



دانشگاه فنی و حرفه ای
آموزشکده فنی و حرفه ای امام
خمینی (ره) قاین



دانشگاه فنی و حرفه‌ای
آموزشکده فنی و حرفه‌ای امام خمینی (ره) قاین

آموزش پایگاه داده‌ها

جلسه پنجم: فصل دوم فرایند (قسمت دوم)

مدرس:

فریبا رحیمی

آموزشکده فنی و حرفه‌ای امام خمینی

(ره) قاین



مدل‌های فرآیند تجویزی (سنتی)

prescriptive process models

این مدل در ابتدا برای نظم بخشیدن به آشوب موجود در توسعه ی نرم افزار پیشنهاد شد.

این دو مورد در جلسه ی
قبل گفته شد

The Waterfall Model

۱- مدل آبشاری

Incremental Process Models

۲- مدل‌های فرآیند افزایشی

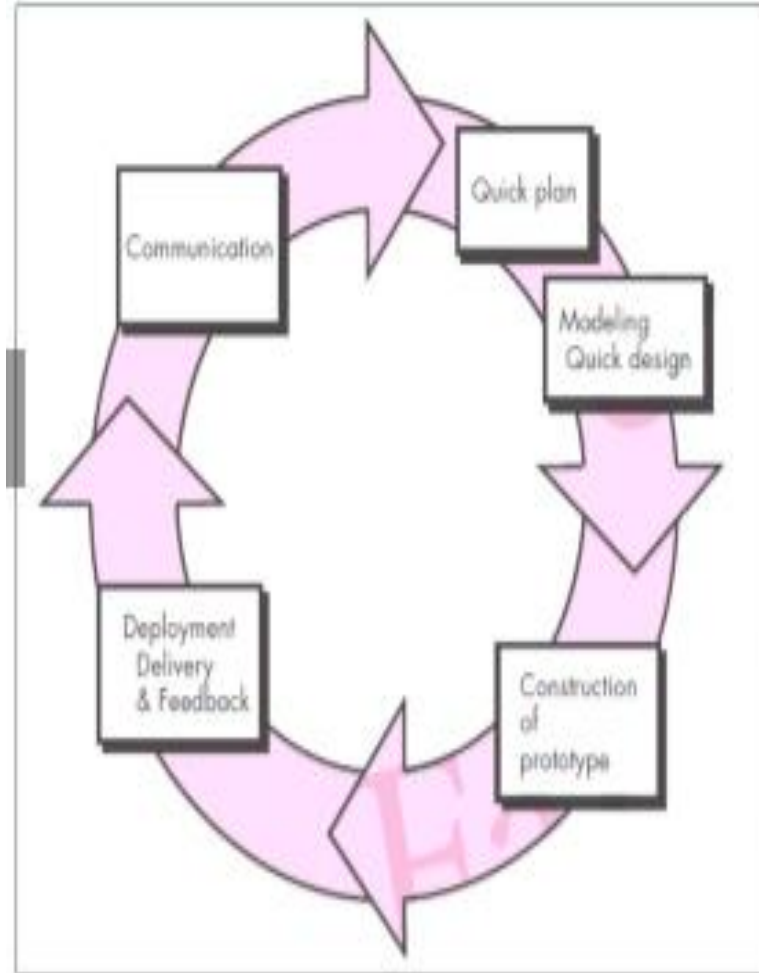
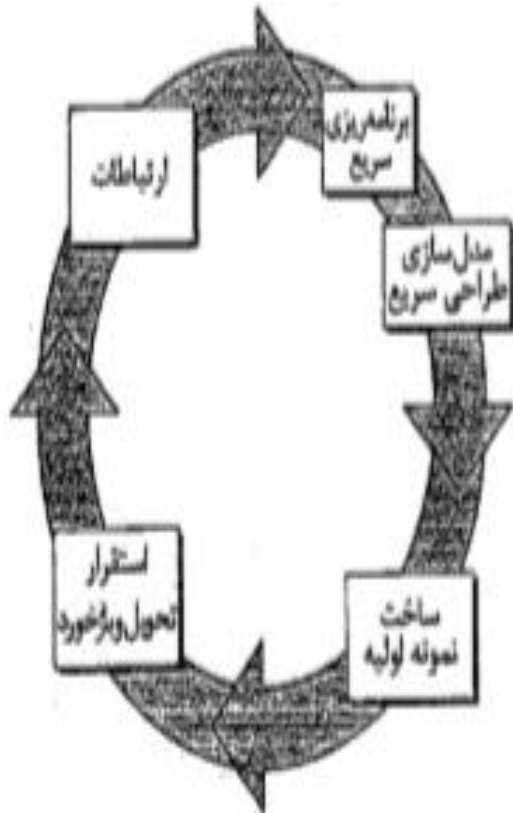
Evolutionary Process Models

۳- مدل‌های فرآیند تکاملی

Concurrent Models

۴- مدل توسعه همروند

مدل های فرآیند تکاملی



الگوی ساخت نمونه اولیه با جمع آوری خواسته ها آغاز می شود.

مشتری و سازنده با هم ملاقات می کنند و اهداف کلی نرم افزار را تعیین می کنند. همه ی خواسته های معلوم را شناسایی می کند و زمینه هایی را مطرح می کند که تعریف بیشتر در آنها ضروری است. سپس یک **طراحی سریع** صورت می گیرد.

در طراحی سریع هدف اصلی ارائه آن دسته از ویژگی های نرم افزار است که مورد توجه کاربر می باشد مثل روش های وارد کردن اطلاعات و فرمت های خارجی، طراحی سریع منجر به ساخت یک نمونه اولیه می شود.

نمونه اولیه مورد ارزیابی مشتری قرار گرفته و از آن برای پالایش نرم افزار مورد نظر استفاده می شود.

با تنظیم نمونه اولیه برای برآوردن نیازهای مشتری تکرار رخ می دهد و در عین حال سازنده بهتر می فهمد که چه نیازهایی باید برآورده شود.



در اکثر پروژه‌ها نخستین سیستمی که ساخته می‌شود چندان قابل استفاده نیست زیرا ممکن است بیش از حد آهسته باشد یا بیش از حد بزرگ باشد و یا استفاده از آن دشوار باشد و یا اینکه هر ۳ ایراد را با هم داشته باشد و چاره‌ای جز شروع دوباره وجود ندارد و باید نسخه دیگری ساخت که این مشکلات در آن حل شده باشد.

نمونه اولیه می‌تواند بعنوان **نخستین سیستم** عمل کند، یعنی دور انداخته شود.

بعضی نمونه‌های اولیه طبیعتی تکاملی دارند و به تدریج به سیستم واقعی تکامل می‌یابد.



دلایل مشکل ساز بودن ساخت نمونه اولیه

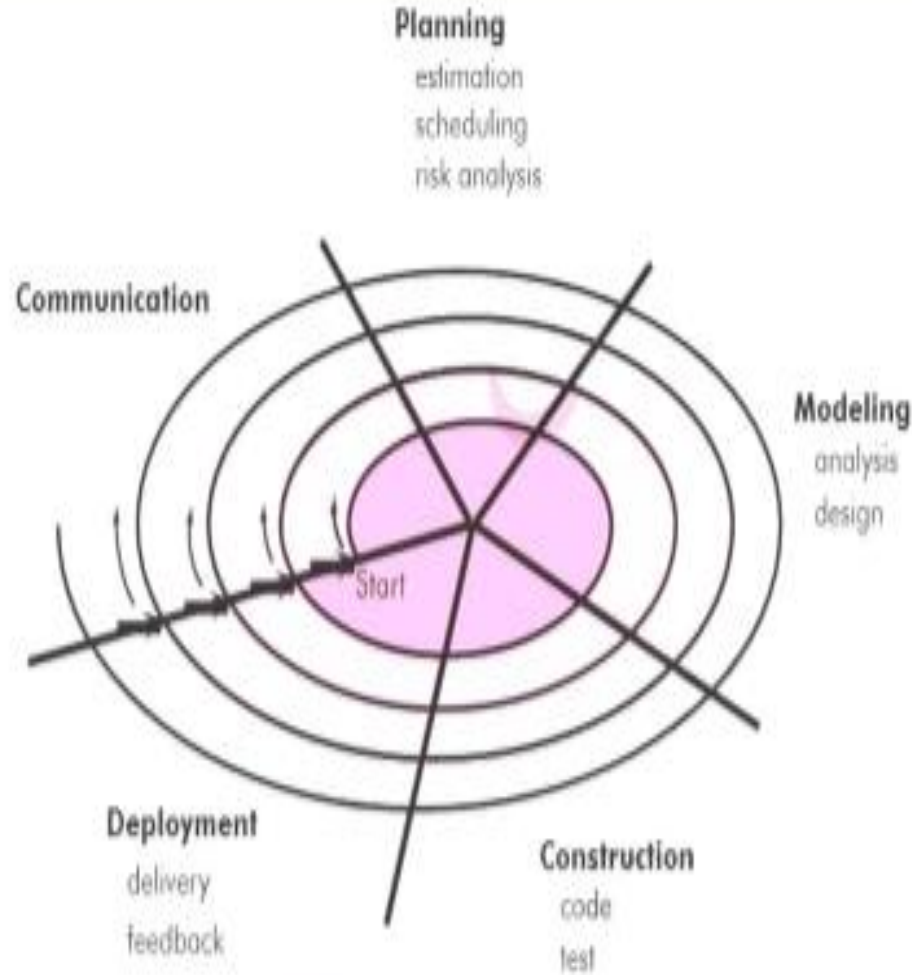
۱- افراد ذی نفع چیزی را می بینند که ظاهراً یک نسخه کاری از نرم افزار است ولی نمی دانند که این نسخه به درستی کار نمی کند و کیفیت کلی نرم افزار و قابلیت نگهداری درازمدت را ندارد.

۲- مهندس نرم افزار غالباً برای بکارگیری هرچه سریعتر نمونه اولیه در پیاده سازی آن کوتاهی می کند.

ممکن است از یک سیستم عامل یا زبان برنامه نویسی نامناسب استفاده شود

و یا حتی ممکن است یک الگوریتم ناکارآمد پیاده سازی شود .

مدل مارپیچی (حلزونی)



یک مدل فرآیند نرم افزاری تکاملی است که ماهیت تکراری مدل ساخت نمونه اولیه را با جنبه های کنترلی و سیستماتیک مدل ترتیبی خطی (آبشاری) تلفیق می کند.

این مدل پتانسیل لازم برای بسط سریع نسخه های تکاملی نرم افزار را داراست.

با استفاده از مدل مارپیچی، نرم افزار به صورت یک سری نگارش های تکاملی توسعه می یابد. طی نخستین دوره های تکرار، نگارش تکاملی، ممکن است یک مدل کاغذی یا یک نمونه اولیه باشد. طی تکرارهای بعدی هر بار نسخه کاملتری از سیستم مهندسی شده تولید می شود.



مدل مارپیچی (حلزونی)...

برخلاف سایر مدل‌های فرآیند کلاسیک که با تحویل نرم افزار پایان می‌یابند، مدل مارپیچی را می‌توان طوری تطبیق داد که در سرتاسر عمر نرم افزار کامپیوتری قابل به کارگیری باشد.

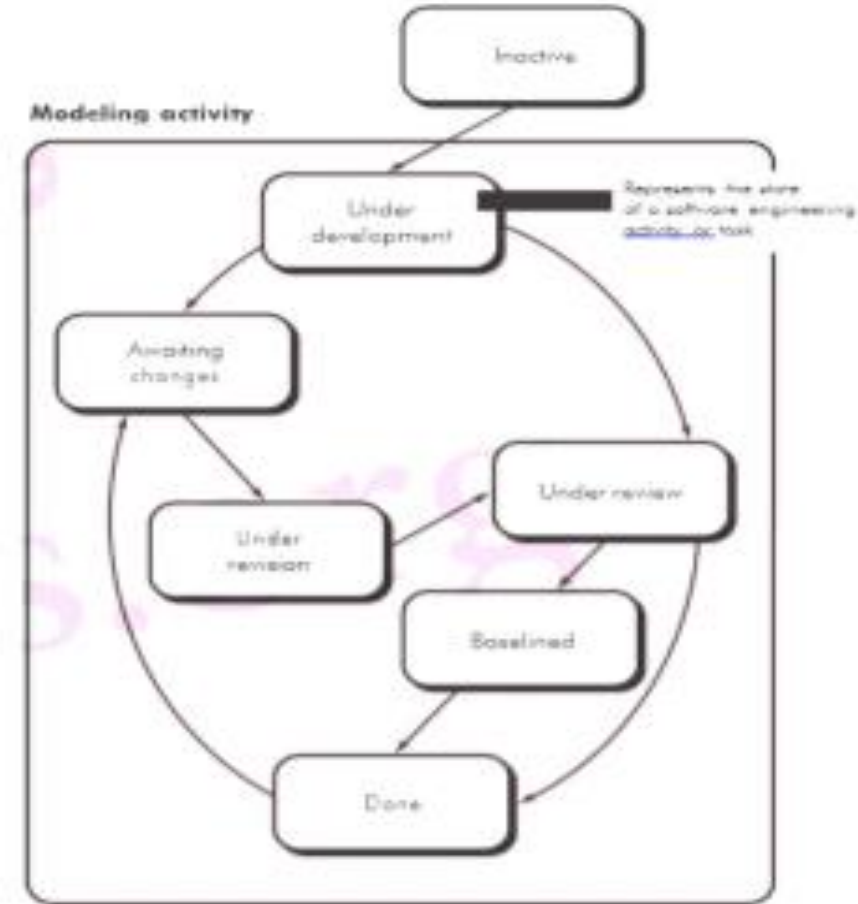
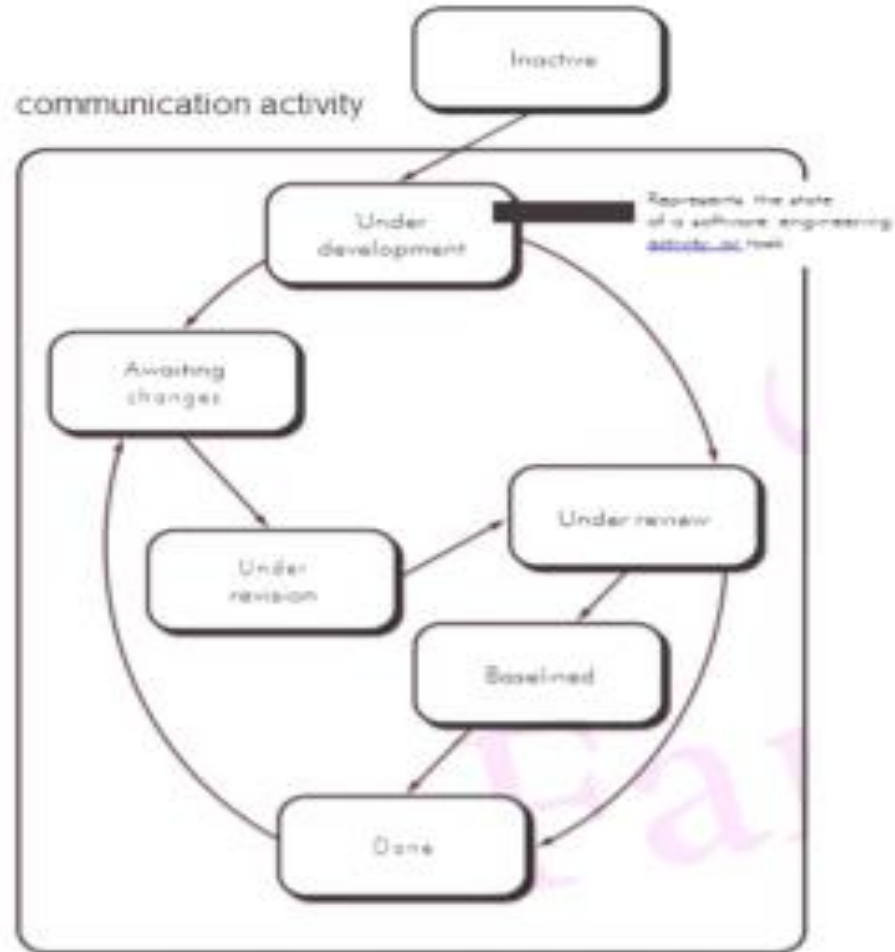
مدل مارپیچی یک روش واقع‌گرا برای توسعه نرم افزارها و سیستم‌هایی در مقیاس انبوه است.

اگر یک خطر عمده کشف و اداره نشود بدون شک مشکلاتی به بار خواهد آمد.



مدل توسعه‌ی همروند

Concurrent Models



مدل توسعه همروند (مهندسی همروند) به تیم نرم افزار این امکان را می دهد عناصر تکراری و همروند هرکدام از مدل های فرآیند را ارائه نماید.



مدل توسعه‌ی همروند...





مدل توسعه‌ی همروند...

شکل طراحی از یک فعالیت یا مدل توسعه همروند ارائه می‌دهد. این فعالیت مدل سازی ممکن است در هر زمان و در حالت های مختلف ایجاد شود برای مثال در ابتدای یک پروژه فعالیت برقراری ارتباط نخستین تکرار خود را به پایان رسانده و در حالت انتظار تغییرات قرار دارد در واقع فعالیت مدل سازی که هنگام کامل شدن ارتباط اولیه با مشتری در حالت غیرفعال قرارداشت اکنون دستخوش گذار به حالت تحت توسعه می‌شود ولی اگر مشتری متذکر شود که تغییراتی در خواسته ها باید صورت گیرد فعالیت تحلیل از حالت تحت توسعه به حالت انتظار تغییرات می‌رود.

درواقع مدل توسعه همروند یک سری رویداد تعریف می‌کند که باعث گذار از حالتی به حالت دیگر برای هر یک از فعالیت های مهندسی نرم افزار می‌شود.

برای مثال طی اولین مراحل طراحی یکی از کنش های اصلی در مهندسی نرم افزار که طی فعالیت مدل سازی انجام می‌شود یک ناسازگاری در مدل تحلیل کشف می‌شود این باعث تولید رویداد تصحیح مدل تحلیل می‌شود که گذار از کنش تحلیل خواسته ها را از حالت انجام شده به حالت انتظار تغییرات سبب می‌شود.



نقاط ضعف فرایندهای تکاملی

- ۱- تهیه نمونه اولیه به دلیل قطعی نبودن تعداد چرخه‌های لازم برای ساخته شدن محصول، برای برنامه ریزی پروژه ایجاد مشکل می‌کند.
- ۲- حداکثر سرعت تکامل را تعیین نمی‌کنند.
- ۳- در فرایندهای نرم افزاری، **انعطاف پذیری** و **بسط پذیری** باید بیش از **کیفیت بالا** مورد توجه قرار گیرد.



مدل های فرایند تخصصی

این مدل ها شامل بسیاری از ویژگی های یک یا چند مدل سنتی ارائه شده در بخش های قبلی می شوند. ولی، این مدل ها را معمولا هنگامی به کار می برند که یک روش مهندسی تخصصی یا روشی با مشخصات دقیق انتخاب می شود.

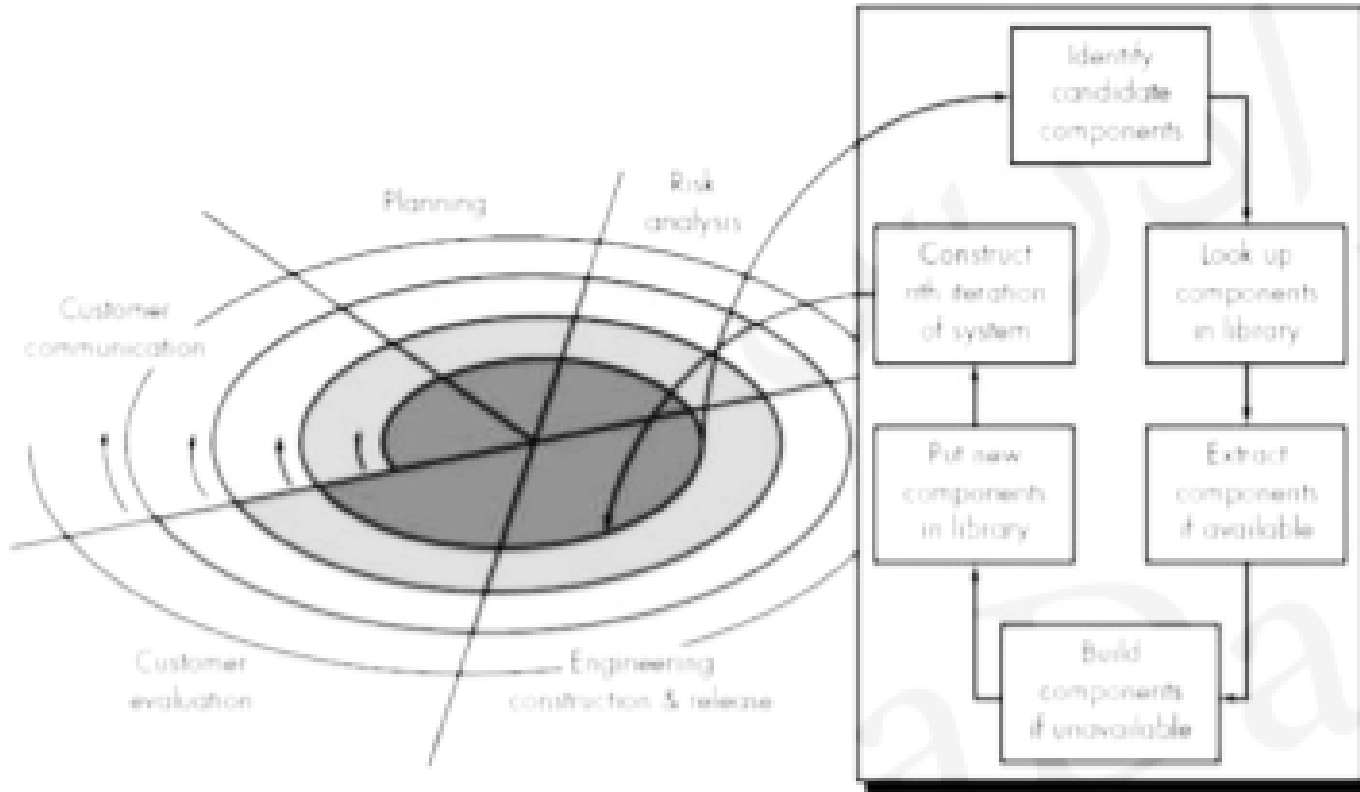
۱- توسعه مبتنی بر مولفه ها

۲- مدل روش های رسمی

۳- توسعه نرم افزار به روش جنبه گرا



توسعه مبتنی بر مؤلفه



مؤلفه های نرم افزاری آماده، توسط عده ای از فروشندگان این مؤلفه ها توسعه داده می شوند، عملکرد مورد نظر را با واسطه هایی مناسب فراهم می آورند، به طوری که مؤلفه را می توان به خوبی در سیستم در حال ساخت الحاق کرد.

توسعه مبتنی بر مؤلفه ها بسیاری از خصوصیات مدل مارپیچی را در برمی گیرد.



توسعه‌ی مبتنی بر مؤلفه شامل موارد زیر است:

- ۱- محصولات مبتنی بر مؤلفه‌های موجود از نظر دامنه‌ی کاربرد مورد نظر بررسی و ارزیابی می‌شوند.
- ۲- مسائل مربوط به انسجام مؤلفه‌ها در نظر گرفته می‌شود.
- ۳- برای چیدمان مؤلفه‌ها یک معماری نرم افزار طراحی می‌شود.
- ۴- مؤلفه‌ها در این معماری قرار داده می‌شوند.
- ۵- آزمونی جامع برای حصول اطمینان از عملکرد درست بدست می‌آید.



مدل روشهای رسمی

شامل مجموعه ای از فعالیت ها می شود که به مشخص کردن ریاضی و رسمی نرم افزار کامپیوتری منجر می شود.

روش های رسمی، مهندسی نرم افزار را قادر می سازد که با اعمال یک نظم **ریاضی** شدید، سیستم کامپیوتری را مشخص کند، بست دهد و واریسی کند.

شکل دیگر از این روش که **مهندسی نرم افزار اتاق تمیز** نامیده می شود در حال حاضر توسط برخی از سازمان های نرم افزاری به کار می رود. **cleanroom software engineering**

مدل روش های رسمی گرچه چندان عمومیت نخواهد یافت اما نوید بخش نرم افزاری عاری از نقص است.

با این حال، ملاحظات مربوط به قابلیت اجرای آن در محیط های تجارتي چنین اعلام شده است:

۱- توسعه این مدل ها، بسیار وقت گیر و **پرهزینه** است.

۲- از آنجا که تعداد محدودی از نرم افزارسازان دارای زمینه ی لازم برای اجرای روش های رسمی هستند **آموزش** موردنیاز است.

۳- استفاده از مدل ها به عنوان راهکار ارتباطی با مشتریانی که دید فنی ندارند **دشوار** است.



دانشگاه فنی و حرفه‌ای
آموزشگاه فنی و حرفه‌ای امام خمینی (ره) قاین

توسعه‌ی نرم افزار به روش جنبه گرا

Aspect-Oriented Software Development(AOSD)

AOSD یک الگوی مهندسی نسبتاً جدید است که رویکردی فرآیندی و روش شناختی برای تعریف، مشخص سازی، طراحی و ساخت جنبه ها ارائه می دهد.

هر فرآیند نرم افزار که انتخاب شود، سازندگان نرم افزارهای پیچیده، مجموعه ای از ویژگی ها، عملکردها و محتویات اطلاعاتی متمرکز را پیاده سازی می کنند.

این خصوصیات نرم افزاری متمرکز به صورت مؤلفه هایی مثل کلاس های شی گرا، مدل سازی و سپس در حیطه یک معماری سیستم بنا می شوند.

با پیچیده تر شدن سیستم های کامپیوتری مدرن، دغدغه های خاصی کل معماری را دربر می گیرد. هنگامی که این دغدغه ها در وظایف، ویژگی و اطلاعات سیستمی با یکدیگر تلاقی می کند غالباً از آنها به عنوان دغدغه های متلاقی یاد می شود. خواسته های جنبه گرا، آن دسته از دغدغه های متلاقی را تعریف می کنند که بر معماری نرم افزار تأثیر می گذارند.



فرآیند یکپارچه (the unified process)

فرآیند یکپارچه (UP) از جهاتی تلاشی برای گرد هم آوردن بهترین ویژگی‌ها و خصوصیات مدل‌های فرآیند سنتی است، ولی آنها را به شیوه‌ای مشخص می‌کند که بسیاری از بهترین اصول توسعه نرم‌افزار چابک را پیاده‌سازی می‌کند.

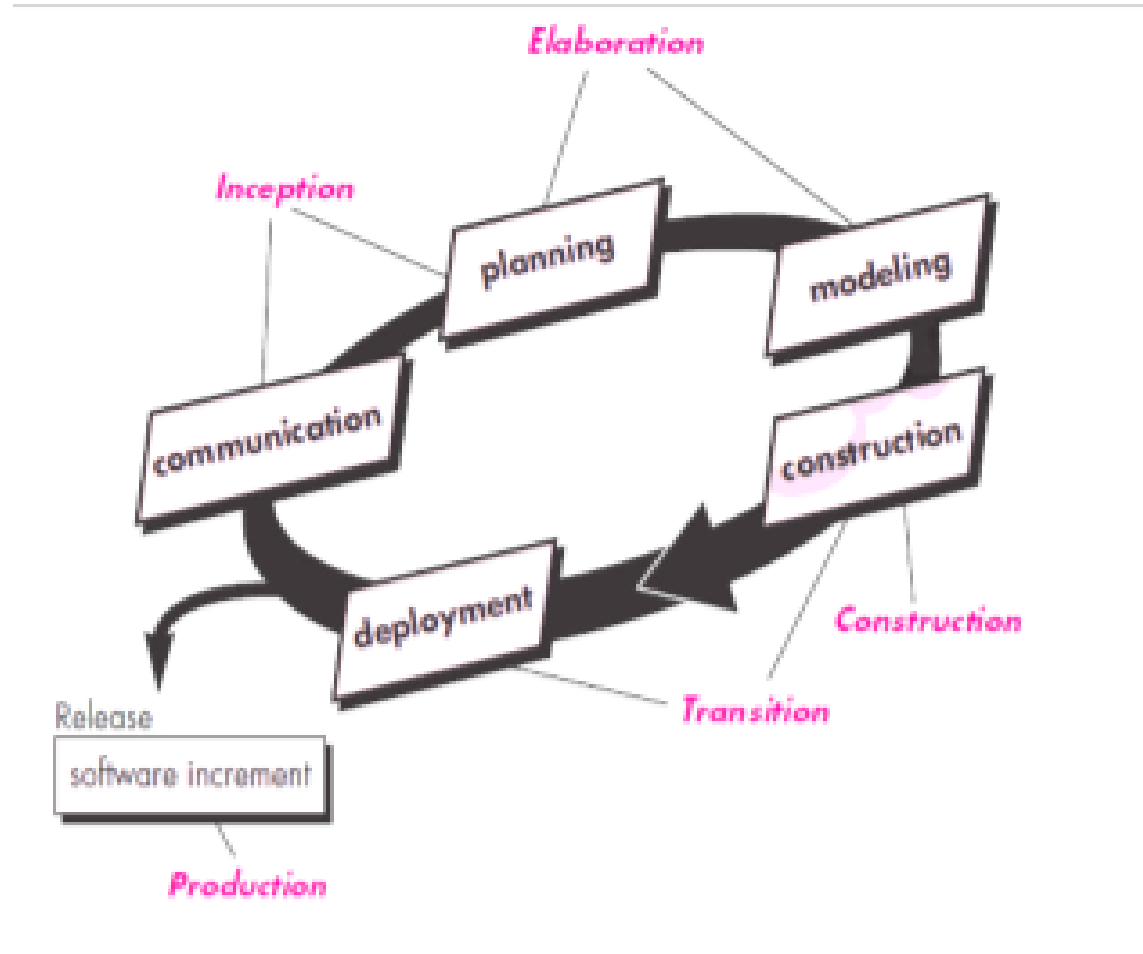
در فرآیند یکپارچه اهمیت برقراری ارتباط با مشتریان و روش‌های ساده برای توصیف دیدگاه مشتریان (use case) یک سیستم به خوبی درک می‌شود.

در این فرآیند بر اهمیت نقش معماری نرم‌افزار تاکید می‌شود.

در این فرآیند، یک جریان فرآیند مبتنی بر تکرار و افزایشی پیشنهاد می‌شود.

در اوایل دهه ۱۹۹۰، کار روی یک روش یکپارچه آغاز شد که بهترین ویژگی‌های هر کدام از روش‌های طراحی و تحلیل شی‌گرا را تلفیق می‌کرد و ویژگی‌هایی از سایر کارشناسان در مدل‌سازی شی‌گرا بر آن منطبق می‌ساخت. نتیجه، زبان مدل‌سازی یکپارچه (UML) است که حاوی یک نمادگذاری قدرتمند برای مدل‌سازی و توسعه سیستم‌های شی‌گرا است.

مراحل فرایند یکپارچه



مرحله اول: شامل هر ۲ فعالیت برقراری ارتباط با مشتریان و برنامه ریزی می شود.

مرحله شناخت: شامل فعالیت های برقراری ارتباط و مدل سازی در مدل فرآیند کلی می شود.

در این مرحله یوزگیس های مقدماتی که به عنوان بخشی از مرحله آغازین ایجاد شوند، پالایش یافته و بسط داده می شوند.

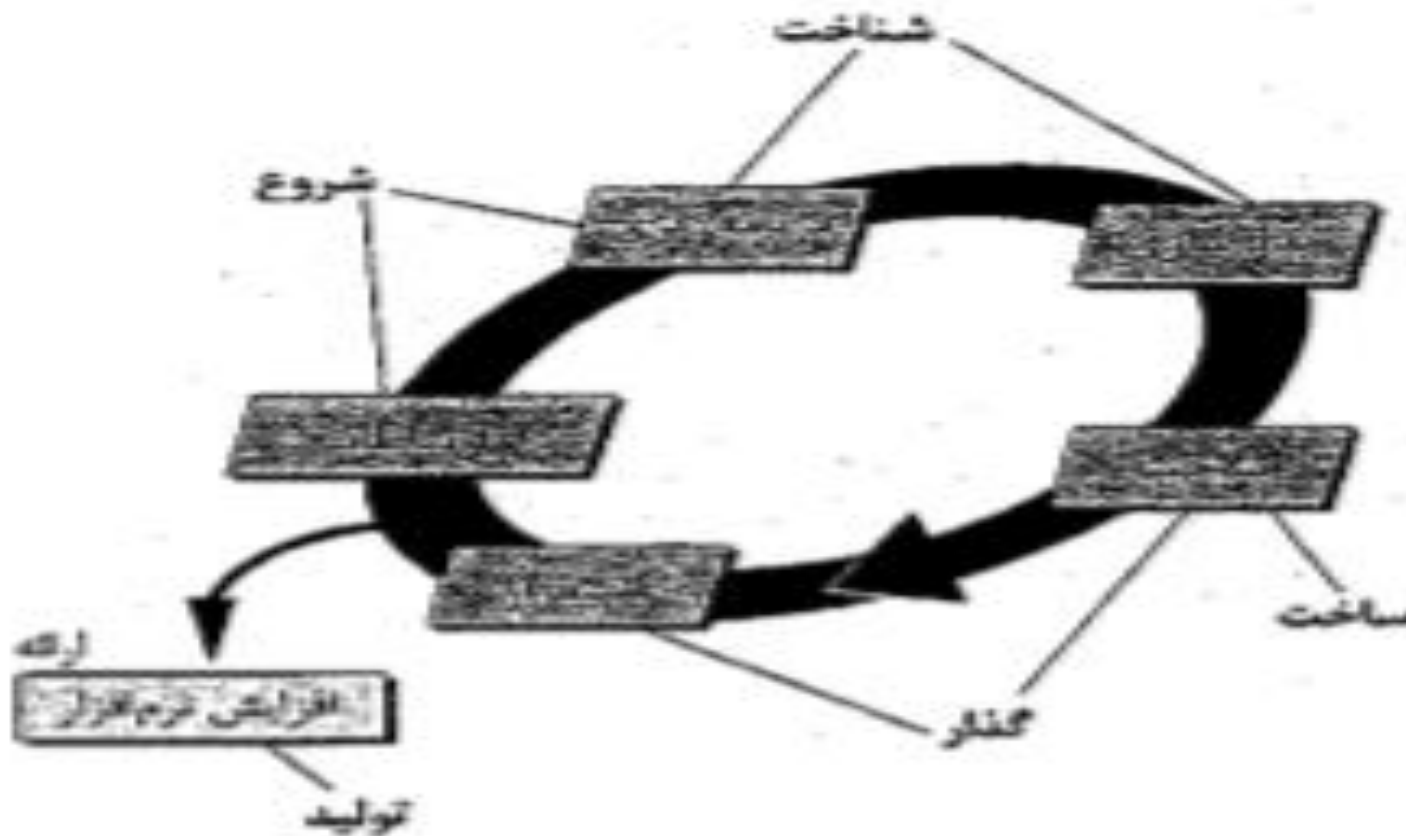
مرحله ساخت: هم ارز فعالیت ساخت است که برای فرایند نرم افزار کلی تعریف شد.

مرحله گذار: شامل آخرین مراحل در فعالیت ساخت در مدار کلی و اولین بخش از استقرار در مدل کلی می شود.

مرحله تولید: منطبق بر فعالیت استقرار در فرآیند کلی است.



مراحل فرایند یکپارچه ...





دانشگاه فنی و حرفه‌ای
آموزشگاه فنی و حرفه‌ای امام خمینی (ره) قاین

مدل های فرایند تیمی و شخصی

PERSONAL AND TEAM PROCESS MODELS

بهترین فرآیند نرم افزار ، فرآیندی است که به کسانی که کار می کنند نزدیک باشد.

فرآیند نرم افزار شخصی (Personal Software Process (PSP)

فرایند نرم افزار تیمی (Team Software Process (TSP)



فرآیند نرم افزار شخصی (PSP) (Personal software process)

هر سازنده ای برای ساختن نرم افزار کامپیوتری از یک فرآیند استفاده می کند.
این فرآیند ممکن است برحسب اتفاق شکل گرفته باشد یا با هدفی خاص ممکن است روزانه تغییر کند.
ممکن است اثربخش نباشد مؤثر نباشد یا حتی موفقیت آمیز هم نباشد ولی فرآیندی وجود دارد.
PSP بر اندازه گیری شخصی محصول کاری تولید شده و کیفیت حاصل از محصول کاری تأکید دارد.

فعالیت های چارچوبی در مدل PSP

(۲) طراحی سطح بالا High-level design

(۱) برنامه ریزی Planning

(۴) توسعه Development

(۳) مرور طراحی سطح بالا High-level design review

(۵) پایان کار Postmortem



فرآیند نرم افزار تیمی (TSP) (team software process)

هدف **TSP** تشکیل یک تیم پروژه خود هدایت گراست که سازماندهی برای تولید نرم افزارهای با کیفیت را خود عهده دار می شود.

فعالیت های زیر برای TSP تعریف می شود:

- ۱- تشکیل تیم های خود هدایت گری که کار خود را برنامه ریزی و پیگیری می کنند، اهداف را تعیین می کنند و خود به تعیین فرآیندها و طرح ها اقدام می نمایند. این تیم ها می توانند تیم های نرم افزاری محض یا تیم های محصولات انسجام یافته شامل ۳ تا حدود ۲۰ مهندس باشد.
- ۲- نشان دادن شیوه ی راهبری و ایجاد انگیزه در تیم ها به مدیران و چگونگی کمک به آنها در حفظ حداکثر کارایی
- ۳- شتاب بخشیدن به بهبود فرآیند نرم افزار با نهادینه ساختن **CMM** سطح ۵.
- ۴- فراهم ساختن دستور العمل بهسازی برای سازمان های بالغ
- ۵- تسهیل آموزش دانشگاهی مهارت های تیمی در سطح صنعتی.



دانشگاه فنی و حرفه‌ای
آموزشگاه فنی و حرفه‌ای امام خمینی (ره) قاین

فعالیت های چارچوبی تعریف شده در TSP

۱- آغاز پروژه

۲- طراحی سطح بالا

۳- پیاده سازی

۴- انسجام دهی

۵- آزمون

پایان فصل ۲