مهندسی نرم افزار

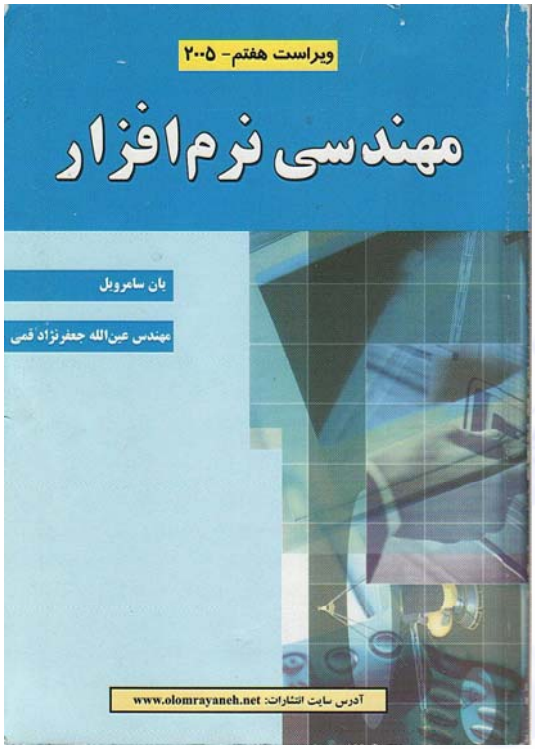
فصل اول: نرم افزار و مهندسی نرم افزار

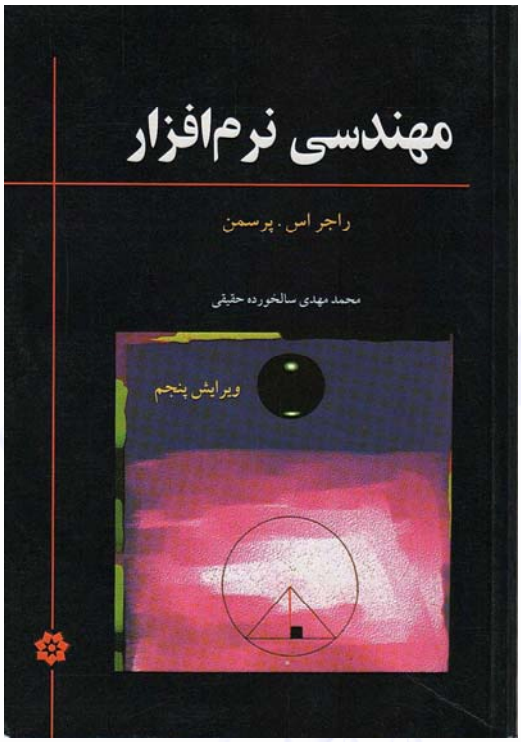
مدرس: اسماعیل نورانی

معرفی منابع:

- مهندسی نرم افزار
  اثر: راجر اس. پرسمن مترجم: محمد مهدی سالخورده حقیقی

- مهندسی نرم افزار
  اثر: یان سامرویل مترجم: عین الله جعفر نژاد قمی





مقدمه

- نرم افزار کامپیوتر محصولی است که :
  - توسط مهندسان نرم افزار طراحی و ایجاد میشود
  - می تواند در اندازه ها و معماریهای متنوعی باشد
  - شامل مستنداتی هستند که عمل برنامه و چگونگی استفاده از آن را شرح می دهند.

مقدمه

- برای اولین بار در سال ۱۹۶۸ در یک کنفرانس در کشور آلمان بر لزوم مهندسی این دستاورد جدید بشر یعنی نرم افزار، تاکید شد.

- امروزه با گسترش تکنیک های مهندسی، ابزارها و دانش و تجربه، صنعت نرم افزار به یکی از صنایع برتر جهانی تبدیل شده است.

مهندسی نرم افزار

- تعریف:
  - شاخه ای است از مهندسی که با بهره گیری از دانش علمی به ارائه راه حل هایی مقرون به صرفه در قالب دستاوردهای نرم افزاری و به منظور حل مسائل و مشکلات عملی و خدمت به جامعه بشری اقدام می نماید.

- تعریف:
  - فرآیندی است که طراحی، ساخت و نگهداری یک نرم افزار قابل اطمینان را در بر می گیرد.

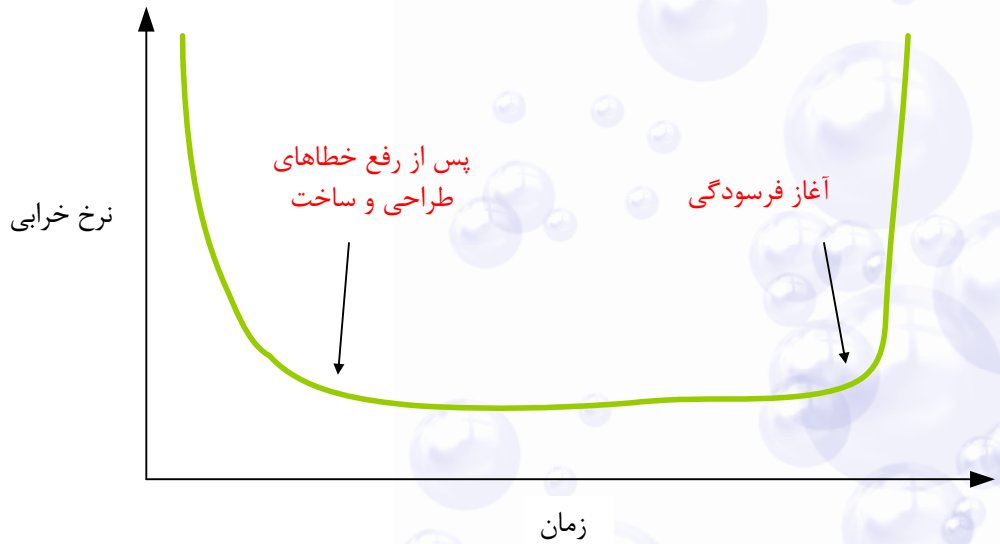
تفاوت نرم افزار و سخت افزار

۱ فرایند تولید نرم افزار یک فرایند مهندسی است نه یک فرایند تولید صنعتی

	سخت افزار	نرم افزار
تعیین مشخصات:	بوسیله متخصصین	بوسیله استفاده کننده گان نهائی
ماهیت فرایند تولید:	مکانیکی	فرایند مهندسی که برای هر کاربرد جدید منحصر به فرد است
ماهیت محصول:	فیزیکی	منطقی

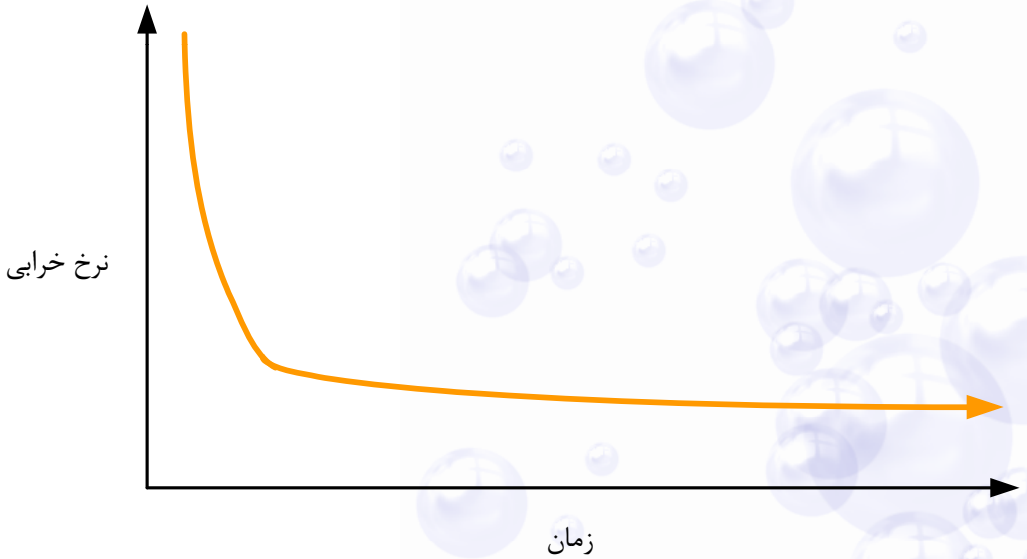
تفاوت نرم افزار و سخت افزار (ادامه)

۲ نرم افزار با گذشت زمان دچار فرسودگی نشده بلکه **فاسد** می گردد



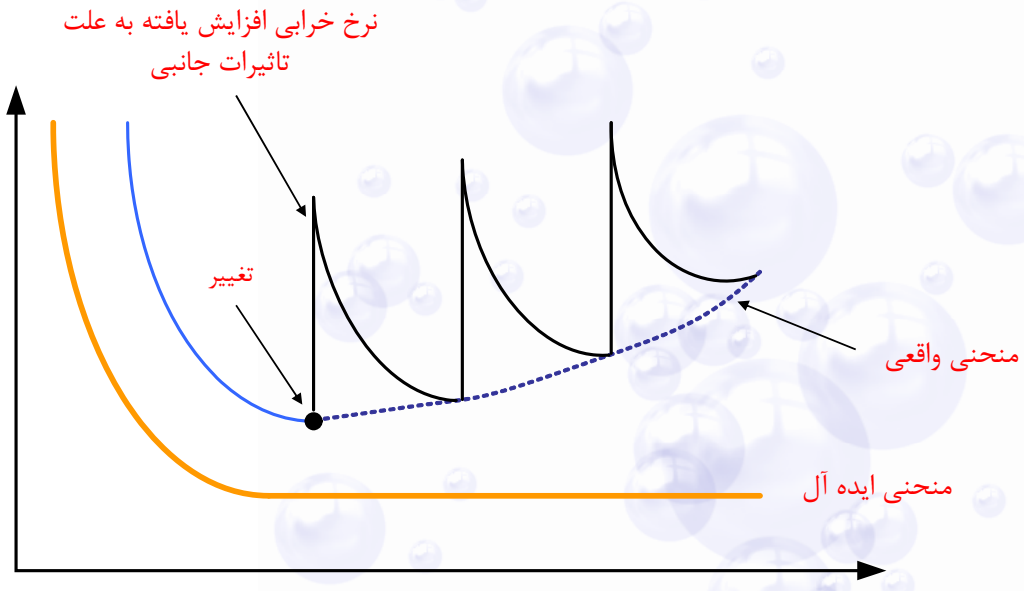
منحنی نرخ خرابی سخت افزار نسبت به زمان

تفاوت نرم افزار و سخت افزار (ادامه)



منحنی نرخ خرابی **ایده آل** نرم افزار نسبت به زمان

تفاوت نرم افزار و سخت افزار (ادامه)



منحنی نرخ خرابی **واقعی** نرم افزار نسبت به زمان

مراحل تولید نرم افزار

- ۱- شناخت و تجزیه و تحلیل صورت مساله
- ۲- طراحی و مدلسازی
- ۳- پیاده سازی
- ۴- تست و آزمایش
- ۵- نگهدار ی

چرخه حیات (life cycle) توسعه نرم افزار

- مراحل مختلف توسعه نرم افزار : شناخت و تجزیه تحلیل ، طراحی ، پیاده سازی، تست و نگهداری می باشد.
- Life cycle یک نرم افزار شامل تمام فعالیت های پیش از تولید ، توسعه و دوره اجرای نرم افزار است.
- شروع پروژه، RFP (Request For Proposal) به تیم نرم افزاری ارائه میشود و تیم توسعه برای دریافت پروژه پیشنهاد خود را در قالب Proposal ارائه میکند.

بحران نرم افزاری :

- پیچیدگی و قدرت سخت افزار باعث شده که نوشتن نرم افزاری که بتواند از این پتانسیل سخت افزاری استفاده کامل کند مشکل گردد.
- توان ما برای ایجاد برنامه های جدید به اندازه تقاضا نیست
- توان نگهداری برنامه های موجود به خاطر طراحی ضعیف مورد تهدید قرار می گیرد.
- برنامه های تولید شده مشکلاتی دارند .
  - هزینه تولید نرم افزار دائماً افزایش می یابد.
  - هزینه ها مطابق هزینه پیش بینی شده نیستند.
  - عملیات مطابق برنامه زمان بندی انجام نمی گیرند
  - نرم افزار تولید شده همه کارهای مورد نظر را انجام نمی دهد
  - نرم افزار دارای خطاست

تکامل نرم افزار

- دوره اول
  - دوره کامپیوترهای بزرگ (Mainframe)
  - پردازش دسته ای (batch)
  - نرم افزارهای سفارشی و مخصوص
- دوره دوم:
  - کامپیوترهای اشتراک زمانی و چندین کاربری به صورت تعاملی (Interactive)
  - ظهور سیستم های بلادرنگ (Real Time)
  - پایگاه داده ها
  - نرم افزارهای از قبل تولید شده

تکامل نرم افزار (ادامه)

- دوره سوم
  - سیستم های توزیع شده (Distributed)
  - سخت افزار ارزان
- دوره چهارم:
  - سیستم های خبره (Expert Systems)
  - ماشین های هوش مصنوعی
  - معماری موازی
  - روش های شی گرا

تکامل نرم افزار (ادامه)

- **نکته**: تکامل صورت گرفته در اثرکاهش هزینه سیستم های کامپیوتری بوده و به آن منتهی شده که جوامع را از یک جامعه صنعتی به یک جامعه اطلاعاتی تبدیل کرده است.
- نکته: پیچیدگی و قدرت سخت افزار باعث شده است نوشتن نرم افزاری که بتواند به صورت کامل از پتانسیل سخت افزاری استفاده کند مشکل گردد.

انواع و کاربردهای نرم افزار

- ۱- نرم افزارهای سیستمی
  - مجموعه ای از برنامه ها برای سر و سامان دادن به برنامه های دیگر
  - مثال: کامپایلرها، برنامه های مدیریت فایل و یا سیستم عامل ها
  - فعل و انفعال زیاد با سخت افزار
- ۲- نرم افزارهای بلادرنگ (Real Time)
  - مدت زمان انتظار برای پاسخ سیستم، تضمین شده و مشخص می باشد.
  - مثل سیستم هایی که وقایع را اندازه گیری، تجزیه تحلیل و کنترل می کنند.

انواع و کاربردهای نرم افزار (ادامه)

- ۳- نرم افزارهای تجاری (Business Software)
  - سیستم هایی که برای مشاغل خاص طراحی شده اند و برای مکانیزه کردن آنها و حل مسائل و مشکلات یک حرفه خاص می باشندو
  - مثال: حقوق دستمزد، حسابداری،انبارداری و .....
- ۴- نرم افزارهای مهندسی یا علمی
  - مثال: CAD، شبیه سازی، سیستم زمین شناسی، Proteus (برق)

انواع و کاربردهای نرم افزار (ادامه)

- ۵- نرم افزارهای توکار – جاسازی شده (Embedded software)
  - نرم افزارهای موجود روی محصولات هوشمند الکترونیکی
  - مثل: سیستم های موجود در خودروها، تلفن ها، سیستم های امنیتی و یا هواپیما

- ۶- نرم افزارهای کامپیوترهای شخصی
  - مثال: واژه پردازها، گرافیک کامپیوتری، برنامه های تفریحی و .....

- ۷- نرم افزارهای هوش مصنوعی
  - مثال: سیستم های خبره، تشخیص الگو و ...

معیارهای ارزیابی نرم افزار

- ۱- قابلیت نگهداری و توسعه (Maintainability & Extendibility)
  - نرم افزار باید بتواند با تغییرات نیازهای کاربران تکامل و توسعه یابد و نیازهای جدید آنها را رفع نماید.

- ۲- قابلیت اطمینان (Reliability)
  - احتمال کارکرد صحیح نرم افزار بالا بوده و خرابی نداشته و یا قابل اغماض باشد.

معیارهای ارزیابی نرم افزار (ادامه)

- ۳- قابلیت استفاده مجدد (Reusability)
  - قابلیت استفاده بخش هایی از کد در تولید نرم افزارهای دیگر بدون تغییر و یا با کمترین تغییر

- ۴- کارایی (Efficiency)
  - استفاده کارا و بهینه از منابع سیستم

معیارهای ارزیابی نرم افزار(ادامه)

- ۵- قابلیت پذیرش (Acceptability)
  - نرم افزار باید از طرف کاربرش قابل فهم، مفید و سازگار با سایر سیستم ها باشد

- ۶- قابلیت حمل (Portability)
  - قابلیت اجرا روی سکو (Platform)های مختلف
- Platform : به ترکیب خاصی از معماریهای سخت افزار و یا چارچوبهای نرم افزاری که برای اجرای نرم افزار خاصی لازم است سکو گفته می شود.
  - مثال برای سکوهای سیستم عاملی: لینوکس، ویندوز، سولاریس و ...
  - مثال برای سکوهای سیستم نرم افزاری: Java ، .Net ،
  - مثال برای سکوهای سخت افزاری: x86 ، Macintosh

تعریف متدولوژی

- یک متدولوژی مجموعه ای از روش ها و توصیه ها (guidelines) می باشد که به همراه راهبرد مشخص و طی مراحل مختلف در توسعه سیستم به کار گرفته می شود.
- یک متدلوژی دارای ابزار تعریف شده و از یک مدل مفهومی می باشد و از گرامر مشخص استفاده می کند.

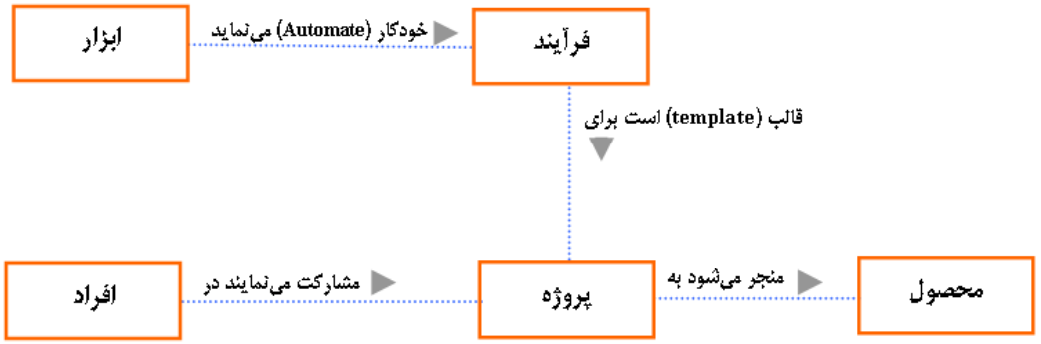
- برای مثال مدل شی گرا و یا مدل ساخت یافته در توسعه نرم افزار دو متدولوژی توسعه نرم افزار هستند.

فرآیند نرم افزار

- الگو و قالبی که چگونگی طیِ مراحل مختلفِ یک پروژه را تعریف می نماید فرآیند نرم افزار نامیده می شود.

- یک فرایند تولید، به ما میگوید که برای دستیابی به هدف مطلوب، چه کسی، چه کاری را،چه موقع، و چگونه باید انجام دهد.

مدل ارتباط میان مفاهیم پروژه، فرایند، فراورده (محصول)، ابزار، و افراد



- افراد شامل تولید کنندگان و استفاده کنندگان
- ابزار: CASE (Computer Aided Software Engineering)
  - Rational Rose برای مدل سازی
  - VS.NET و Eclipse برای برنامه نویسی

تعریف مدل فرآیند

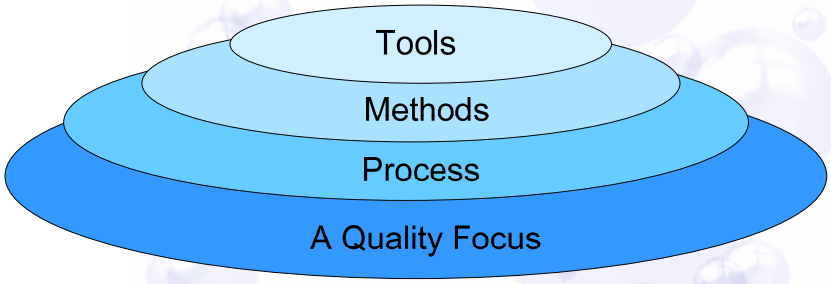
- تعیین یک راهکار توسعه نرم افزار را الگوی مهندسی نرم افزار یا مدل فرآیند نرم افزار می گویند.

- مدل فرآیند نرم افزار قدم ها و استراتژی توسعه نرم افزار می باشد.

- اهداف فرایند
  - ایجاد بستری جهت انجام روشهای فنی ، تولید محصولات کاری (مدلها، مستندات ، گزارشها ، فرمها ، داده ها و غیره) ، مشخص نمودن مراحل ، حصول اطمینان از کیفیت و مدیریت خوب تغییرات

تفاوت های متدولوژی و مدل فرآیند

- متدولوژی، روش طی کردن قدم هایی است که مدل فرآیند تعریف می کند.
- تکنولوژی مهندسی نرم افزار یک تکنولوژی لایه ای است و متدولوژی بروی لایه فرآیند قرار دارد.



فعالیتهای مهندسی نرم افزار

- بطور کلی فعالیتهای مربوط به مهندسی نرم افزار در سه فاز زیر دسته بندی می گردد:
  - فاز تعریف
  - فاز توسعه
  - فاز پشتیبانی

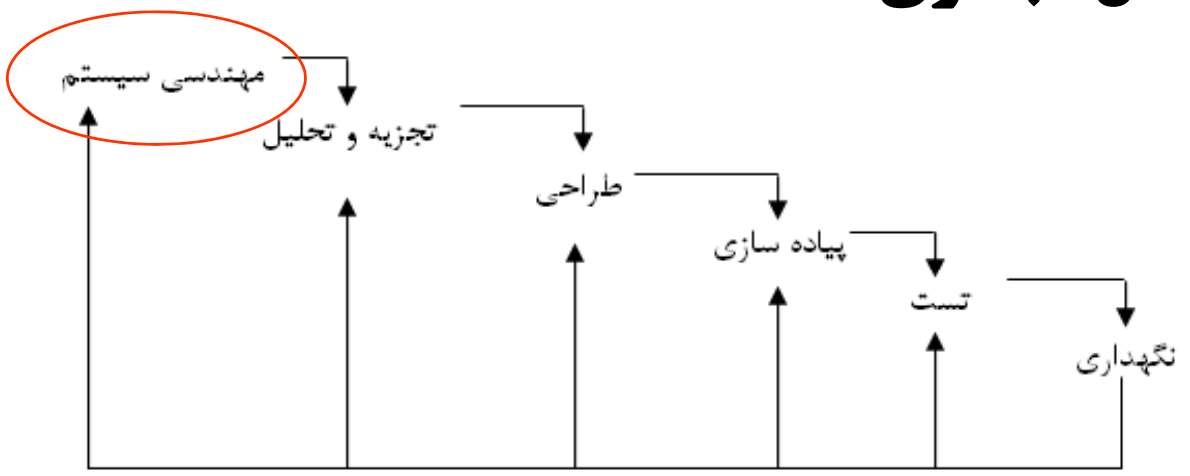
- فازهای فوق با یکسری فعالیتهای چتری (Umbrella activities) تکمیل می گردد. مهمترین آنها عبارتند از:

کنترل و ردیابی نمودن پروژه های نرم افزاری ، تضمین کیفیت نرم افزار ، تهیه مستندات ، مدیریت قابلیت استفاده مجدد ، سنجش و مدیریت ریسک

مدلهای فرآیند نرم افزار

- مدلهای فرایند نرم افزار عبارتند از :
  - مدل ترتیبی خطی (آبشاری)
  - مدل ایجاد نمونه اولیه (Prototyping Model)
  - مدل RAD
  - مدل افزایشی (Incremental Model)
  - مدل حلزونی(Spiral Model)
  - مدل حلزونی برنده - برنده (Win-Win)
  - مدل توسعه همزمان
  - مدل توسعه مبتنی بر مولفه (Component Base Development)
  - مدل روشهای رسمی (Formal Method)
  - تکنیکهای نسل چهارم

مدل آبشاری

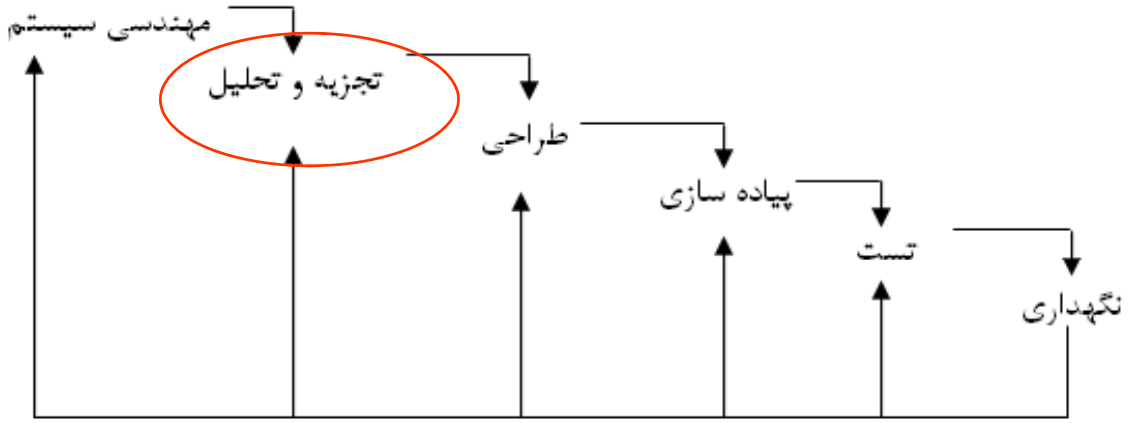


مهندسی سیستم:

از آنجاییکه نرم افزار همیشه قسمتی از یک سیستم بزرگتر است، کار از مشخص کردن نیازمندیهای کل سیستم آغاز می شودو سپس زیرمجموعه ای از این نیازمندیها را به نرم افزار نسبت می دهیم.

به خاطر اینکه نرم افزار مجبور به داشتن ارتباط با مولفه های دیگر سیستم از قبیل سخت افزار، مردم و پایگاه داده هاست، داشتن این دیدگاه از سیستم، یک امر اساسی است .

مدل آبشاری

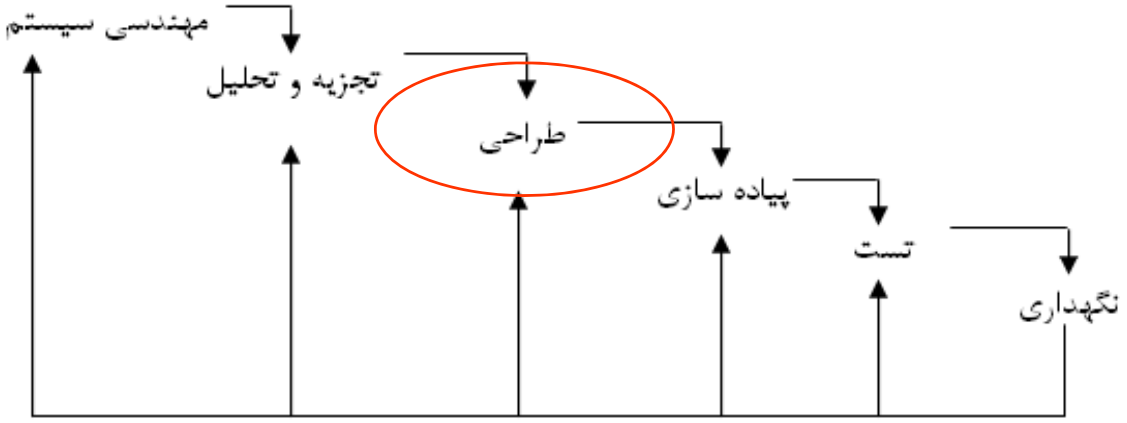


جمع آوری نیازمندیها مشخصا مربوط به نرم افزار برای فهمیدن چگونگی برنامه هایی که باید ساخته شوند. مهندس نرم افزار (تحلیلگر) بایستی

- دامنه اطلاعات
- عملیات مورد نظر (ورودیها- خروجیها و پردازشها)
- واسط ها

را بداند. نیازمندیهای سیستم و نرم افزار مستندسازی شده و با مشتری بازنگری می شود.

مدل آبشاری

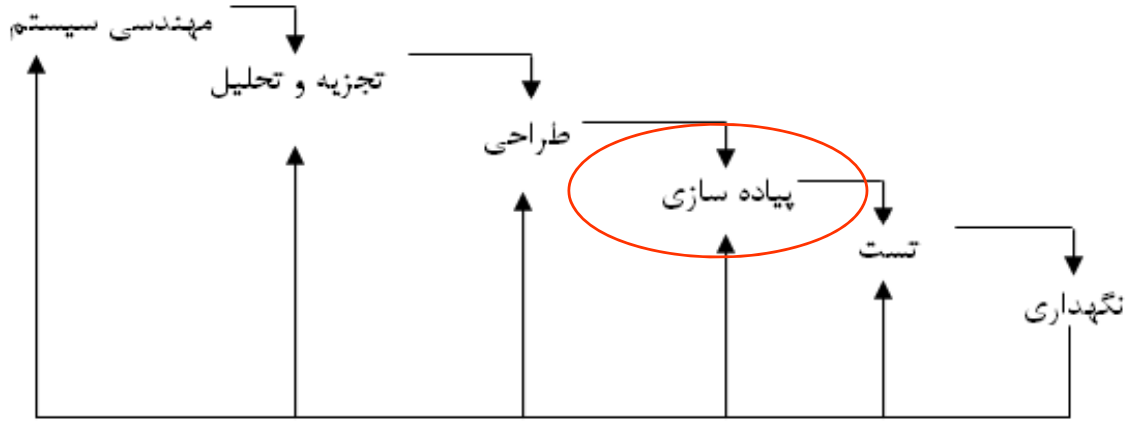


طراحی نرم افزار یک فرایند چند مرحله ای است که روی سه مشخصه متفاوت از برنامه تاکید دارد:

- ساختمان داده ها
- معماری نرم افزار
- جزئیات رویه ها

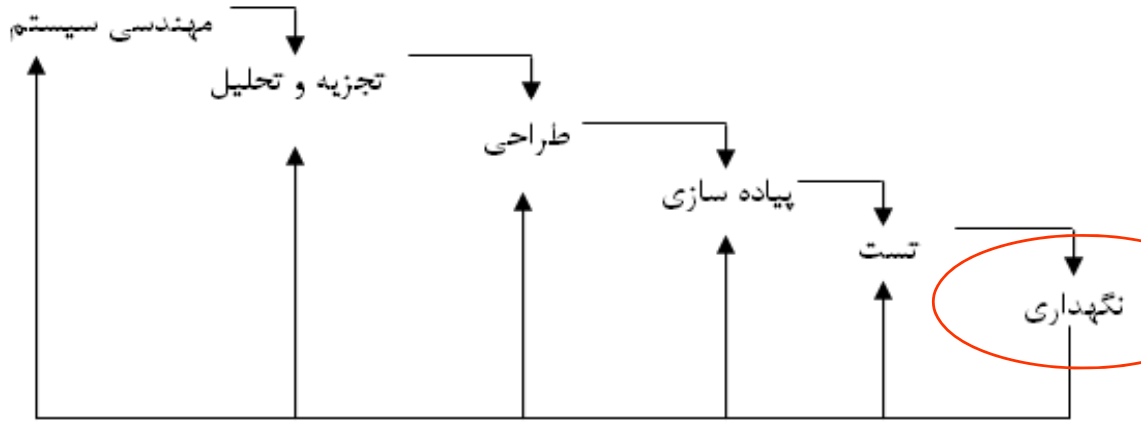
در فرایند طراحی نیازمندیها تبدیل به نمایشی از نرم افزار می شوند تا قبل از به کد درآوردن، قابل ارزیابی باشند. طراحی نیز بایستی مانند نیازمندیها مستندسازی گردد.

مدل آبشاری



طراحی بایستی به صورتی که برای ماشین قابل فهم باشد در بیاید. اگر طراحی در حد جزئیات باشد، پیاده سازی می تواند بسیار سریع و به صورت مکانیزه انجام شود.

مدل آبشاری



نرم افزار بدون شک پس از تحویل به مشتری دچار تغییر می شود . تغییرات به علت

- خطاها
- تطبیق نرم افزار با تغییرات محیط بیرونی
- درخواست مشتری جهت توسعه در عملیات یا بالا بردن کارایی

نگهداری نرم افزار تمام مراحل قبل را روی یک برنامه موجود اجرا می کند.

ایرادات مدل آبشاری

۱- جریان ترتیبی به ندرت وجود دارد. همیشه برگشت به عقب هست.

۲- بیان صریح همه نیازمندیها در ابتدا مشکل است.

۳- برنامه ها دیر حاضر می شود. مشتری باید صبر داشته باشد.

مدل ساخت نمونه اولیه (Prototyping)

نمونه سازی فرایندی است که تولید کننده را قادر به ایجاد یک مدل از نرم افزار مورد نظر می کند. این مدل می تواند به صورت

۱) یک نمونه روی کاغذ ***paper prototype***

۲) یک نمونه کاری ***working prototype***

۳) یک برنامه موجود ***existing prototype***

نمونه باید دور انداخته شود

مدل ساخت نمونه اولیه (Prototyping)

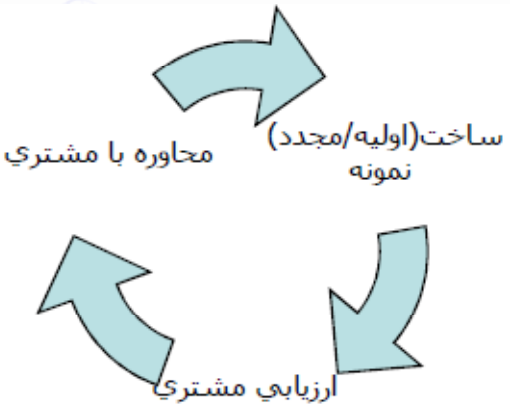
- این روش با جمع آوری نیازمندیها شروع می شود.
- طی یک طراحی سریع نمونه اولیه ای آماده می شود که فقط تاکید بر بخشهایی چون روشهای ورود اطلاعات و فرمت خروجی دارد.
- نمونه توسط مشتری ارزیابی شده و اصلاحات لازم در تکرار بعدی اعمال می شود.

مدل ساخت نمونه اولیه (Prototyping)

طراحی سریع

منجر به ساخت یک نمونه اولیه می شود

به عنوان راهکاری جهت تشخیص نیازمندیهای نرم افزار عمل می نماید



مدل RAD

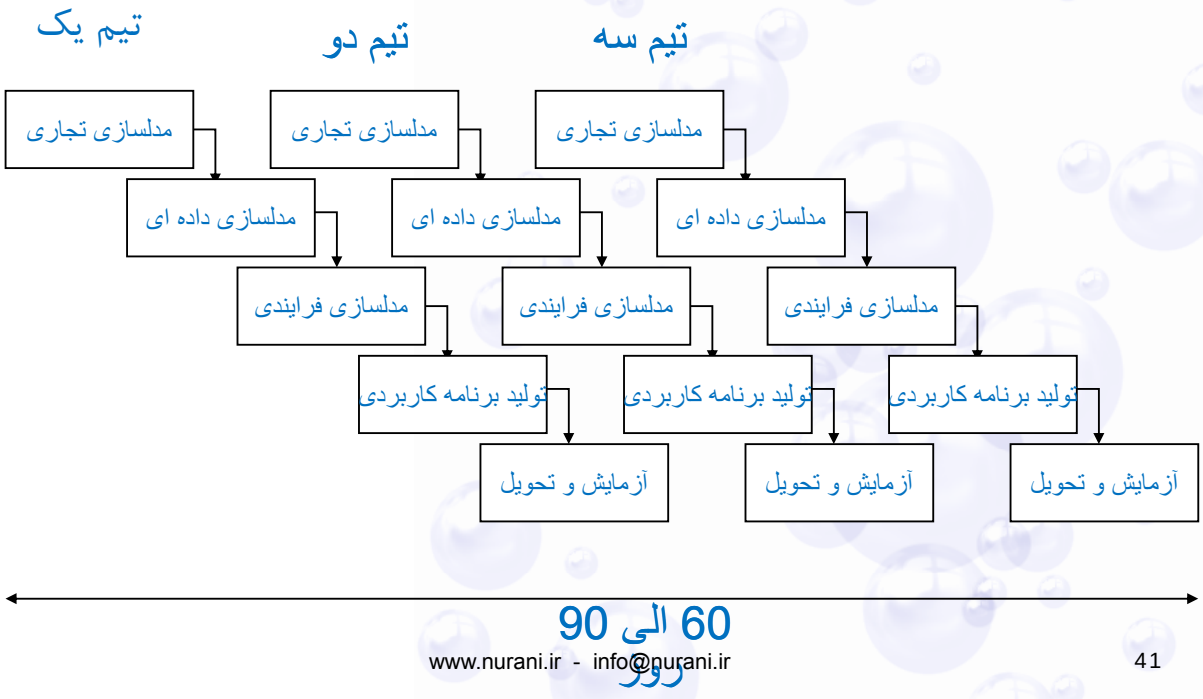
- مدل توسعه کاربردی سریع (Rapid Application Development) بر ساخت مبتنی بر مولفه تاکید می نماید .
- فازهای مدل مذکور عبارتند از:

  - مدلسازی کاری : جریان اطلاعات بین واحدهای عملکردی مدل می شود
  - مدلسازی داده ای : اشیا داده ای ، صفات و ارتباطات مشخص می شود.
  - مدلسازی فرایند: پردازش ها و فرآیند های روی اشیا داده ای تعریف می شوند
  - تولید برنامه کاربردی: بیشتر از قطعات کد از قبل آماده شده استفاده می شود.
  - تست

مدل RAD

- چنانچه بتوان سیستم را به چند زیر سیستم تقسیم نمود به گونه ای که توسعه هر زیر سیستم حداکثر ظرف مدت سه ماه به اتمام برسد ، می توان هر زیر سیستم را به یک تیم RAD سپرد و در انتها تمام زیر سیستمها را یکپارچه نمود.

مدل RAD (ادامه)



مدل RAD

- **موارد نامناسب برای استفاده از RAD:**
  - مواردی که نیاز به کارایی بالا می باشد
  - مواردی که امکان واحد بندی مناسب سیستم نباشد

- **مشکلات:**
  - نیاز به نیروی انسانی کافی (مخصوصا در پروژه های بزرگ برای تشکیل تیم های RAD)
  - تعهد تولید کننده و مشتری برای کامل کردن سیستم در زمان کوتاه لازم است

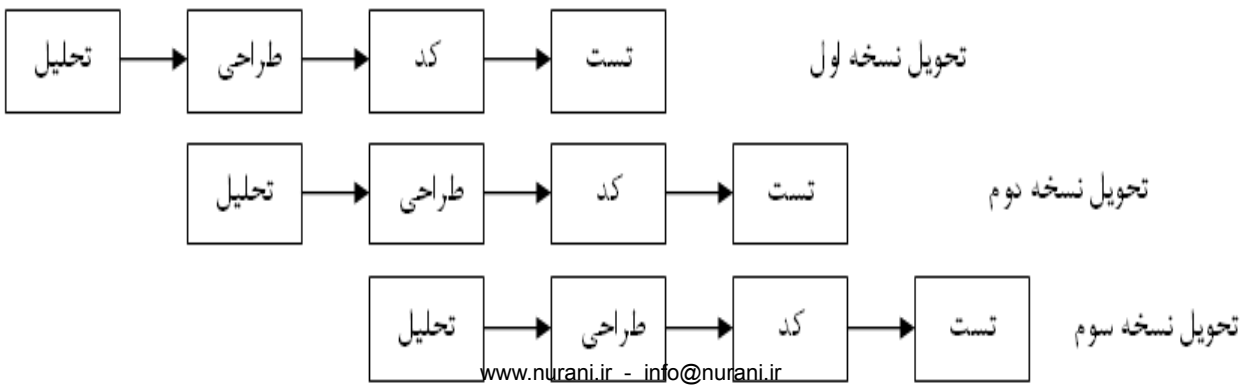
مدلهای تکاملی

- در ادامه سه مدل زیر معرفی میکنیم که در دسته فرآیندهای تکاملی قرار دارند و مبنای کار آنها نمونه ساده اولیه ای است که در طول زمان آن را توسعه و تکامل می دهند

  - مدل افزایشی
  - مدل حلزونی
  - مدل حلزونی win-win

مدل افزایشی

- ترکیب مدل خطی و مدل ساخت نمونه اولیه
- در انتهای هر ترتیب خطی یک محصول از نرم افزار ارائه می گردد.اولین محصول با نام محصول هسته ای (Core Product) به نیازمندیهای پایه ای پرداخته و پس از بازنگری توسط کاربر اصلاح و بهینه می گردد.



تفاوت مدل افزایشی با نمونه سازی

- در مدل افزایشی در انتهای هر مرحله افزایش، یک سیستم قابل تحویل به مشتری داریم ولی در روش نمونه سازی، فرآیند تکراری تنها در مرحله جمع آوری نیازمندیها انجام می شود.

مدل حلزونی(Spiral Model)

- این مدل نیز همانند مدل قبل از ترکیب دو روش زیر استفاده می کند:
- ماهیت تکراری مدل ساخت نمونه اولیه + ماهیت سیستماتیک مدل ترتیبی

مدل حلزونی

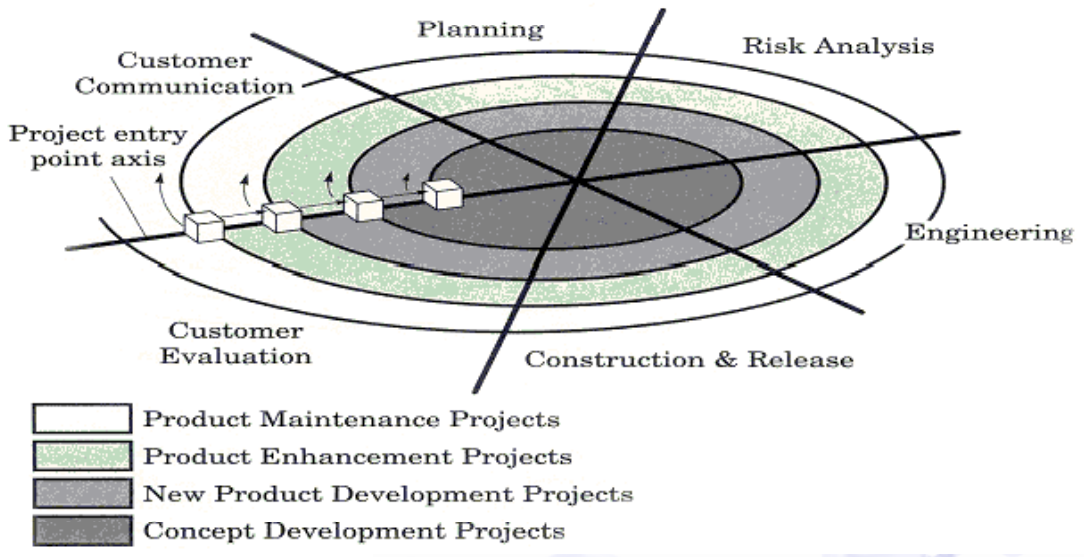
مدل حلزونی(Spiral Model)

- نسخه اولیه از محصول در این مدل نسخه ساده ای می باشد که در تکرارهای بعدی کامل می گردد. در این مدل فعالیتها به شش دسته تقسیم می گرددکه هر کدام از آنها را با نام نواحی کاری(Work Area) می شناسند :

  - تعامل با مشتری و تعریف نیازمندیها : تعیین خواسته ها از جانب مشتری
  - برنامه ریزی (Planning) : تعیین اهداف ، آلترناتیوها و محدودیتها تعیین منابع و ایجاد زمانبندی
  - آنالیز ریسک : تحلیل آلترناتیوها ، شناسائی ریسکها و راهکارهای مقابله با آنها
  - مهندسی (Engineering) : توسعه محصول سطح بعدی
  - ساخت و ارائه : ساخت ،آزمایش و انتقال (تحویل مستندات ، آموزش و ...)
  - ارزیابی مشتری (Customer Evaluation) : ارزیابی نتایج مهندسی

مدل حلزونی(ادامه)

- در تمامی مراحل فوق فعالیتهای چتری نیز به موازات اجرا می گردند.



مدل حلزونی برنده برنده(Win-Win)

- در بخش تعامل با مشتری نیازمندیها از سوی مشتری می بایست مشخص شوند .
- جهت این موضوع لازم است مشتری به یک موازنه (trade off) بین نیازمندیهای خود و تیم توسعه برسد.
- به عبارت دیگر موازنه ای بین عملکرد ، قابلیتهای سیستم و کارائی از طرفی و هزینه و زمان از سوی دیگر برقرار نماید.
- در این شرایط تلاش می گردد اکثر نیازمندیهای مشتری در مقابل زمان و قیمت مناسب جهت تیم توسعه دهنده نرم افزار فراهم گردد(برد-برد).

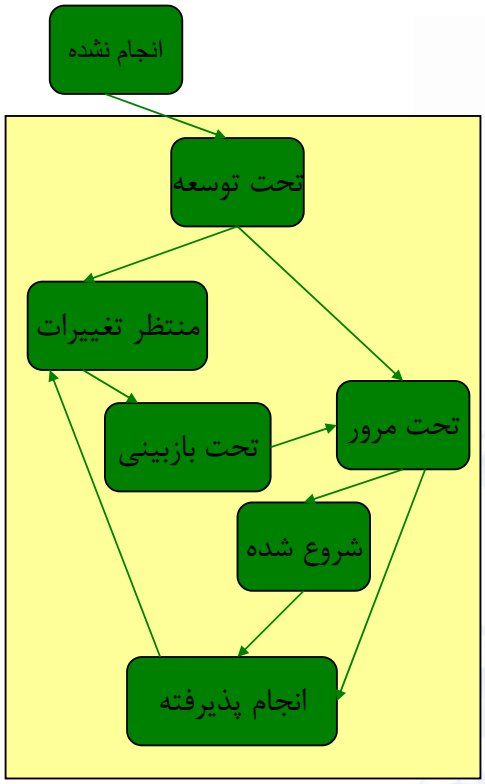
مدل حلزونی برنده برنده(Win-Win)

- در مدل مذکور در بخش تعامل با مشتری و تعیین نیازمندیها قسمتهای زیر جایگزین می گردد:
  - شناسائی ذی نفع ها (Stakeholders)
  - مذاکره جهت حصول به توافق (در راستای قاعده برد – برد)

مدل توسعه همزمان (Concurrent Development)

- در این مدل هر فعالیت دارای چندین حالت می باشد که با تعریف مجموعه ای از رخدادها شاهد گذار از حالتی به حالت دیگر خواهیم بود. به بیان دیگر تمامی فعالیتها بصورت همزمان وجود دارند لیکن در حالتهای متفاوتی قرار می گیرند.
- بعنوان مثال در یک مدل فعالیتهای نمونه سازی، تعریف نیازمندیها و طراحی را داشته باشیم که می خواهیم به صورت همروندآنها را به پیش ببریم و هر یک در طول زمان در وضعیتها و حالات متفاوتی قرار می گیرند.

مدل توسعه همزمان (ادامه)



- به عنوان مثال بعد از اتمام فعالیت تعامل با کاربر این فعالیت در حالت «منتظر تغییرات» رفت ، فعالیت تحلیل از حالت «انجام نشده» به حالت « تحت توسعه» وارد می گردد. چنانچه کاربر احتیاج به انجام تغییرات در نیازمندیهای خود داشته باشد فعالیت تحلیل به حالت «منتظر تغییرات» می رود.
- در این مدل مجموعه ای از وقایع (event) ها تعریف می شوند که برای هر فعالیت انتقال از حالتی به حالت دیگر را رقم می زنند.

مدل توسعه مبتنی بر مولفه (Component Base Development)

- این مدل بر اساس الگوی شیء گرائی استوار است.
- الگوی شیء گرائی بر روی مفهومی به نام کلاس که تلفیقی از ساختمان داده و الگوریتم است تاکید می نماید.
- مدل مذکور بصورت تکاملی بوده و اغلب ویژگیهای مدل حلزونی را شامل می گردد . لیکن در این مدل ساخت سیستم بر مبنای مولفه ها (بعنوان مثال کلاس) شکل می گیرد .
- این مدل قابلیت استفاده مجدد از نرم افزار را افزایش می دهد.

مدل توسعه مبتنی بر مولفه (ادامه)

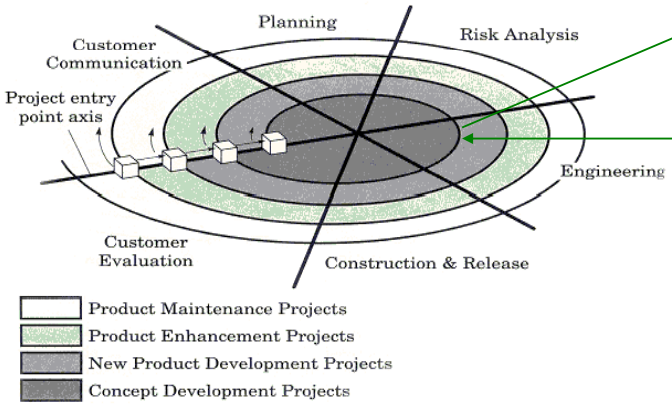
انجمن QSM associates در گزارشی ضمن تاکید بر قابلیت استفاده مجدد از نرم افزار برخی از مزایای آن را به شرح زیر بیان می نماید:

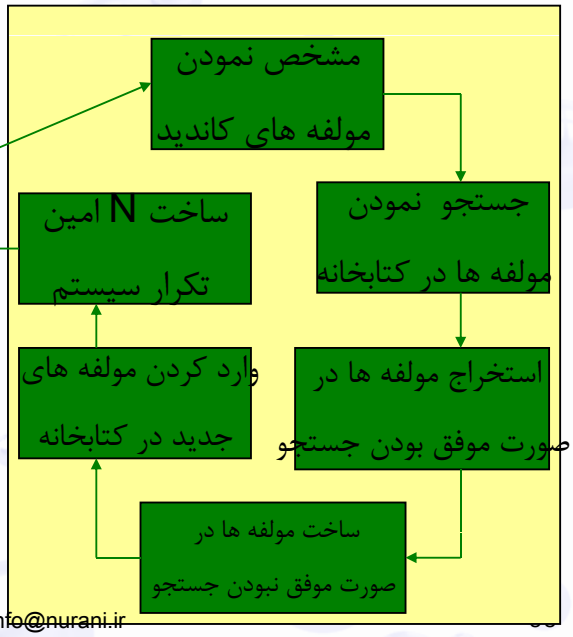
مونتاژ مولفه ها :

- باعث کاهش ۷۰ درصدی زمان توسعه سیتم می گردد .
- موجب کاهش ۴۸ درصدی هزینه های تولید سیتم می گردد.
- سبب افزایش ضریب بهره وری می گردد.

مدل توسعه مبتنی بر مولفه (ادامه)

- نمودار تکامل این مدل نیز دقیقا شبیه مدل حلزونی می باشد، اما در مرحله مهندسی اعمال زیر نیز انجام می شود.





مدل روشهای رسمی یا صوری (Formal Method)

- این مدل شامل مجموعه ای از فعالیتها می باشد که نرم افزار را بصورت ریاضی و رسمی بیان می نماید. به دلیل اینکه در روشهای قراردادی از تئوری مجموعه ها ، نشانه گذاری منطقی و نماد گذاری ریاضی استفاده می شود ، در نتیجه این روش بطور ذاتی ابهام کمتری نسبت به روشهای غیر قراردادی دارد.

- کاربرد: در ساخت سیستمهای امن و حساس مثل کنترل هواپیما و تجهیزات پزشکی

- معایب :
  - زمان و هزینه بالا
  - آموزش گسترده جهت مهندسین نرم افزار
  - دشواری بکارگیری مدلها در تعامل با مشتریانی که فاقد دید فنی می باشند.

تکنیکهای نسل چهارم

- محور اصلی این روش استفاده از ابزارهای مهندسی نرم افزار است. مهندس نرم افزار یک مشخصه از نرم افزار را در سطح بالا تعریف نموده (معمولا با استفاده از نمادها و مدلها) سپس این ابزار کد لازم را تولید می نمایند.

تکنیکهای نسل چهارم

- در محیط های توسعه نرم افزار کنونی که از 4GT حمایت می کنند ابزارهای زیر وجود دارد:
  - زبانهای غیر رویه ای برای پرس و جو از پایگاه داده
  - تولید گزارش
  - دستکاری داده ها
  - قابلیتهای گرافیکی سطح بالا