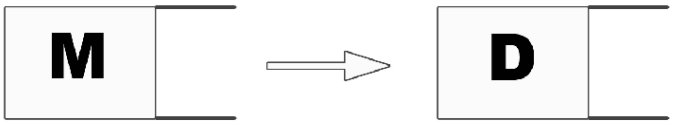
پروسهٔ ۱ به روز رسانی آنچه به انبار اضافه شده، مشخص نمی کند که بروز رسانی بصورت online یا غیر online (دسته ای)است یا اینکه به روز رسا نی روی کاغذ صورت می گیرد

یا به عنوان بخشی از سیستم ماشینی . بنابراین و رود بلافاصله داده های محموله به سیستم ماشینی در پروسهٔ ۱لحاظ شده است . بطور مشابه پروسهٔ" ۳ تولید سفارشات" مشخص نمی کند که آیا باب با کامپیوتر سفارشات را تولید می کند، یا بصورت online یا غیر online ایجاد می شوند . پس تقاضای باب در مورد اینکه سفارشات بطور خودکار توسط سیستم تولید شوند، از قبل توسط پروسهٔ ۳ نشان داده شده است .مرحلهٔ بعدی

ایجاد یک دیاگرام جریان داده برای سیستم فیزیکی جدید است ،که نشان دهد تقاضای باب صورت گرفته است .این مرحله با توجه به راه حل های متفاوتی که برای مشکل انبار باب وجود دارد متفاوت خواهد بود.

نکات قابل توجه در تبدیل از DFD فیزیکی به منطقی:

۱- تبدیل مخازن داده دستی به ماشینی (Data Store)



۲- حذف DS های موقت

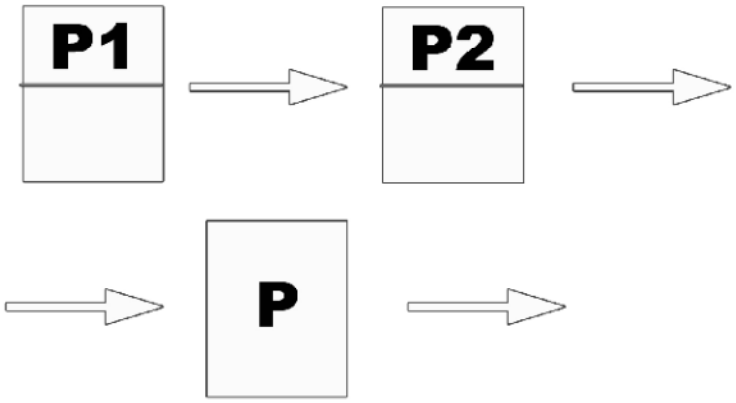
۳- حذف فرآیندهایی که کامپیوتری نمی شوند. (مثلا تحویل کتاب در سیستم کتابخانه)

۴ - حذف فرآیندهایی که عناوی ساز مانی دارند . (مثلا فرآیند مرتب کردن پرونده ها توسط اشخاص (یا) مثلا بایگان نامه ها را بایگانی می کند)

۵-حذف فرآیندهایی که داده را تغییر نمی دهند.

۶-حذف فرآیندهایی که کار یکسانی روی داده انجام می دهند.

۷-ترکیب فرآیندهایی که به یک جریان داده متصل هستند

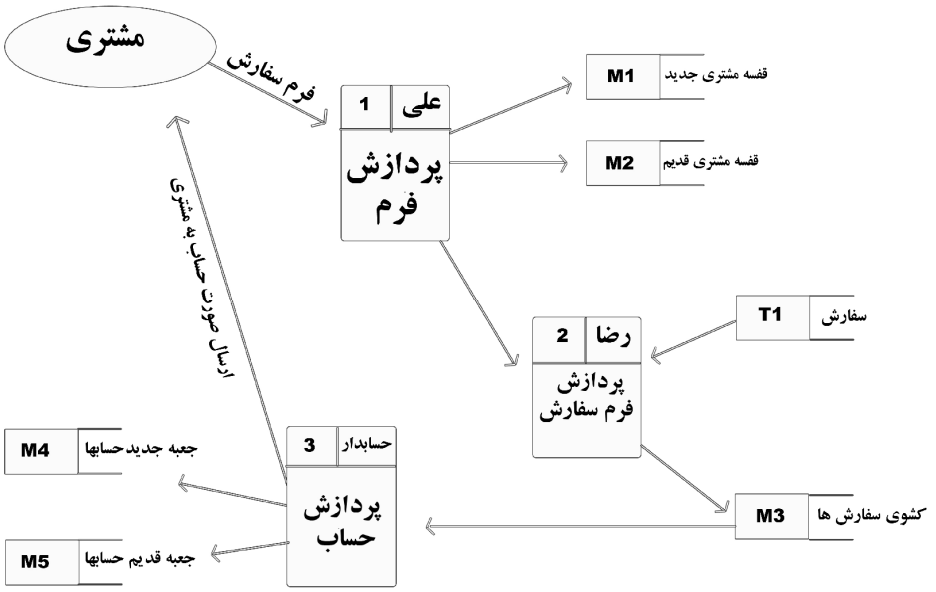


۸-حذف فرآیندهایی که به خاطر نسبت با سیاست ایجاد شده اند(موارد خاص و استثنائات)

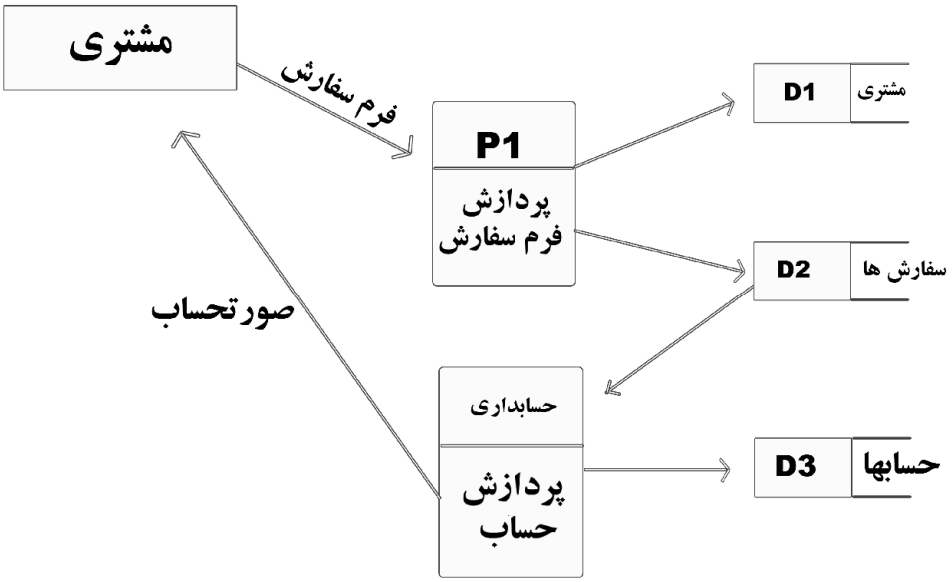
۹-حذف دسترسی های فیزیکی

مثال:

DFD فیزیکی زیر را به DFD منطقی تبدیل کنید.



DFD منطقی:



ممکن است از شما سؤال شود که آیا واقعاً برای یک تحلیلگر لازم است که مجموعهٔ کامل ی از نمودارهای جریان داده بسازد؟

خیلی از کارشناسان امروزه به این سؤال پاسخ منفی می دهند، یعنی تجزیه و تحلیل بایستی هرچه سریعتر با DFD منطقی جدید شروع شود.

بعضی از تحلیل گران به دلیل سه فرضیه توصیه می کنند که هر چهار مرحلهٔ DFD ها ساخته شود:

۱- تحلیل گران دانش کمی از شغل کاربر دارند و نیاز است که یک DFD فیزیکی فعلی مشروح فراهم نمایند تا شغل یا حرفه را درک کنند.

۲- کاربران قادر نیستند با یک DFD منطقی جدید آنطور که باید کار کنند.

۳- کار زیادی برای برگردا ندن DFD منطقی فعلی به DFD منطقی جدید وجود ندارد.

تحلیل گران تمایل دارند وقت زیا دی به ایجاد و پالایش مجموعۀ مشروحی از DFD ها برای سیستم فیزیکی فعلی اختصاص دهند، که عمدۀ آن در تبدیل به DFD بی منطقی فعلی هدر می رود . ما توصیه می کنیم که به سیستم فیزیکی فعلی تا حدی بپردازید که د یدگاه خوبی از سیستمی فعلی داشته باشید . ما با کارشناسان موافقیم که تمرکز بایستی روی سیستم منطقی جدید باشد.

راهنمای ترسیم DFD

علاوه بر موارد ذکر شدۀ قبلی بایستی به این موارد اضافی نیز توجه نمائیم.

۱- completeness کامل بودن با حدی که در آ ن همۀ مؤلفه های ضروری نمودار جریان داده لحاظ شده و بطور کامل توصیف شده اند.(هر یک از مؤلفه ها بایستی در دیکشنری پروژه کاملا توصیف شوند).

هر ابزار case این قابلیت را دارد که هر وقت پروسه، جریان داده، مبدأ، مقصد و یا مخزن داده ای تعریف می کنید بطور خودکار ی ک ورودی برای آن مؤلفه به انباره اضافه نماید . بعداً شما بایستی به انبار ه رفته توصیف عناصر را کامل کنید.

۲- Timing زمان: DFD ها ،در نمایش زمان انجام عملیات کاری نمی توانند انجام دهند.

(State transition Diagram) نمودار تبدیل حالت به این کار می آید.

۳- Iterative Development توسعه تکراری :توسعۀ تکراری DFD تشخیص می دهد که تعیین ملزومات و سازماندهی ملزومات روی هم اثر متقابل دارند، نه رشته ای از مراحل فاز تجزیه تحلیل SDLS.

:Primitive DFD

DFD های اولیه ،پایین ترین سطح تجزیه اند (یعنی رسیدن به پایین ترین سطح منطقی)

قوانینی برای توقف تجزیه:

-وقتیکه پروسه را به یک تصمیم منحصر بفرد، یا محاسبات و یا یک عمل پایگاه داده تک ، نظیر بازیابی ،ایجاد، حذف یا خواندن کاهش دهید.

-وقتیکه هر مخزن داده بیانگر داده ای در مورد یک موجودیت یکتا نظیر مشتری، کارمند، محصول یا سفارش باشد.

-و قتیکه کاربر سیستم ،توجهی به جزئیات بیشتر ندارد، یا وقتیکه شما و سایر تحلیل گران جزئیات مستند شده کافی برای وظایف بعدی توسعۀ مستقیم در دست دارید.

-وقتیکه جریان داده نیاز نیست به بخش های بعدی تقسیم شود، تا نشان دهد که داده های مختلف به روشهای مختلف بکار گرفته می شوند.

-وقتیکه شما عقیده داشته باشید که هر فرم تجاری یا فعل و انفعالات ورود اطلاعات خطی و گزارش گیری را به صورت یک نمودار جریان داده منحصر به فرد نمایش داده اید . (این امر اغلب بدین معنی است که بعنوان مثال هرنمایش سیستم و عنوان گزارش مطابق با نام جریان داده منحصر بفرد است.)

-وقتیکه شما عقیده دارید، پروسهٔ مجزایی برای انتخاب هر یک از گزینه های پایین ترین سطح برای سیستم وجود دارد.

چند سوال:

-نمودار جریان داده چیست؟چرا تحلیلگران ازDFD استفاده می کنند؟

CD چیست؟ DFD سطح n (۰ یا ۱ ویا...) چیست؟

قوانین مربوط به ترسیم DFD؟

تجزیه چیست ؟ تجزیه کی متوقف می شود؟

Balancing چیست؟چگونه تشخیص می دهید که DFD ها بالا نس نیستند؟

چرا DFD می توان د بعنوان وسیلهٔ تجزیه و تحلیل مورد استفاده قرار گیرد؟

....-

نمونه ای از فرم های مورد نیاز:

فرم موجودیت خارجی

شماره	نام	توضیحات
S1	مشتري	تحويل فرم سفارش و دريافت صورتحساب

فرم جریان داده

شماره	مبدأ	مقصد	نام	توضیحات
	S1	P2	فرم سفارش	

از تعیین ملزومات شما می فهمید که سیستم فعلی چه می کند و کاربران دوست دارند سیستم جدید چه کاری انجام دهد.

از سازماندهی ملزومات، شما می ف همید که در سطح منطقی ،مستقل از هر پیاده سازی فیزیکی، جریانات داده، منطق پردازش و جریان پردازش در سیستم جایگزین چگونه باید شکل گیرد.

یک **استراتژی طرح** سیستم، روشی برای توسعه سیستم است که شامل عملکردهای سیستم، سطوح سخت افزاری و نرم افزاری سیستم و روش فراگیری آن می باشد.

طراحی سیستم:

در طراحی بایستی ویژگیهای زیر در نظر گرفته شود.

۱-پیچیدگی : سیستم ها حتی الامکان ساده طراحی شوند.

۲-قابلیت حمل : طراحی سیستم بگونه ای باشد که با هرسخت افزار و نرم افزاری قابل اجرا باشد.

۳-نگهداری سیستم :کم هزینه باشد

مرحلهٔ طراحی را می توان به دو بخش طراحی کلی و طراحی تفصیلی(جزیی) تقسیم کرد.

در مرحله **طراحی کلی** سیستم عملیات زیر انجام می شود:

۱-تعریف اهداف و نیازمندیهای سیستم جدید (رفع مغایرتها و سوء تفاهم های بین تحلیل گر و مشتری)

۲- DFD رسم منطقی سیستم جدید (ماکزیمم تا پایان سطح دوم،چون کلیات مدنظر است)

۳-تشریح کلی اجزای سیستم پیشنهادی : واسط کاربر،خروجیها، ورودیها، پردازش ها، پایگاه های داده و کنترل هایی که بایستی صورت گیرد.

تحقیق: واسط کاربر(User Interface)

در طراحی کلی ابتدا خروجیها مشخص می شوند و بعد با توجه به خروجیها می توان ورودیها را تعیین کرد.

خروجیها :خروجیها با مطالعه نیازهای کاربران تعیین می شوند.وطراحی آن (تعیین خروجیها) با توجه به نیازهای کاربران است.در طراحی خروجی بایستی نکات زیررادر نظرگرفت:

۱-مرتبط بودن با نیاز کاربر

۲- به روز بودن (UPDATE)

۳- صحت و دقت

۴-نحوهٔ دریافت

۵-شکل و محتوا

براساس شرح وظایف کاربران می توان فهمید که سیستم اطلاعاتی چه خروجی هایی باید به کاربر بدهد.

فرم تعيين نياز هاي اطلاعاتي كاربر				
نام زير سيستم: آموزشي نام كاربر: متصدي رشته نام تحليل گر:				
رديف	شرح وظايف	اطلاعات مورد نياز	منبع يا نحوة دريافت	تناوب دريافت
1	راهنمايي دانشجو در زمان انتخاب واحد	- معدل كل دانشجو - تعداد واحدهاي گذرانده - تعداد ترم هاي مشروطي - معدل ترم - تعداد واحدهاي ترم		
2	صدور گواهي اشتغال به تحصيل	- وضعيت ترم جاري - اطلاعات مورد نياز آن دوره		

دو روش دیگر که کمتر مورد استفاده قرار می گیرد،عبارتند از:

۲- تعیین خروجیها براساس غربال (filtering method)

خروجیها از مکانی به مکان دیگر و از کاربری به کاربر دیگر باید پالایش شوند . یعنی از شکل ی به شکل دیگر درآید در حقیقت خروجیها تغییر شکل داده و خلاصه می شوند به نحوی که خیلی از جزئیات حذف می گردند.

۳- روش نظارت کردن (Monitoring Method)

تعیین متغیر های کلیدی (مثلا در سیستم تعیین موقعیت شرکت در بازار :تعداد مشتریان،فروش دوره های قبل، میزان سفارشات دورهٔ جدید)

-گزارش مغایرتها (تفاوت بین پیش بینی ها و واقعیت ها)

ورودی ها : جهت تعیین ورودیهای مناسب بایستی به موارد زیر توجه کرد:

۱-ذیربط بودن با خروجی

۲- به روز بودن (up to date)

ورودی ها می توانند به دو صورت باشند:

Batch
Online

۳- به سادگی قابل ثبت و ورود به سیستم باشد

۴-صحت و دقت(صحیح بودن و دقیق بودن ورودیها)

پردازش ها:

از روی DFD های ترسیم شده می توان پردازش های اصلی را تعیین کرد.

پایگاه های داده : محل نگهداری فایل ها در سیستم می باشند و تحلیل گر بایستی پایگاه های داده را برای برنامه نویس تشریح کند . فایل های مورد نیاز از روی DFD مشخص می شوند.

در یک سیستم دستی فرم هاواسناد و مدارک همان فایل ها می باشند که برای مکانیزه کردن سیستم بایستی مجدداً طراحی شوند

جهت تشریح فایل ها از **فرهنگ داده** ها یا Data Dictionary استفاده می شود. دیکشنری داده شامل دیکشنری ساختار داده(تشریح ساختار فایل ها) ودیکشنری اجزاء داده (تشریح تمامی فیلدهای موجود در فایل ها بطور مشروح) می باشد،که مورد اول در این مرحله ومورد بعدی در مرحله طراحی جزیی مشخص می گردد.

دیکشنري ساختار داده ها
نام فایل: سفارش مشتري شرح: جهت نگهداري سفارشات يك يا چند قلم كالا ترکیب: مشخصات مشتري = [نام و نام خانوادگي/ نام شرکت] + (کد اقتصادي) + آدرس مشتري شهر + خیابان + پلاك + کدپستي + جزئیات اقلام سفارشي [کد کالا + نام کالا + شرح + اندازه + تعداد + قیمت واحد]

=حاوی + ترکیب ** توضیحات

[/]فیلد انتخابی ( ) فیلد اختیاری

{ }فیلد تکراری @ فیلد کلید

دردیکشنری اجزاء داده اسامی فیلدها به همراه مشخصات تفصیلی آنها بیان می گردد:

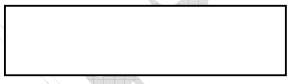
دیکشنري اجزاء داده							
نام فایل: سفارش مشتري							
نام فیلد	شرح	نوع	طول	دامنه مقادیر	مقادیر غیر مجاز	اسامي دیگر	ملاحظات
نام کالا		کاراکتري	35				
کد کالا			2	01-99			

تشریح ارتباطات داده ها (Entity Relationship Diagram)

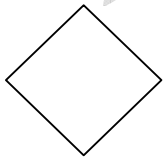
ERD یا ER به تشریح ارتباط داده های در حالت سکون در یک سیستم ، فارغ از هر گونه تغییر و تبدیل می پردازد (درDFD توجه اصلی به جریان داده ها بود).

هر سیستم مجموعه ایست از داده هایی که در خود ذخیره می کند،این ذخیرهٔ داده ها عناصر یا موجودیت سیستم را توجیه می نماید . نمادهایی که در ERD استفاده می شوند عبارتند از:

۱- Object Type یا نهاد (همان فایل ها و موجودیت ها درDFD)



۲- Relationship یا رابطه (که نحوهٔ ارتباط بین نهادها یا موجودیت ها را نمایش می دهد)

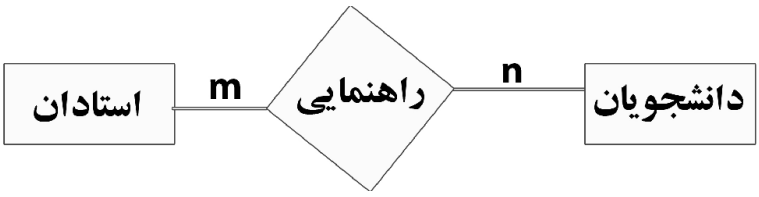


ERD می تواند مستقلا ، به موازات DFD یا بر مبنایDFD (از روی DFD رسم شده) ترسیم گردد.

مثال :در یک سیستم انتخاب واحد مطلوبست نمایش ERD برای تشریح پایگاه های داده و ارتبا طات آنها برای برنامه نویس.

نهادها :دروسی ارائه شده، دانشجویان، اساتید، برنامهٔ درسی دانشجو

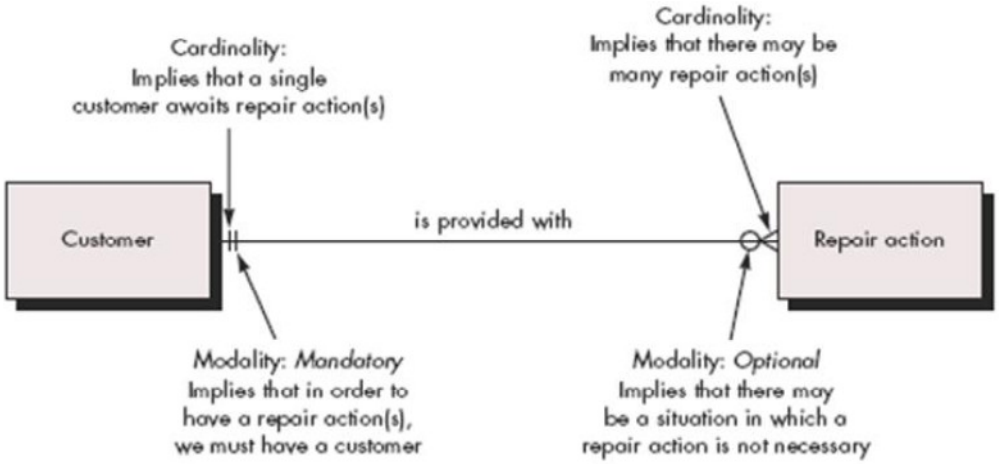
Relationship Cardinality درجه یا کمیت: یعنی چند به چند بودن رابطه نهادها،که می تواند ۱:۱ با۱:N یا M:N باشد.



یک استاد می تواند چند دانشجو را راهنمایی کند یا یک دانشجو می تواند توسط چند استاد راهنمایی شود.

هر دانشجو یک برنامهٔ درسی ایجاد می نماید.

Relationship Modality ضرورت یاکیفیت :یعنی الزامی یا اختیاری بودن رابطه نهادها.



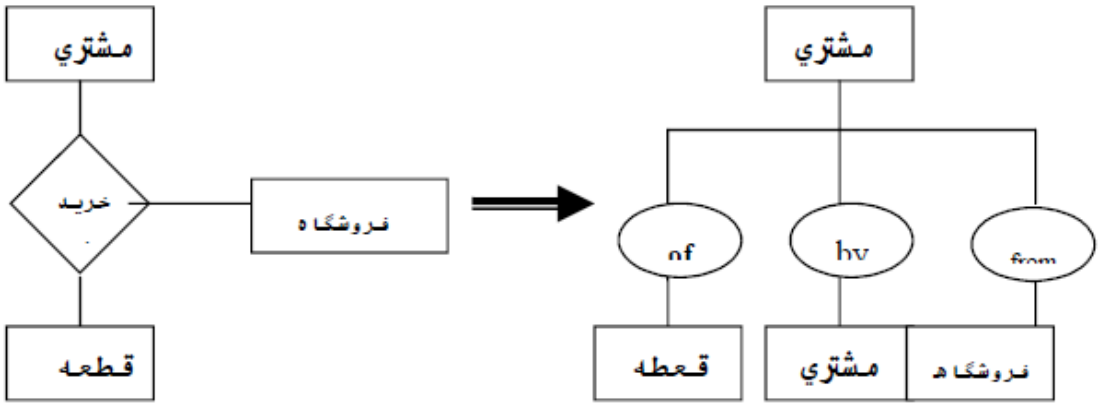
نتیجه: ERD نشان می دهد که چه داده هایی در یک فایل با چه داده هایی از فایل دیگر در رابطه اند و چه نوع رابطه ای دارند.

ترسیم ERD مبتنی بر DFD:

۱- فایل ها و موجودیت های DFD به نهاد در ERD تبدیل می شوند(داده های فایل = صفات نهاد).

۲- روابط را می توان از دل پردازشها به دست آورد(روابطی نظیر:met by ,Of , From , TO).

۳-حتی المقدور ازایجاد یک رابطهٔ بین چند نهاد خودداری کنید.



طراحی مجدد فرم ها:

حتماً مورد نیاز باشند.

اقلام اطلاعاتی ضروری را داشته باشند.

در اندازهٔ مناسب باشند.

در این مرحله فرم ها بایستی مجدداً طراحی شوند، می توان روند نمای فرم ها را کشید و در ساختار فرم ها و گردش جریان فرم اصلاحاتی انجام داد ساختار فرم می تواند به این شکل باشد:

شمارهٔ ترتیب نام فرم تاریخ نام سازمان آرم سازمان	مشخصات فرم
	اطلاعات ثابت اطلاعات متغیر
تأیید	تأیید و گردش فرم ها : رنگها ، توزیع نسخ

می توان برای فرم ها شناسنامهٔ فرم نیز در نظر گرفت:

شناسنامه فرم
نام فرم :
شمارهٔ فرم :
مبدأ صدور :
گیرندگان فرم :
تعداد نسخ :
شرح اقلام اطلاعاتي مندرج در فرم :
ابعاد استاندارد

نوشتن روش ها: روش ها یا پردازه هابه دو صورتند:

جزئی یا کمی :تشریح با P.D.E

کیفی :تشریح با Chapin، جداول تصمیم گیری یا P.S

در این قسمت تحلیلگر بایستی روش های انجام عملیات و پردازش اطلاعات را در پایین ترین سطوح DFD (primitive DFD) (پردازه های جزئی) که دیگر قابل تجزیه نیستند، تشریح نماید . برای اینکار می توان از دیکشنری پردازش استفاده نمود:

Process Dictionary Entry دیکشنري پردازش
نام پردازش: محاسبة حقوق ناخالص
شمارة پردازش:
شرح پردازش:
ورودیها: جریانات دادة ورودي به پردازش مثلا Rate, Hours
خروجیها: جریانات دادة خروجي از پردازش مثلا Pay
منطق برنامه: منطق پردازش با شبه کد نوشته مي شود

If hours-work > 40 then
Pay=40 *pay-rate + (hours-work -40)*(1.5*pay-rate)
Else pay= pay-rate *(hours-work
End if

Process Specification (PS)

نام پردازش: دریافت چك
شمارة پردازش:
شرح پردازش: چك هاي دریافت شده از مشتریان را گرفته و اطلاعات چك ها که شامل نام شعبه، مبلغ، تاریخ، نام پرداخت کننده است را وارد فایل چك مي نماید.

نکته :برای تمام پردازش ها و فایل ها نیاز به ترسیم دیکشنری نمی باشد بلکه فقط مواردی که امکان دارد ابهاماتی برای برنامه نویس داشته باشد ترسیم می شوند.

Data flow Dictionary Entry

نام جریان داده: دیکشنری جریان داده اطلاعات دانشجو
شماره جریان داده:
شرح جریان داده:
از پردازش: شمارة پردازش، نام پردازش یا موجودیت
به پردازش: شمارة پردازش، نام پردازش یا موجودیت
ساختار داده (دادة مورد نظر از چه اقلام اطلاعاتی تشکیل شده است.) نام، نام خانوادگی، شمارة شناسنامه و نام پدر و ...

کد گذاری های لازم در سیستم (Coding):

جهت طبقه بندی اطلاعات یا موجودیت ها در سیستم استفاده می شود.

در طراحی کدها بایستی موارد زیر رعایت گردد.

-ساختار کد منطبق بر نیازهای استفاده کنندگان یا روش های پردازش باشد.

-طراحی کدها بایستی قابلیت انعطاف جهت توسعه سیستم را داشته باشد.

-کدها قابل درک و از لحاظ منطق با مفهوم باشند.

-برای کاهش خطا بایستی از کد های حرفی عددی مشابه یکی را حذف کرد.

پیاده سازی:

شامل مراحل زیر است:

-کد نویسی

-تست

-نصب (تبدیل سیستم فعلی به سیستم جایگزین ، آموزش کاربران و تهیهٔ مستندات)

-بازنگری (انجام پاره ای عملیات نگهداری پس از تحویل) پس از انتخاب زبان برنامه نویسی مناسب بایستی موارد زیر را انجام داد:

نوشتن کد برنامه

تهیه راهنمای استفادهٔ کاربر از نرم افزار (user manual)

تهیه دستورالعمل استفاده از نرم افزار (operational manual) (راهنمای فنی و عملیاتی شامل روش های تجزیه و تحلیل طراحی برنامه و کلیهٔ مستندات مورد استفاده در توسعهٔ سیستم)

تبدیل سیستم فعلی به سیستم جدید:به روش های زیر صورت می گیرد:

- **مستقیم** یا یکپارچه(باریسک بالا وهزینه ی پایین،قابل استفاده در سازمانهای کوچک که سیستم شان پیچیده نیست.)

-**موازی** یا همزمان(باریسک پایین وزمان وهزینه ی بالا)

-**پیمانه ای** (در سازما□ایی که دارای شعب متفاوت است)

-**مرحله ای** یا تدریجی(تقسیم سیستم به چند زیر سیستم وجایگزین کردن زیر سیستم ها بصورت مرحله ای)دراین روش نیاز به یک سیستم واسط می باشد تا بین سیستم قدیمی وجدید که یکپارچه نیستند،ارتباط برقرار نماید.

-**ترکیبی** از روش های قبلی

تست:شامل تست قبل وبعد از پیاده سازی می باشد. تست قبل از پیاده سازی شامل تست واحد ،تست ماژول،تست زیر سیستم وتست سیستم می باشد.

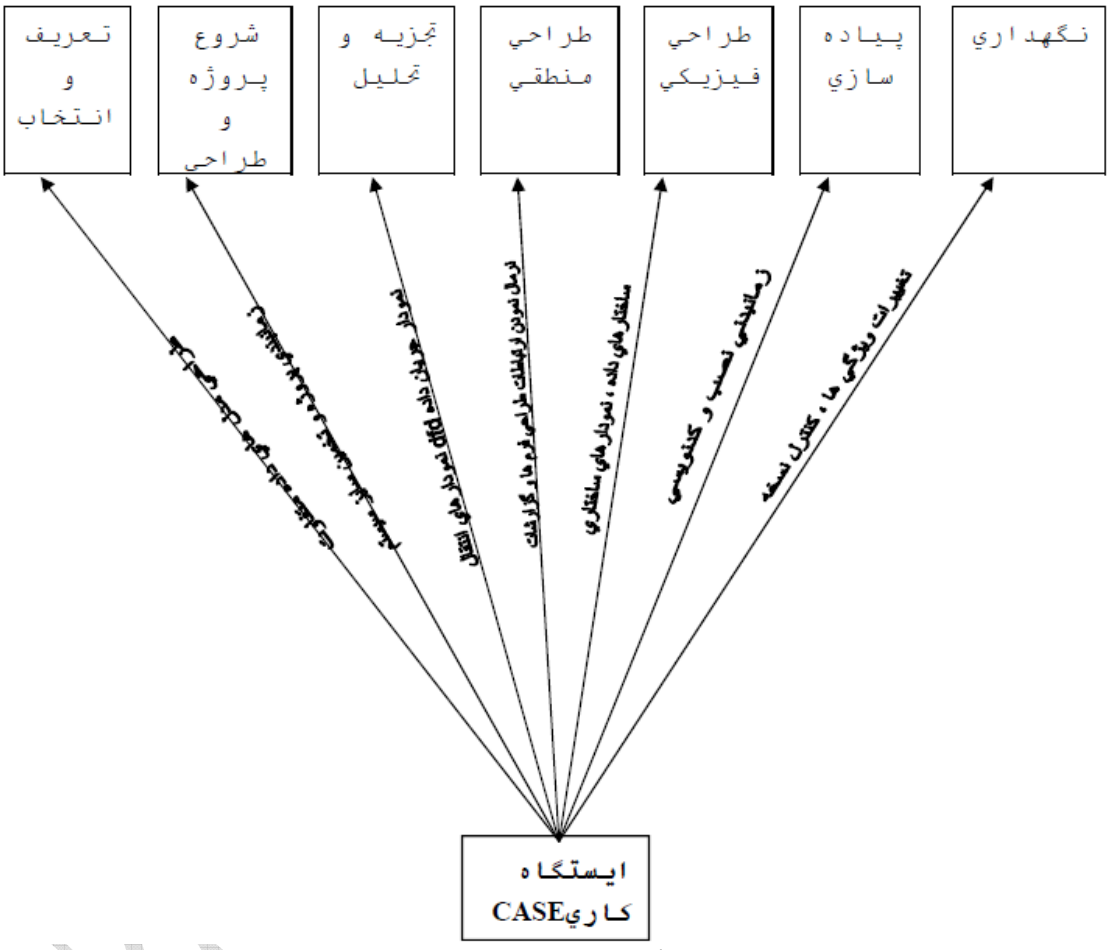
تست بعداز پیاده سازی شامل تست پذیرش α (بازنگری مجدد قبل از تولیدانبوه) و β (تولیدانبوه) می باشد.

Computer Aided Software Engineering (CASE)

ابزار نرم افزاری ماشینی که توسط تحلیلگران سیستم برای توسعهٔ سیستم های اطلاعاتی مورد استفاده قرار می گیرد. این ابزار می تواند برای ماشینی کردن فعالیته ای پردازش توسعهٔ سیستم ها ،با هدف افزایش سودمندی و بالا بردن کیفیت سیستم ها بکار روند.

استفاده از CASE در سازمانها

هدف از CASE تسهیل تصویب یک فلسفه طراحی واحد در یک سازمان با پروژه ها، سیستم ها و افراد زیاد است. CASE می تواند اکثر فعالیت های توسعهٔ سیستم را حمایت نماید، شکل زیر ویژگیهای بارز CASE در هر مرحلهٔاز چرخهٔ عمراز هفت مرحله ای را مشخص می نماید:



CASE سازندگان سیستم را در م دیریت پیگیریهای پروژه ی سیستم های اطلاعاتی یاری نموده، به تضمین ساخت سیستم های با کیفیت بالا در زمان مقرر و با بودجه پیش بینی شده کمک می نماید.

اهداف CASE : اکثر سازمان هاازCASE استفاده می نمایند تا:

-توسعهٔ سیستم ها را بهبود ببخشند.

-افزایش سرعت با سیستم ه ایی که از این طریق توسعه و طراحی شده اند.

- تسهیل و بهبود بخشیدن پروسه های تست با استفاده ازابزار بررسی ماشینی.

-بهبود جامعیت و یکپارچگی فعالیتهای توسعه از طریق متدولوژیهای معمول.

-بهبود بخشیدن کیفیت و تکمیل اسناد و مستندات.

-کمک به استاندارد سازی فرآیند توسعه.

-بهبود مدیریت پروژه.

-تسهیل نگهداری برنامه.

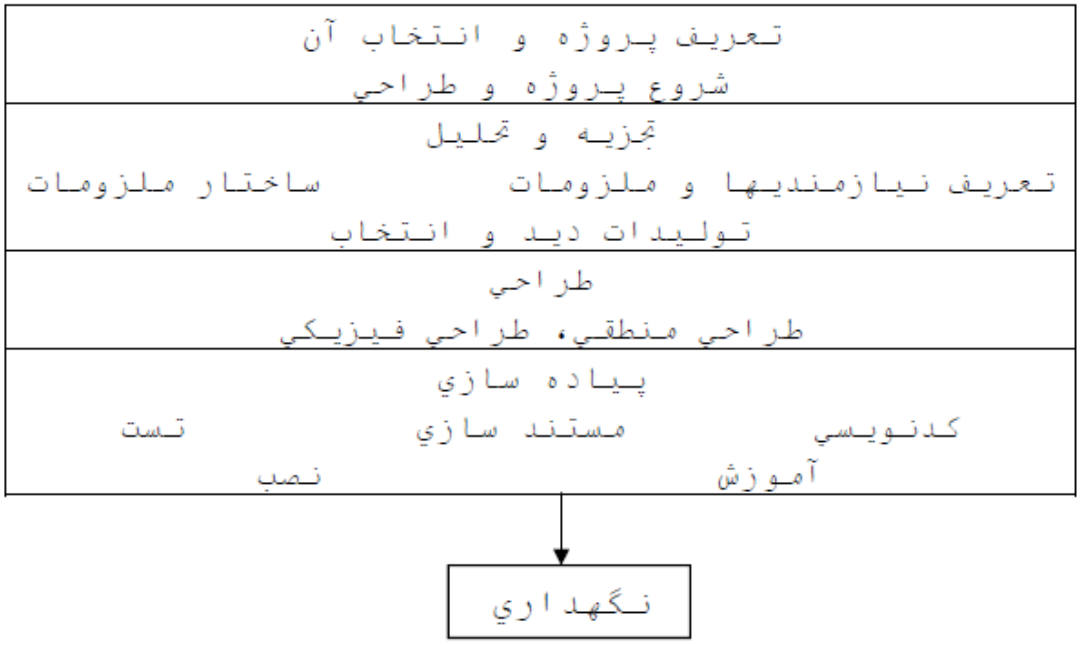
-پیش بینی قابلیت استفاده مجدد از ماژول ها و مستندات.

-توسعهٔ قابلیت حمل نرم افزار در محیط های مختلف.

اجزاء یا مؤلفه های CASE

ابزارCASE برای پشتیبانی فعالیت های متفاوت SDLC ، مورد استفاده قرار می گیرند .ابزار Upper CASE برای کمک به مراحل تعریف و انتخاب پروژه، شروع پروژه و طراحی، تجزیه و تحلیل و طراحی و ابزار Lower CASE در مراحل پیاده سازی و نگهداری در چرخهٔ SDLC بکار می روند .گروه سوم ابزار CASE ابزار cross life cycle هستند،که برای حمایت از فعالیت هایی درچندین مرحله از CASE مورد استفاده قرار می گیرند .به عنوان مثال برای کمک به پیشرفت فعالیتهایی نظیر مدیریت پروژه، بدست آوردن تخمین های زمانی برای فعالیت ها و ایجاد مستندات که در اکثرمراحل صورت می گیرند . در طی چند سال گذشته فروشندگان محصولات و ابزارهای متفاوت CASE استفاده از سیستم هایشان را از طریق استفاده از پایگاههای داده استاندارد و مبدل های داده آزاد نموده اند.

شکل زیر رابطهٔ بین ابزار CASE و چرخهٔ حیات سیستم را نشان می دهد:



مثالهایی از موارد کاربر CASE در SDLC:

مرحلة SDLC:	فعاليت هاي اصلي نمايش و سازماندهي اطلاعات سطح بالاي سازماني	مورد استفادة ابزار CASE ترسيم نمودار براي ايجاد و سازماندهي اطلاعات
تعريف پروژه و انتخاب		
شروع پروژه و طراحي	تعيين هدف پروژه و امكان سنجي براي فراهم نمودن طرح پروژه	توليد كننده هاي اسناد و انباره براي طرح پروژه
تجزيه و تحليل	تعيين و سازمان دهي نيازمنديها و ملزومات	رسم نمودار براي ايجاد مدل هاي پردازش منطقي و داده
طراحي منطقي و فيزيكي	تهيه طرح هاي سيستم جديد	توليد كننده هاي فرم و گزارش براي طرح هاي مقدماتي (نمونة اوليه) توليد كننده مستندات و تجزيه و تحليل براي تعريف ويژگيها
پياده سازي	تبديل طرح به سيستم اطلاعاتي	توليد كننده هاي كد، آناليز،فرم و گزارش براي تهية سيستم، توليد كننده هاي مستندات براي توسعة سيستم و اسناد كاربر
نگهداري	ارزيابي سيستم اطلاعاتي	همة ابزارمورد استفاده قرارميگيرند (تكرارچرخةحيات)

جدول زیر توسعهٔ سیستم های متعارف در برابر توسعهٔ سیستم مبتنی برCASE را نمایش می دهد:

توسعة سيستم هاي معمول	توسعة مبتني بر CASE
تأكيد روي كد نويسي و تست	تأكيد روي تجزيه و تحليل و طراحي
ويژگيهاي مبتني بر كاغذ	توليد نمونة اوليه محاوره اي سريع
برنامه نويسي دستي	توليد كد ماشيني
تهية مستندات دستي	توليد مستندات ماشيني
تست نرم افزار	بررسي طراحي ماشيني
نگهداري كد و مستندات	نگهداري ويژگيهاي طراحي

در ادامه به بحث بیشتری روی مؤلفه های CASE می پردازیم

ابزار ترسیم نمودار CASE

ابزاری هستند که برای ایجاد و نمایش گرافیکی عناصر گوناگون سیستم نظیر جریان پردازش ها ،ارتباطات داده و ساختارهای برنامه بکار می روند.

ابزار تهیهٔ گزارش و فرم

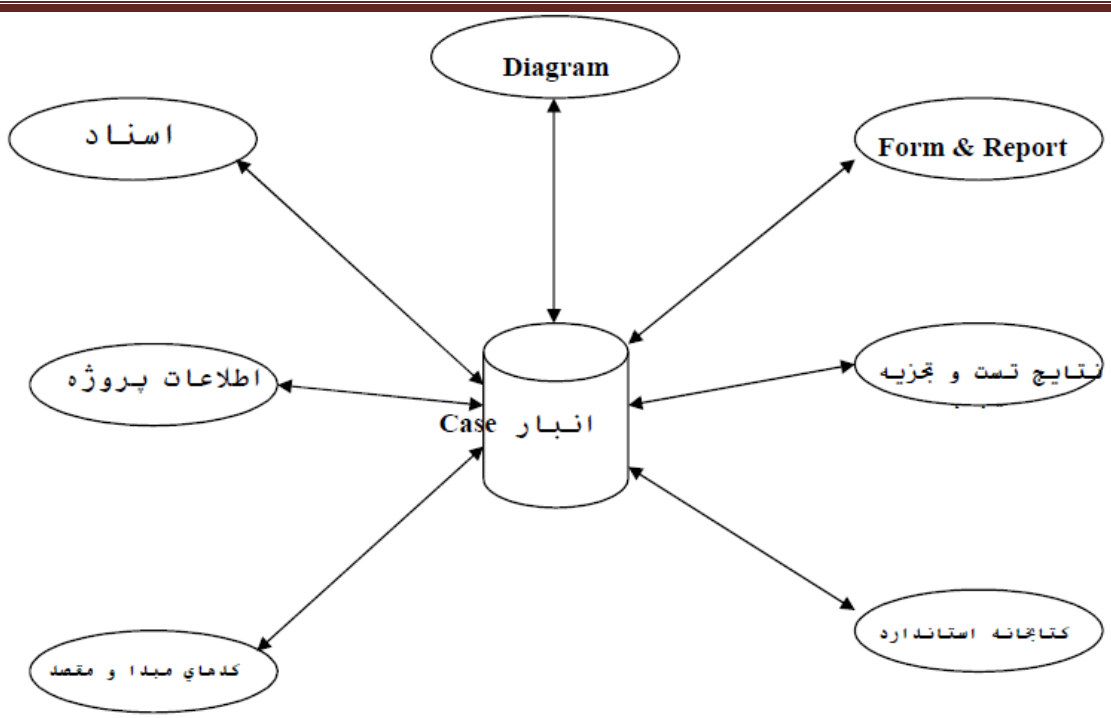
ابزاری که برای ایجاد فرم ها، گزارش ها به منظور قابل لمس ساختن الگوی اولیه برای کاربران بکار می رود.

ابزار تجزیه و تحلیل CASE

ابزار تجزیه و تحلیل که بطور خودکار ویژگیهای ناقص، متناقض وغلط را در دیاگرام ها، فرم ها و گزارشات بررسی می نماید.

انباره یا انبار CASE(Repository)

یک بانک اطلاعاتی متمرکز که شامل همهٔ دیاگرام ها، تعریف فرم ها، گزارشات، ساختارهای داده، تعاریف داده ها، منطق و جریان پردازش و تعریف سایر مؤلفه های سیستم وسازمانی که مجموعه ای از مکانیزم ها و ساختارها برای نایل شدن به جامعیت در کلیهٔ مراحل SDLC را داراست.



ابزار تولید اسناد: ابزاری ازCASE هستند که تولید ساده و آسان مستندات فنی و همچنین مستندات کاربر را به شکل استاندارد ممکن می سازند.

ابزار تولید کد:

ابزاری که تولید ماشینی برنامه ها و کدهای تعریف پایگاههای داده را به طور مستقیم از مستندات طراحی، فرم ها و گزارشات ذخیره شده در انبار ممکن می سازد.

با استفاده از CASE مهندسی مجدد و مهندسی معکوس ممکن می باشد.

مهندسی معکوس(Reverse Engineering) :

ابزار ماشینی که کد منبع برنامه را بعنوان ورودی پذیرفته و نمایش تصویری و متنی و اطلاعات سطح طراحی نظیر ساختارهای کنترل برنامه،ساختار های داده ، جریان منطقی و جریان داده را تولید می نماید.

مهندسی مجدد:

ابزار ماشینی که کد منبع برنامه را به عنوان ورودی پذیرفته و روی داده های برنامه و منطق برنامه تجزیه و تحلیل انجام داده و سپس بطور خودکار یا بصورت محاوره ای با یک تحلیل گر س یستم، سیستم موجود را تغییر می دهد تا کیفیت و کارآیی آن را افزایش دهد.