درس مهندسی نرم افزار

مدرس: مرضیه هدایت پناه

نرم افزار چیست؟

بسیاری از مردم، نرم افزار را با برنامه های کامپیوتری یکسان می دانند. نرم افزار فقط برنامه های کامپیوتری نیست، بلکه تمام مستندسازی ها و داده های پیکربندی را شامل می شود که برای درست کارکردن این برنامه ها ضروری اند. سیستم نرم افزاری معمولا شامل تعدادی از برنامه ها، فایل های پیکربندی برای تنظیم این برنامه ها و مستندات سیستم برای تشریح ساختار سیستم و مستندات کاربر برای تشریح چگونگی کار با سیستم است.

طبقه بندی نرم افزارها:

۱. **نرم افزاهای کاربردی:** این نرم افزارها توسط یک سازمان تولید کننده، تولید می شوند و به بازار عرضه می گردند و مشتریان می توانند برحسب نیاز آنها را تهیه کنند. مانند آفیس، فتوشاپ
۲. **نرم افزارهای سیستمی:** مجموعه ای از برنامه ها هستند که برای سرویس دهی به برنامه های دیگر طراحی و ایجاد می گردند. مانند سیستم عامل ها و برنامه های مدیریت فایل
۳. **نرم افزارهای سفارشی:** این ها نرم افزارهایی هستند که توسط مشتری خاصی سفارش داده می شود. این محصولات توسط پیمانکاران نرم افزا برای آن مشتری ایجاد می شود. مانند نرم افزار حسابداری
۴. **نرم افزارهای بلادرنگ:** نرم افزارهایی هستند که برای نظارت و تحلیل و کنترل رویدادهای محیط به محض وقوع این رویدادها طراحی و پیاده سازی شده اند که این نرم افزارها باید کمترین زمان ممکن به رویدادها پاسخ دهند. مانند سیستم کنترل اطلاعات هواپیماهای درحال پرواز.
۵. **نرم افزارهای علمی و مهندسی:** این نرم افزارها به وسیله الگوریتم هایی که اعداد و ارقام را پردازش می کنند به وجود آمدند و کاربردهای بسیاری در نجوم و زمین شناسی دارند.
نرم افزارهای شبیه سازی در این دسته قرار دارند.
۶. **نرم افزارهای مبتنی بر وب:** می توان گفت هر صفحه وب نرم افزاری است که دستورات برنامه نویسی مثل زبان html و داده هایی مانند تصاویر و فایل های صوتی را به هم متصل می کنند. به طور کلی می توان گفت شبکه وب به یک ابر کامپیوتر تبدیل شده که یک منبع نرم افزاری نامحدود را به وجود می آورد.
۷. **نرم افزارهای جاسازی شده:** امروزه محصولات هوشمند بسیاری یافت می شوند که تمامی این محصولات از نرم افزارهای جاسازی شده در حافظه فقط خواندنی(ROM) خود استفاده می کنند. بیشتر این نرم افزارها مقاصد کنترلی و صنعتی دارند و حتی در بعضی از لوازم منزل نیز استفاده می شوند. مانند نرم افزار کنترل آسانسور، کنترل صفحه کلید مایکروفر.
۸. **نرم افزارهای هوش مصنوعی:** این نرم افزارها از روش های غیرعددی برای حل مسائل پیچیده ای که با الگوریتم های عددی قابل حل نیستند، استفاده می کنند. مانند سیستم های خبره، پایگاه های دانش، تشخیص صدا.
۹. **نرم افزارهای کامپیوترهای شخصی:** نام دیگر این نوع نرم افزار، نرم افزارهای دسکتاپ است. مانند نرم افزارهای گرافیکی، سرگرمی

صفات نرم افزار خوب کدامند ؟

محصولات نرم افزاری صفاتی دارند گه کیفیت آن نرم افزار را مشخص می کند. چند نمونه از این صفات عبارتند از:

۱. **قابلیت نگهداری :** نرم افزارها باید طوری نوشته شوند که نیازهای جدید کاربر را برآورده کنند.
۲. **قابلیت اتکا**: شامل ویژگی هایی مثل **قابلیت** اعتماد، امنیت و حفاظت است. نرم افزار قابل اتکا نباید هنگام خرابی سیستم منجر به خرابی فیزیکی یا اقتصادی شود.
۳. **کار آمدی** : نرم افزار نباید منابع سیستم مثل حافظه و چرخه های پردازنده را هدر دهد. کارآمدی شامل پاسخگویی، زمان پردازش و بهره وری حافظه است.
۴. **قابلیت استفاده:** نرم افزار باید برای کاربردی که تهیه شده است قابل استفاده باشد. یعنی واسط کاربری مناسب و مستندات کافی داشته باشد.

مسئولیت تخصصی و اخلاقی:

مهندسین نرم افزار باید کارشان را در یک چارچوب و اجتماعی انجام دهند و قوانین معنوی، ملی و بین المللی پیروی کنند.

۱- محرمانگی.
۲- صلاحیت.
۳- حقوق معنوی.
۴- سوء استفاده از کامپیوتر.

عوامل موثردر قابلیت اعتماد سیستم :

۱- قابلیت اعتماد سخت افزار.
۲- قابلیت اعتماد نرم افزار.
۳- قابلیت اعتماد اپراتور .

سیستم چیست؟

سیستم مجموعه ای است که از اجزاء به هم وابسته که وابستگی حاکم بر اجزای خود کلیت جدید را احراز کرده و از نظم و سازمان خاصی پیروی می نماید و در جهت تحقق هدف معینی که دلیل وجودی آن است فعالیت می کند.

اجزاء سیستم :

۱. ورودی(input) : آنچه بنحوی وارد سیستم می شود و سبب تحرک سیستم می شود
۲. فرایند تبدیل(process) : جریان تغییر و تبدیل آنچه وارد سیستم می شود
۳. خروجی(output) : آنچه از تغییر و تبدیل از سیستم (به شکل کالایا خدمات)خارج می شود
۴. بازخورد (feed back) : فرایندی دورانی که قسمتی از خروجی به عنوان اطلاعات به ورودی بازخورانده می شود

کیس چیست؟

مهندسی نرم افزار به کمک کامپیوتر و دامنه وسیعی از انواع مختلف برنامه ها را در برمی گیرد که برای پشتیبانی از فعالیت های نرم افزاری مثل تحلیل خواسته ها، مدل سازی سیستم و عیب یابی بکار می آیند.

چرخه عمر توسعه سیستم (SDLC)

چرخه عمر سیستم چارچوبی است برای توصیف مراحل توسعه سیستم های اطلاعاتی. مجموعه ای از مراحل سیستم از ابتدا تا انتها را مشخص می کند این مراحل عبارتند از: طرح پروژه و تعریف خواسته ها، آنالیز(تحلیل)، طراحی، پیاده سازی، نگهداری و بکارگیری.

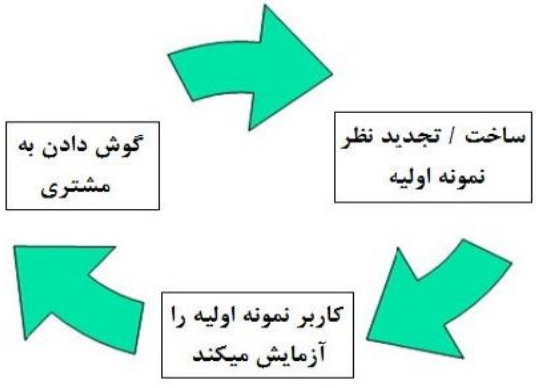
بعضی از مدلهای مشهور SDLC عبارتند از: مدل خطی، مدل آبشاری، مدل مارپیچی، مدل پیش نمونه سازی، مدل افزایشی و مدل اسکرام.

مدل خطی: این مدل را که چرخه حیات کلاسیک نیز می نامند، بیانگر یک نگرش نظام مند و زنجیری نسبت به تولید نرم افزار است.

مدل آبشاری: این مدل دارای مراحل خطی است یعنی هر مرحله از پروژه به طور کامل انجام می شود، وقتی آن مرحله تکمیل شد مرحله بعدی آغاز می شود. این مدل فرض می کند که نیازها پس از تعریف، پایدار و بدون تغییر باقی خواهند ماند. گرچه امکان برگشت به مرحله قبل در هر مرحله گنجانده شده ولی این برگشت هزینه بسیاری را بر دوش تیم می گذارد.

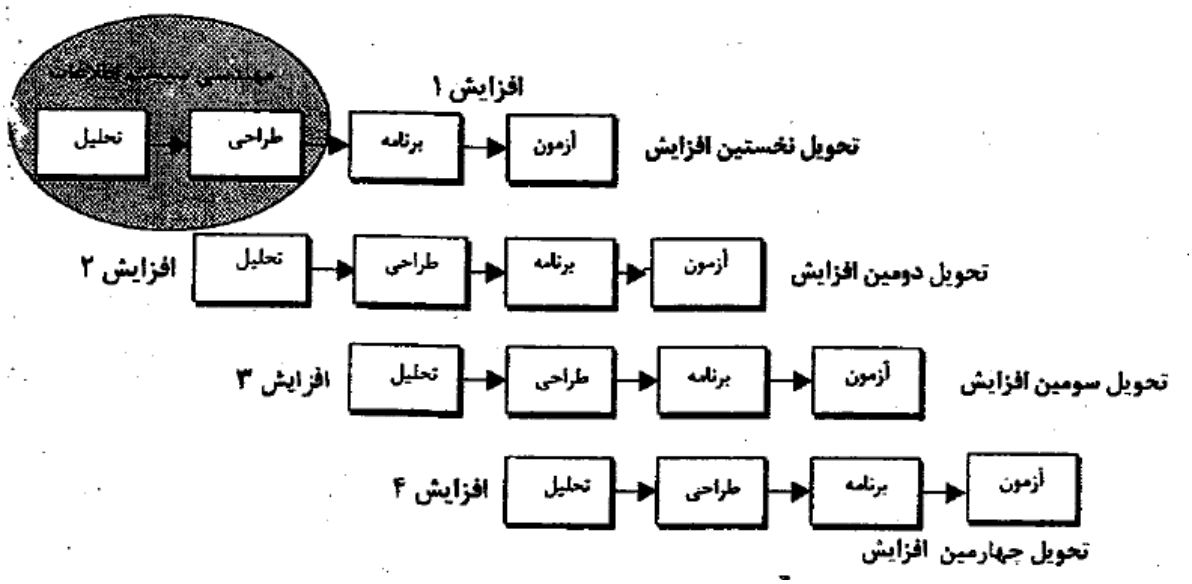
مدل مارپیچی: این مدل براساس تجربه حاصل از اصلاحات مدل آبشاری ایجاد شد. این مدل روش تکرار شونده را بجای روش خطی توسعه می دهد. در روش مارپیچی هر فاز، مرحله به مرحله کاملتر می شود. این کار اینقدر تکرار می شود تا سیستم به مرور زمان کامل شود. این روش در طول تکرارهای اولیه، ممکن است نسخه اولیه یک مدل روی کاغذ یا تنها الگوی اولیه باشد. در طول تکرارهای بعدی، نسخه کاملتری از سیستم تولید می شود.

مدل پیش نمونه سازی(prototyping model): این مدل زمانی بکار گرفته می شود که مشتری مجموعه ای از نیازها را بیان می کند، اما از ورودی و خروجی الگوریتم داخلی اطلاع کافی ندارد. یک نمونه ساده و کم حجم برای ارائه دادن به مشتری ایجاد می شود.



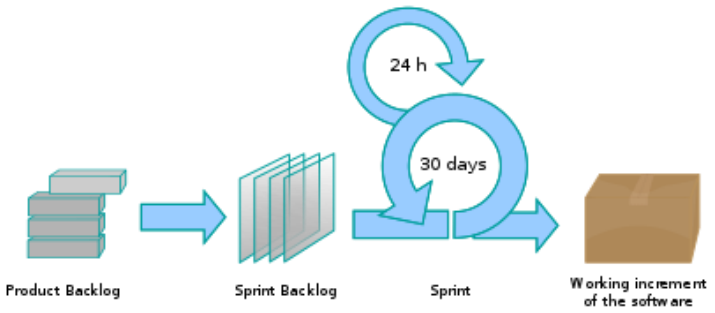
مدل افزایشی:

این مدل عناصر مدل خطی را با دیدگاه تکرار مدل نخست، ترکیب می کند. برای توسعه تدریجی نرم افزارهایی ارائه شد که در آن مرحله به مرحله قابلیت های محصول اضافه می شود. در مدل افزایشی در هر مرحله نرم افزار تکامل پیدا می کند و خیلی سریع بازار را در اختیار می گیرد. نسخه های بعدی نرم افزار به سرعت وارد بازار می شوند. (در هر مرحله تمام گروه ها در حال کار هستند)



مدل اسکرام (scrum):

در این مدل چرخه عمر تکرارها را sprint می نامند و به طور معمول ۳۰ روز به طول می انجامند. در هر روز کل اعضای تیم پروژه برای یک جلسه کوتاه مدت دورهم جمع می شوند که اسکرام نامیده می شود و در آنجا تصمیم می گیرند که در آن روز چه کارهایی انجام دهند. اعضای تیم موانعی را که ممکن است در رسیدن به اهدافشان وجود داشته باشد، تعیین می کنند و مدیر پروژه باید این موانع را کنار بگذارد.



مفاهیم مدیریت پروژه:

هنگام ساخت محصولات و سیستم های کامپیوتری ضروری به نظر می رسد مدیریت پروژه شامل طرح ریزی، نظارت و کنترل افراد، فرآیند و رویدادهایی می شود که به موازات تکامل نرم افزار از مفهوم مقدماتی تا پیاده سازی عملی رخ می دهند. هر کسی تا حدی مدیریت می کند ولی حوزه فعالیت های مدیریتی بسته به شخص تغییر می کند مهندس نرم افزار فعالیت های روزانه و وظایف فنی، کنترل، نظارت و طرح ریزی خود را مدیریت می کند. ساخت نرم افزارهای کامپیوتری وظیفه ای پیچیده

است، به ویژه اگر افراد زیادی در آن شرکت داشته باشند و مدت زمان کار نسبتا طولانی باشد از همین رو است که پروژه ها باید مدیریت شوند.

بازیگران: مدیریت پروژه کارآمد بر ۴ موردِ افراد، محصول، پروسه و پروژه تاکید دارد. بازیگران صحنه پروسه نرم افزار را می توان در یکی از پنج گروه زیر دسته بندی کرد:

مدیران ارشد که موضوعات و مسائل کاری را مشخص می کنند که اغلب تاثیر زیادی بر پروژه دارند.

مدیران فنی(پروژه) که باید افرادی را که کار نرم افزار را انجام میدهند سازماندهی و کنترل و نظارت کنند.

متخصصین افرادی که مهارت های فنی مورد نیاز جهت مهندسی محصول یا کاربرد ارائه می کنند.

مشتریان، کسانی که نیازمندیهای نرم افزار مورد نیاز را مشخص می کنند و سهامدارانی که در پی آمد آن نفعی دارند.

کاربران نهایی که با عرضه نرم افزار جهت استفاده از آن بهره می گیرند.

طرح پروژه و تعریف خواسته ها:

این مرحله شامل برنامه ریزی اولیه برای پروژه، تعریف خواسته ها، امکان سنجی، نقشه زمانبندی و... می باشد.

برنامه ریزی پروژه:

برنامه ریزی پروژه منابع پروژه را تنظیم می کند، توقف کار و زمان بندی انجام کار را مشخص می کند. در بعضی از سازمانها، برنامه ریزی پروژه تنها سندی است که حاوی تمام انواع برنامه ریزی های مطرح شده است. برنامه ریزی یک فرایند تکراری است که وقتی کامل می شود که خود پروژه کامل شده باشد و شامل بخش های زیر است:

۱ - مقدمه : اهداف پروژه را به طور مختصر توصیف می کند و محدودیت های موثر بر پروژه را اعلان می نماید (بودجه، زمان و...)

۲ - سازماندهی پروژه : سازماندهی پروژه مشخص می کند که تیم پروژه چگونه سازماندهی می شود. افراد سازمان و نقش آنها را مشخص میکند.

۳ - تحلیل ریسک : ریسک های ممکن در پروژه، احتمال وقوع این ریسک ها، و راهبردهای کاهش ریسک توصیف می کند.

۴ - منابع سخت افزاری و نرم افزاری مورد نیاز : سخت افزار و نرم افزار مورد نیاز برای انجام کار را مشخص می کند. اگر سخت افزاری باید خریداری شود، قیمت آن برآورده گردد و زمان بندی تحویل آن باید مشخص گردد.

۵ - توقف کار : توقف کار، پروژه را به صورت فعالیت هایی توصیف می کند و نقاط عطف و قطعات قابل تحویل را در هریک از این فعالیت ها مشخص می کند.

۶ - زمانبندی پروژه : وابستگی های بین فعالیت ها رو توصیف می کند، زمان تقریبی مورد نیاز برای رسیدن به نقطه عطف، و تخصیص نیروی انسانی به فعالیت ها را مشخص می نماید.

۷ - راهکارهای نظارت و گزارش: گزارش های مدیریتی را توصیف می کند که باید تولید شوند، مشخص می کند که این گزارش ها چه وقت باید آماده شوند و راهکارهای نظارت را تعیین می نمایند.

مطالعات امکان سنجی:

برای تمام سیستم های جدید، فرآیند مهندسی خواسته ها باید با مطالعات امکان سنجی شروع شود. ورودی مطالعه امکان سنجی، توصیف طرح کلی سیستم و چگونگی بکارگیری آن در سازمان است. خروجی مطالعه امکان سنجی : گزارشی است که پیشنهاد می کند آیا اجرای مهندسی خواسته ها و فرایند توسعه سیستم ارزشمند است یا خیر؟ این مطالعات شامل موارد زیر است : برآورد اطلاعات. جمع آوری اطلاعات. نوشتن گزارش.

تعریف خواسته های سیستم :

این فعالیت، عملکرد و خواص اساسی و مطلوب سیستم را مشخص می کند. در این فرایند با مشتریان و کاربران نهایی مشورت می شود. در این مرحله تعریف خواسته ها بر سه نوع خواسته تاکید دارد: ۱ -خواسته های عملکردی انتزاعی .(عملکردهای اساسی که باید توسط سیستم انجام گیرد، در یک سطح انتزاعی تعریف می شود.) ۲ - ویژگی های سیستم. (شامل ویژگی هایی مثل قابلیت دستیابی، کارایی، امنیت و... می باشد.) ۳- خواصی که سیستم نباید از خود نشان دهد.

طراحی سیستم :

مشخص می کند که عملکرد سیستم چگونه باید توسط قطعات مختلف سیستم انجام شود . فعالیت های موجود در این فرایند :

- **تقسیم بندی خواسته ها** (خواست ها تحلیل میشوند و در گروه های مرتبط به هم جمع آوری می گردند)
- **شناسایی زیر سیستم ها** (زیرسیستم های مختلف شناسایی می شوند که می توانند خواسته های سیستم را به تنهایی یا در کنار یکدیگر برآورده کنند. گروهی از خواسته ها برای زیر سیستم ها تعیین می شوند)
- **انتساب خواسته ها به زیر سیستم ها** (خواسته ها به زیر سیستم ها نسبت داده می شوند)
- **تعیین عملکرد زیر سیستم ها** (عملکرد هریک از زیر سیستم ها تعیین می شود. این کار ممکن است به عنوان بخشی از مرحله طراحی سیستم باشد. روابط بین سیستم ها باید در این مرحله مشخص شود)
- **تعریف واسط های زیر سیستم** .(در این جا واسط های مورد نیاز هریک از زیرسیستم ها تعریف می شوند. وقتی برروی واسط ها توافق به عمل آمد، توسعه موازی زیرسیستم ها ممکن خواهد شد)

مدلسازی سیستم:

در اثنای خواسته ها و طراحی سیستم ، سیستم باید بصورت مجموعه ای از قطعات و روابط بین آنها مدل سازی شود . این ها معمولا در مدل معماری سیستم به صورت گرافیکی تشریح می شوند. سیستم به چند زیر سیستم تجزیه می شود که هر زیر سیستم می تواند به زیر سیستم های دیگری تجزیه شود تا قطعات عملکردی به دست آیند .

توسعه زیرسیستم:

در این فرایند ، زیر سیستم هایی که در اثنای طراحی سیستم شناسایی شدند ، پیاده سازی می شوند.

جامعیت زیر سیستم :

زیر سیستم هایی که مستقل از هم توسعه داده شدند، در کنار هم قرار می گیرند تا سیستم کامل ایجاد کنند .

تکامل سیستم:

سیستم های بزرگ ، در اثنای زندگی شان، باید تکامل یابند تا خطاهای موجود در خواسته های اصلی سیستم را اصلاح کنند و خواسته های جدید را برآورده نمایند. ممکن است لازم باشد کامپیوترهای جدید و سریعتر بجای کامپیوترهای موجود قرار گیرد.

تجزیه سیستم:

سیستم پس از طول عمر مفید عملیاتی آن، کنار گذاشته می شود.

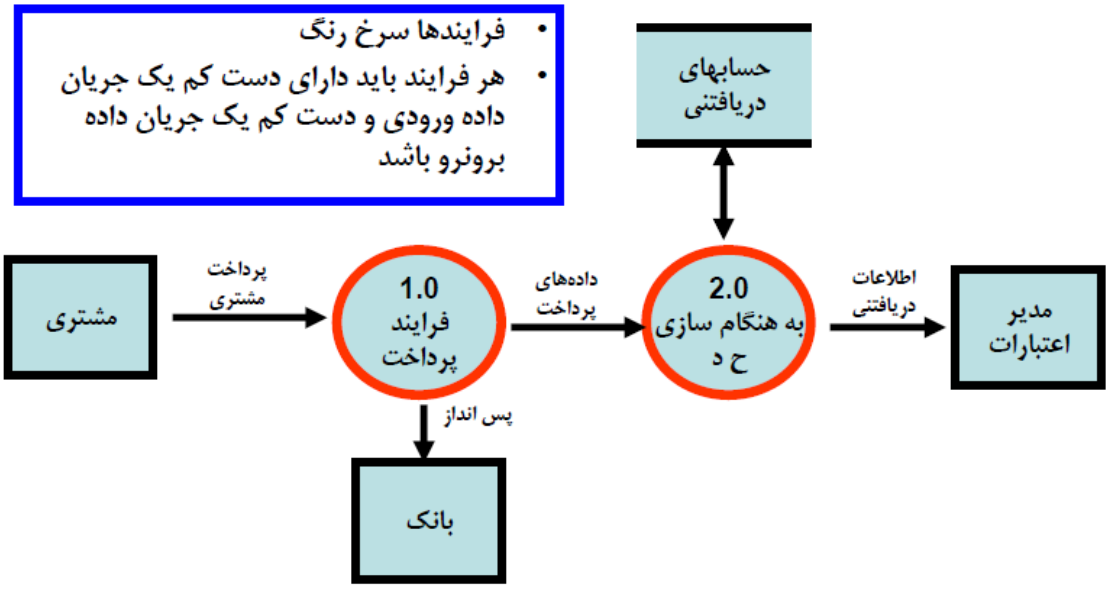
مدلسازی سیستم:

یک سیستم یا زیر سیستم در یک سازمان می تواند با مجموعه ای از روشها به صورت گرافیکی نمایش داده شود. این روشها می خواهند مرزهای سیستم را شناسایی کرده و اطلاعاتی که در یک سیستم جریان دارند را نشان دهند. نمودار جریان داده های یا Data Flow Diagram یکی از این روش ها می باشد.

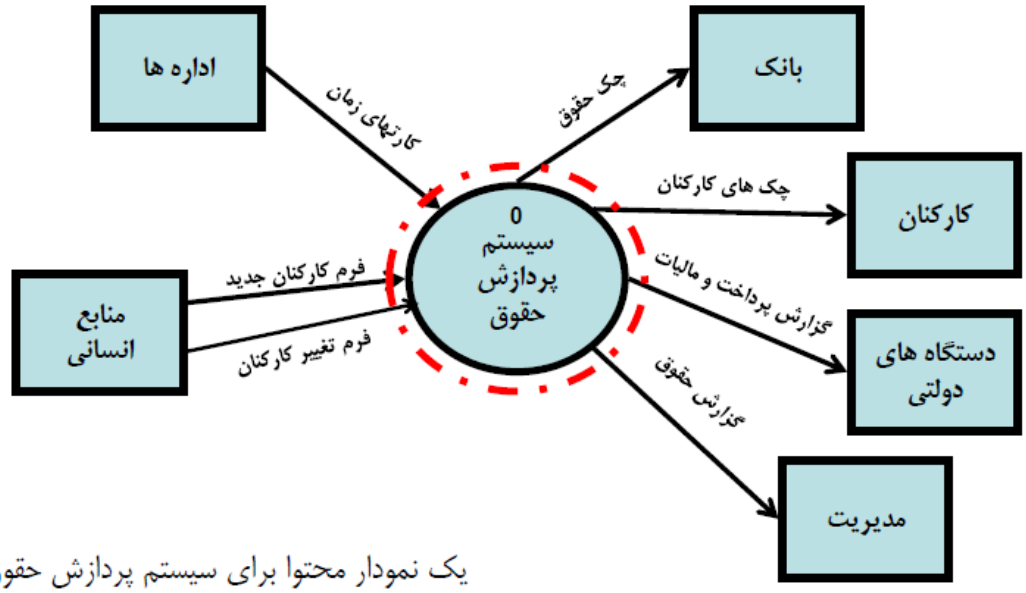
نمودار جریان داده ها (DFD) بر روی ورودی ها و خروجی های اطلاعات به یک سیستم و همچنین پردازش های انجام شده بر روی آنها تمرکز دارد. این نمودار با استفاده از چهار جزء اصلی ساخته می شود: یک مربع با سایه، فلش، مستطیل با گوشه های گرد و یک مستطیل با یک سر باز. این اشکال در شکل زیر نمایش داده شده اند. مربع با سایه برای نمایش نهاده ها و داده های (Entity) خارجی (یک دپارتمان دیگر، یک فرد یا یک ماشین) که می تواند به سیستم، داده ارسال کرده و یا از آن داده ای دریافت نماید، استفاده می شود. هر داده باید توسط اسم نامگذاری شود. برای جلوگیری از قطع خطوط جریان داده ها، یک داده می تواند چندین بار در یک نمودار تکرار شود.

فلش ها انتقال اطلاعات از یک نقطه به نقطه ی دیگر را نشان می دهند. از آنجایی که اطلاعات در مورد یک شخص، مکان یا چیزی است، بنابراین جریان اطلاعات باید توسط یک اسم شرح داده شود. مستطیل با گوشه های گرد برای نمایش پردازش اطلاعات استفاده می شود. پردازش اطلاعات باعث ایجاد تغییرات یا تبدیلات در اطلاعات ورودی می شود، بنابراین خروجی اطلاعات از یک پردازش باید با یک عنوان جدید، نامگذاری شود.

آخرین عنصر کاربردی در نمودارهای DFD، یک مستطیل با یک سر باز می باشد که برای نمایش مرکز ذخیره سازی اطلاعات استفاده می شود. همانند سایر عناصر DFD، این عنصر نیز باید توسط یک اسم مشخص نامگذاری شود. مراکز داده توسط یک کدگذای مشخص مانند ۱D، ۲D، ۳D و ... شماره گذاری می شوند.



بالاترین لایه نمودار را نمودار محتوا (context diagram) می نامند که بیانگر دید کلی از سیستم است.



یک نمودار محتوا برای سیستم پردازش حقوق

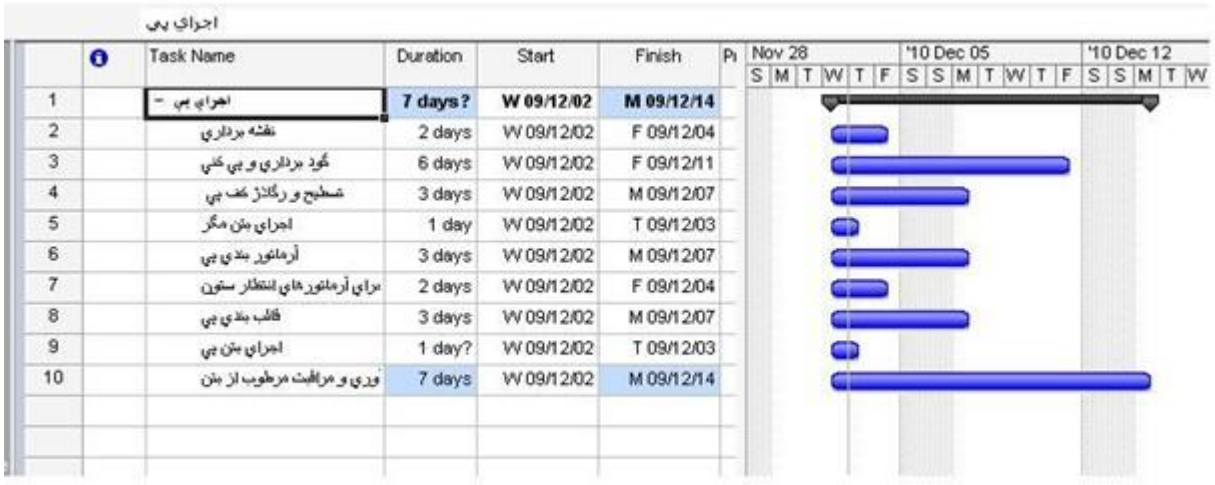
زمانبندی پروژه:

این فرایند کل کار پروژه را به فعالیت های جداگانه ای تقسیم می کند و زمان مورد نیاز برای کامل کردن این فعالیت ها را برآورد می کند.

نمودارهای گانت:

نمودار گانت نوعی نمودار میله‌ای است که برنامه زمان‌بندی پروژه را نشان می‌دهد. گانت چارت وسیله‌ای مفید برای برنامه‌ریزی و زمانبندی پروژه‌ها می‌باشد. در سمت چپ این نمودار لیستی از فعالیت ها و در رو به روی آن زمان مناسب برای

انجام هر فعالیت نمایش داده میشود. هر فعالیت با یک مستطیل نمایش داده میشود که از طول و جای قرار گرفتن این مستطیل تاریخ شروع ، پایان و زمانی که صرف آن فعالیت میشود استناد میگردد.



برنامه ریزی هزینه:

مدیران پروژه باید پروژه را به درستی و روشنی تعریف کرده و تخمین درستی از هزینه پروژه داشته باشند. طبقه بندی هزینه به دو گروه هزینه های محسوس و هزینه های نامحسوس تقسیم می شود. هزینه های محسوس آن دسته از هزینه های هستند که یک سازمان می تواند به سادگی آن را اندازه گیری کند. هزینه نامحسوس آن دسته از هزینه هایی است که اندازه گیری آنها با عبارت پولی مشکل است. (سطح مهارت، حسن نیت و...) و هزینه های مستقیم و غیرمستقیم.

مدیریت ریسک:

ریسک را می توان شرایط نامطلوبی دانست که واقعاً رخ می دهند. ریسک های پروژه ، نرم افزار در حال توسعه و سازمان را تهدید می کنند . این دسته از ریسک را می توان بصورت زیر تعریف کرد :

۱- ریسک های پروژه .
۲- ریسک های محصول .
۳- ریسک های کاری .

فرایند مدیریت ریسک شامل مراحل زیر است :

۱- شناسایی ریسک.
۲- تحلیل ریسک
۳- برنامه ریزی ریسک
۴- نظارت بر ریسک .

انواع ریسک:

ریسک های **انسانی** مثل مهارتهای ناکافی(مدیریتی و تکنیکی)، بی تجربگی عمومی و بی تجربگی در یک حوزه کاری خاص یا فناوری خاص. ریسک های **ساختاری** مانند تغییراتی که در یک پروژه جدید در حوزه کاری کاربر و رویه کاری کسب و کار

ایجاد خواهد کرد. ریسک **فناوری** مانند استفاده از فناوری جدید یا تجربه نشده. ریسک **زمانی** مانند اشتباه در تخمین زمان، ریسک **هزینه** مانند اشتباه در تخمین هزینه.

تحلیل نیازها:

وقتی اطلاعات مربوط به نیازهای موجود جمع آوری شد، مرحله تحلیل نیازها را شکل می دهد. در این مرحله، نیازمندیها دسته بندی شده و به صورت مجموعه ها و زیرمجموعه های مربوطه در می آیند. آنگاه هریک از نیازمندیها در ارتباط با دیگر نیازها بررسی می گردد آنها از نظر ثبات و سازگاری، قابلیت حذف شدن و عدم وجود ابهام سنجیده و مورد آزمون قرار می گیرند و سرانجام اولویت نیازها براساس خواسته ها و انتظارات کاربران/ خریداران تعیین می گردند.

مشتری و مهندس نرم افزار هردو نقش فعال در مهندسی نیازمندیهای نرم افزار ایفا می کنند. مشتری تلاش می کند تا شرح داده ها، کارکرد و رفتار سیستم گاهی اوقات مبهم را دوباره فرموله کند. توسعه دهنده به عنوان پرسشگر، مشاور، حلال مشکلات و مذاکره کننده عمل میکند. ابتدا تحلیل گر مشخصات سیستم را مطالعه می کند و بعد طرح پروژه نرم افزار را بررسی می کند. سپس برای تحلیل باید ارتباطات برقرار شود تا از فهم مشکل اطمینان حاصل شود. ارزیابی مشکل و یافتن راه حل، حوزه اصلی تلاش بعدی مربوط به تحلیل است. تحلیلگر باید تمام جنبه های ملموس داده های قابل مشاهده برونی را تعریف کرده، جریان و مضمون اطلاعات را ارزیابی کرده و کلیه کارکردهای نرم افزار را تعریف کند و در مورد آن به تفصیل بپردازد. رفتار نرم افزار را در بافت حوادثی که روی سیستم اثر می گذارد، درک نمایند. خصوصیات تعامل سیستم را تثبیت کرده و محدودیت های طراحی را علنی نماید.

انواع تست نرم افزار:

تست نرم افزار به فرایند ارزیابی یک نرم افزار به منظور تشخیص تفاوت بین خروجی کنونی(خروجی نرم افزار) و خروجی مورد انتظار گفته می شود. علاوه بر آن تست نرم افزار ارزیابی امکانات و ویژگی های یک نرم افزار را نیز شامل می شود. به عبارت دیگر تست نرم افزار یک فرایندی است که به وسیله آن می توانیم اطلاعاتی در رابطه با کیفیت نرم افزار بدست آوریم که شامل تایید و اعتبار سنجی است.

تایید: در این مرحله اطمینان حاصل می شود که آیا نرم افزار مورد نظر، با توجه به انتظار ما رفتار می کند یا رفتار دیگری از خود نشان می دهد.

اعتبار سنجی:

این مرحله مشخص کننده این موضوع است که آیا نرم افزار نیازمندی ها را پاسخ گو هست یا خیر. هر نرم افزاری برای انجام کار یا کارهای خاصی نوشته می شود که باید این کارها را درست و صحیح انجام دهد.

انواع رویکرد تست:

دو رویکرد برای تست نرم افزار داریم که هر کدام دارای انواع و مراحل خاصی می باشد که باید به ترتیب و در زمان خود انجام گیرد. این دو رویکرد تست جعبه سیاه (blackbox testing) و تست جعبه سفید (whitebox testing) می باشد.

تست جعبه سیاه: در این رویکرد ، تست تمامی مکانیسم های داخلی یک سیستم نادیده گرفته می شود و روی خروجی تولید شده تمرکز می شود. به این رویکرد تست functional نیز می گویند.

تست جعبه سفید:

در این نوع تست با مکانیستم داخلی و متدهای یک سیستم سرو کار داریم. به این نوع تست structural تست نیز گفته می شود.

انواع تست :

انواع مختلفی از تست وجود دارند که در ادامه آن ها را مشاهده می کنید:

- Unit Testing : در این نوع تست ما یک واحد و یا یک گروه از واحد های مرتبط با هم را تست می کنیم. این نوع تست زیر مجموعه تست جعبه سفید است.
- Integration Testing: این نوع تست به ما این امکان را می دهد که چند نوع کامپننت مختلف را کنار یکدیگر تست کنیم. در این صورت حتی ما می توانیم وابستگی های میان سخت افزار و نرم افزار را نیز تست کنیم. این دسته از تست ها زیر مجموعه تست جعبه سیاه هستند.
- Functional Testing: در این تست اطمینان حاصل می شود که عملکرد برنامه به درستی است. توجه شود که در این نوع تست برخلاف آزمون واحد ما می توانیم عملکرد یک سیستم را تست کنیم و نه فقط یک واحد را. این تست زیر مجموعه تست جعبه سیاه هستند.
- System Testing : این نوع تست به ما اجازه می دهد که از عملکرد برنامه در محیطهای مختلف اطمینان حاصل کنیم(مثل سیستم عامل های مختلف). این تست زیر مجموعه تست جعبه سیاه است.
- Stress Testing : این نوع تست عملکرد برنامه را در شرایط نامطلوب مورد بررسی قرار می دهد. این تست زیر مجموعه تست جعبه سیاه است.
- Performance Testing : تست عملکرد و کارایی که در مجموعه تست جعبه سیاه جای میگیرد به ما این اطمینان را می دهد که برنامه مان عملکرد و کارایی لازم را در یک مدت زمان مشخص داراست.
- Usability Testing: این تست از دیدگاه مشتری انجام می شود و در واقع مشخص کننده فاکتورهای زیر است:
- آیا برنامه کاربر پسند است؟ آیا برنامه ساده و قابل یادگیری است؟ آیا برنامه جذاب طراحی شده است؟ و... این نوع تست زیر مجموعه تست جعبه سیاه است.
- Acceptance Testing: این نوع تست معمولا از طرف مشتری انجام می شود. هدف آن، مشخص کردن این است که آیا برنامه نیازهای مشتری را پاسخ می دهد و آیا برنامه همان چیزی که مشتری می خواهد هست یا خیر.
- Regression Testing : این نوع تست به منظور صحت عملکرد سیستم بعد از تغییرات استفاده می شود و زیر مجموعه تست جعبه سیاه است.
- Beta Testing :تستی است که توسط کاربر نهایی یا یک تیم خارج از تیم توسعه انجام می شود. هدف تست بتا پوشش دادن خطاهای غیر منتظره است. این تست زیر مجموعه تست جعبه سیاه است.

مراحل تست نرم افزار:

۱. تست واحد: در این نوع تست قسمت های کوچک برنامه را تست می کنیم مثل توابع یا کلاس ها. در این تست با کل برنامه کار نداریم و هدف ما اطمینان از کارکرد قسمت های کوچک برنامه است.
۲. تست جامعیت: در این تست برای اینکه مطمئن شویم کامپوننت های مختلف برنامه با هم درست کار می کنند.

۳. تست سیستم: در این نوع تست تمام سیستم تست می شود و حتی خود نرم افزار و سخت افزار و ارتباطات بین کامپوننت ها مورد توجه قرار می گیرد. تست سیستم شامل تست گرافیک رابط کاربری، تست کارایی و تست نصب می باشد.

رابط کاربر نرم افزار:

رابط کاربر محیطی است در نظام های کامپیوتری اعم از سایت ها، پایگاه داده، نرم افزار و... که بین این نظام ها وکاربر تعامل ایجاد می کند. یعنی باعث انتقال اطلاعات از کاربر به نظام و بالعکس می شود و با بازخورد همراه است. امکان تعامل دو سویه میان سیستم و کاربر از طریق رابط کاربر برقرار می شود و باعث رضایتمندی کاربر از سیستم می شود. بدون وجود یک رابط خوب پایگاه ها، نظام های اطلاعاتی از کارایی لازم برخوردار نخواهد شد. در نتیجه به کاربر نهایی امکان می دهد تا به سیستم و محتوای آن دسترسی داشته باشد. به مدیران و کاربران سیستمی نیز امکان می دهد تا مجموعه را کنترل کنند.

مراحل طراحی رابط کاربر:

طراحی رابط کاربر با شناسایی کاربر، وظیفه و نیازمندیهای محیطی آغاز می شود. پس از شناسایی وظایف کاربر، سناریوهای کاربر به منظور تعیین مقاصد و اعمال رابط ایجاد شده و مورد بررسی و تحلیل قرار می گیرد. این مراحل مبنای ایجاد آرایش صفحه نمایش قرار می گیرند که این صفحه آرایی طرح گرافیکی و جاگذاری شمایل ها، شرح متن توصیفی صفحه نمایش، مشخصات و عنوان گذاری پنجره ها و مشخصات موارد اصلی و فرعی گزینشی را به نمایش می گذارد. برای الگوسازی و نهایتا اجرای مدل طراحی، ابزارهایی مورد استفاده قرار می گیرند و نتیجه از لحاظ کیفی ارزیابی می شود.

دیدگاه های طراحی رابط کاربر در سیستم های اطلاعاتی:

۱. دیدگاه کاربرمدار: تاکید بر کاربر نهایی است.
۲. دیدگاه مهندسی (برنامه نویس): اصل در طراحی به عهده برنامه نویس است.
۳. دیدگاه طراح: به عنوان واسط بین کاربر و برنامه نویس عمل می کند.

زبان مدلسازی یکپارچه(UML)

UML یک زبان مدلسازی است و روشی برای توصیف ویژگی ها، نمایش گرافیکی، ساختن و مستندسازی اجزای یک سیستم نرم افزاری در حال توسعه می باشد. از UML برای فهمیدن، طراحی، مرور، پیکربندی، نگهداری و کنترل اطلاعات سیستم های نرم افزاری استفاده می شود. با استفاده از UML می توان تقریبا هر گونه برنامه کاربردی که ممکن است بر روی هر ترکیبی از سخت افزار، سیستم عامل، زبان برنامه نویسی و شبکه اجرا می شود را الگوسازی نمود. UML شامل ۹ نمودار پایه است:

۱- نمودار کلاس
۲- نمودار شیء
۳- نمودار مورد کاربرد
۴- نمودار تعامل
۵- نمودار همکاری
۶- نمودار حالت
۷- نمودار فعالیت
۸- نمودار اجزاء
۹- نمودار استقرار.

نمودار کلاس:

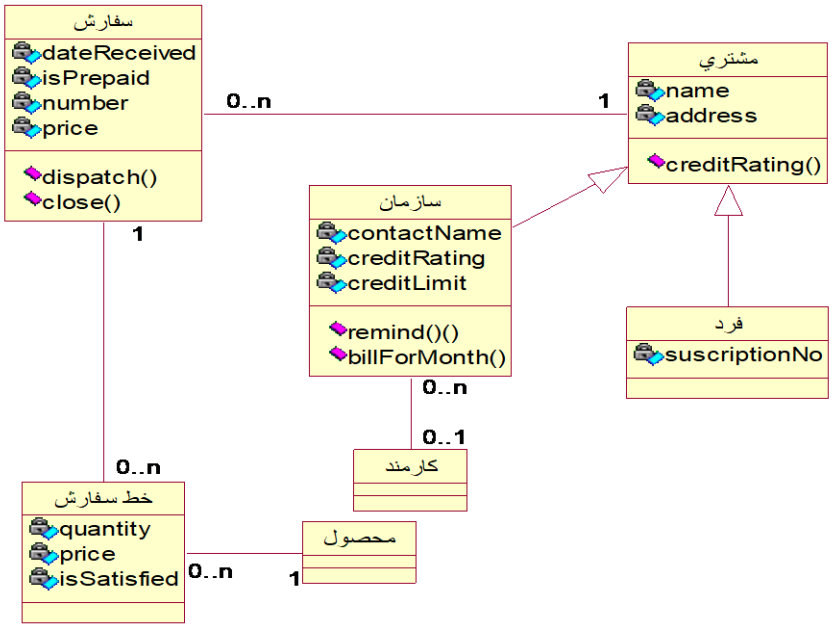
نمودار کلاس شامل کلاسهای متعددی است که با ارتباطاتی متصل است. مستطیل یک شمایل برای کلاس است . نام کلاس در بالای مستطیل نوشته میشود.

صفت یک ویژگی کلاس است. این مورد دامنه ای از مقادیری را که یک ویژگی باید در اشیاء کلاس نگه دارد را تشریح می کند. یک کلاس ممکن است صفر یا تعداد بیشتری صفت داشته باشد. فهرستی از نام صفات در زیر خطی که آنها را از نام کلاس جدا میکند آورده می شود. در شمایل کلاس ، برای هر مقدار صفت میتوانید یک نوع مشخص نمایید رشته ای، اعشاری، عدد، عدد صحیح، منطقی یا نوعی که خود کاربر تعریف نموده باشد (همچنین شما میتوانید برای یک صفت یک مقدار پیش فرض نشان دهید.)

یک عملیات چیزی است که یک کلاس میتواند انجام دهد یا شما (یا کلاسهای دیگر) میتوانید در ارتباط با یک کلاس انجام دهید. فهرست عملیات در زیر خطی که صفات را از عملیات جدا میکند شروع میشوند.

هنگامیکه کلاسها بطور مفهومی به یکدیگر مرتبط باشند، با یک خط دو کلاس با هم متصل می شوند. چندی ارتباط تعداد اشیائی که از یک کلاس به تعداد اشیائی در کلاس وابسته دیگر مرتبط هستند را نشان میدهد. یک کلاس میتواند به صورتهای زیر به کلاسهای دیگر مرتبط شود: یک به یک، یک به چند، چند به چند

نام کلاس
صفات کلاس
عملیات کلاس

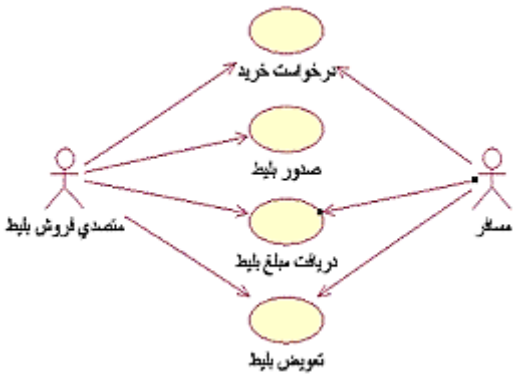


نمودارهای مورد کاربرد[1]

نمودارهای کلاس یک دید استاتیک از کلاسها در یک سیستم فراهم می کنند. نمودار کلاس یک راه مناسب جهت تحریک مشتری برای صحبت کردن درباره یک سیستم از نقطه نظر خود او میباشد، مورد کاربرد یک ابزار بسیار عالی برای تحریک کاربران بالقوه جهت صحبت درباره سیستم از دید آنهاست. همیشه برای کاربران آسان نیست که بطور مفصل درباره اینکه چگونه روش استفاده از یک سیستم را دریافتند صحبت کنند این یک حقیقت است که کاربران بیش از آنچه صحبت میکنند، میدانند: مدل کاربرد کمک میکند تا این حصار شکسته شود.

مورد کاربرد مجموعه ای از سناریوها درباره کاربرد سیستم است. هر سناریو یک توالی از رخدادها را شرح میدهد. هر توالی بوسیله یک شخص ، یک سیستم دیگر، یک قطعه سخت افزار و یا با گذر زمان آغاز شده است. موجودیتهایی که یک توالی را آغاز میکنندactor نامیده میشوند. یک actor یا بازیگر یک مورد کاربرد را آغاز میکند. و در نمودار با علامت آدمک نشان داده می شود. مورد کاربرد با بیضی نشان داده می شود. خط رابط یک مورد کاربرد را به یک بازیگر متصل می کند. هر مورد کاربرد فهرستی از سناریوها است و هر سناریو توالی از مراحل است.

نمودارهای مدل کاربرد- مثال



[1] Use case diagram

نمودارهای توالی

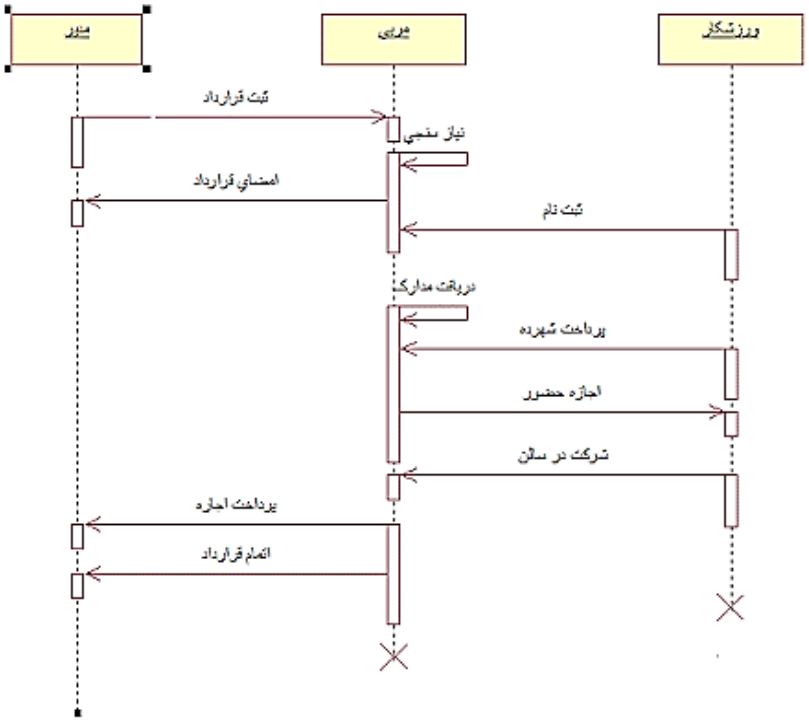
این نمودار از جمله نمودارهای تبادلی است و برای نمایش اتباط بین بازیگرها و اشیاء مختلف در یک توالی زمانی و مبادله پیام بین آنها استفاده می شود. در هر نمودار توالی باید حداقل یک بازیگر وجود داشته باشد که عملیات را راه اندازی کند. اجزاء این نمودار عبارتند از:

بازیگر(actor): همان که در نمودار مورد کاربرد گفته شد.

خط زندگی (life line): نشان دهنده یک شی در نمودار است. یک شیء می تواند نمونه ای از یکی از کلاسهایی که در نمودار کلاس طراحی شده، باشد. همراه خط زندگی یک مستطیل وجود دارد که فعالیت نامیده می شود طول این مستطیل، طول مدت فعالیت است.

پیام (message): ارتباط بین بازیگر و یک شیء و یا ارتباط بین دو شیء را بیان می کند و با یک پیکان جهت دار نشان داده می شوند.

پیام به خود (self-message): ارتباط بین یک شیء با خودش را بیان می کند و به صورت زیر نمایش داده می شود.



منابع:

۱. کتاب مهندسی نرم افزار: رهیافتی برای یک اهل فن ، راجر پرسمن، مترجم: نوید هاشمی طبا
۲. کتاب مهندسی نرم افزار، یان سامرویل، مترجم: عین الله جعفرنژاد قمی
۳. خودآموز UML در شش روز
۴. معرفی زبان مدلسازی UML
۵. نمودار جریان داده ها