

دانشگاه فنی و حرفه ای

دانشکده فنی و حرفه ای امام جعفر صادق (ع) آستانه اشرفیه

مهندسی نرم افزار

فصل دوم : مدل های فرایند

فرآیند چیست؟

وقتی کار می کنید تا یک سیستم یا یک محصول بسازید، حتماً باید یک سری مراحل قابل پیش بینی را چک کنید: یک نقشه راه که در ایجاد نتیجه ای با کیفیت بالا و به موقع شما را یاری می کند.

این نقشه که آن را دنبال می کنید **فرآیند نرم افزار** نام دارد.

چه کسی آن را انجام می دهد؟

مهندسان نرم افزار و مدیران آنها، فرآیند را با نیازهای خود مطابقت داده و سپس آن را دنبال می کنند. به علاوه کسانی که نرم افزار را درخواست کرده اند در فرآیند نرم افزار نقش دارند.

چرا فرآیند نرم افزار اهمیت دارد؟

زیرا باعث ثبات، کنترل و سازماندهی فعالیتی می شود که اگر به حال خود گذاشته شود ممکن است باعث آشوب شود.

چه مراحل دارد؟

مراحل فرآیند به نرم افزاری که میخواهید بسازید بستگی دارد .

از دیدگاه مهندس نرم افزار حاصل کار چیست؟

برنامه ها، داده ها و مستنداتی است که به عنوان نتیجه ای از فعالیت های مهندسی نرم افزار مشخص شده توسط فرآیند تولید می شوند.

چطور مطمئن شوم که درست از عهده کار برآمده ام؟

چند راهکار برای ارزیابی فرآیند نرم افزار وجود دارد.

ولی کیفیت، به موقع بودن و کارایی درازمدت محصولی که ساخت اید، بهترین ملاک ها برای بازدهی فرآیند مورد استفاده است.

Process framework

Umbrella activities

framework activity # 1

software engineering action #1.1

Task sets

work tasks
work products
quality assurance points
project milestones

⋮

software engineering action #1.k

Task sets

work tasks
work products
quality assurance points
project milestones

⋮

framework activity # n

software engineering action #n.1

Task sets

work tasks
work products
quality assurance points
project milestones

⋮

software engineering action #n.m

Task sets

work tasks
work products
quality assurance points
project milestones

یک مدل فرایند کلی

work tasks
work products
quality assurance points
project milestones

جریان فرآیند

یک جنبه ی مهم از فرآیند نرم افزار است.

شرح می دهد که فعالیتهای چتری و کنش ها و وظایفی که در داخل هر فعالیت چارچوبی رخ می دهند از نظر ترتیب زمانی چگونه سازماندهی می شود.

انواع جریان فرایند:

- ۱- خطی
- ۲- مبتنی بر تکرار
- ۳- تکاملی
- ۴- موازی

جریان فرآیند خطی

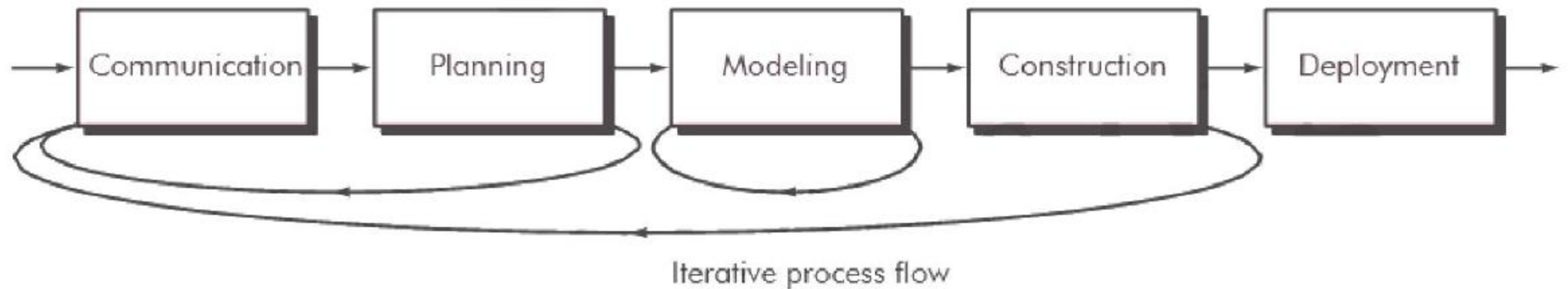
هر کدام از ۵ فعالیت چارچوبی به ترتیب اجرا می شود، به طوری که با ارتباطات آغاز و به استقرار ختم می شود.



Linear process flow

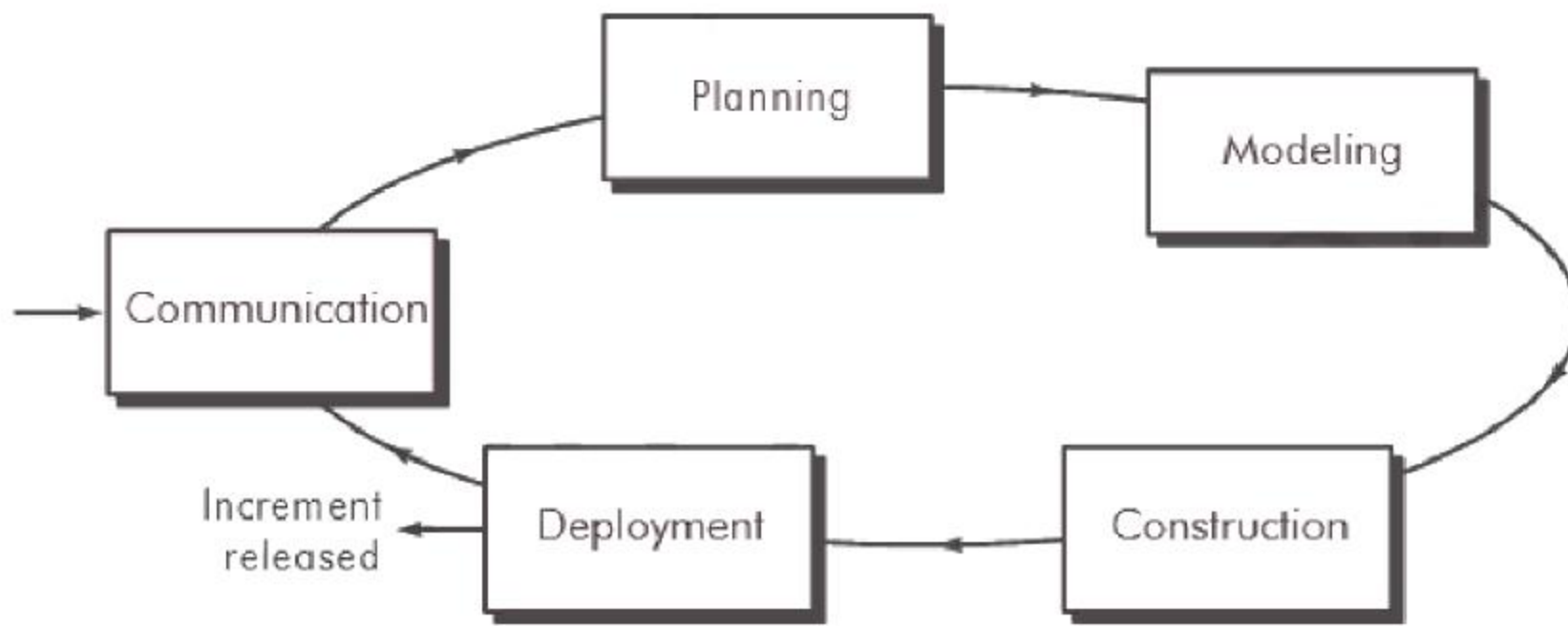
جریان فرآیند مبتنی بر تکرار

در این جریان پیش از رفتن به دور تکرار بعدی یک یا چند فعالیت تکرار میشود.



جریان فرآیند تکاملی

در این جریان فعالیت ها به شیوه ی حلقوی اجرا می شوند .
هر مدار از ۵ فعالیت عبور می کند که به نسخه کاملتری از نرم افزار می انجامد .

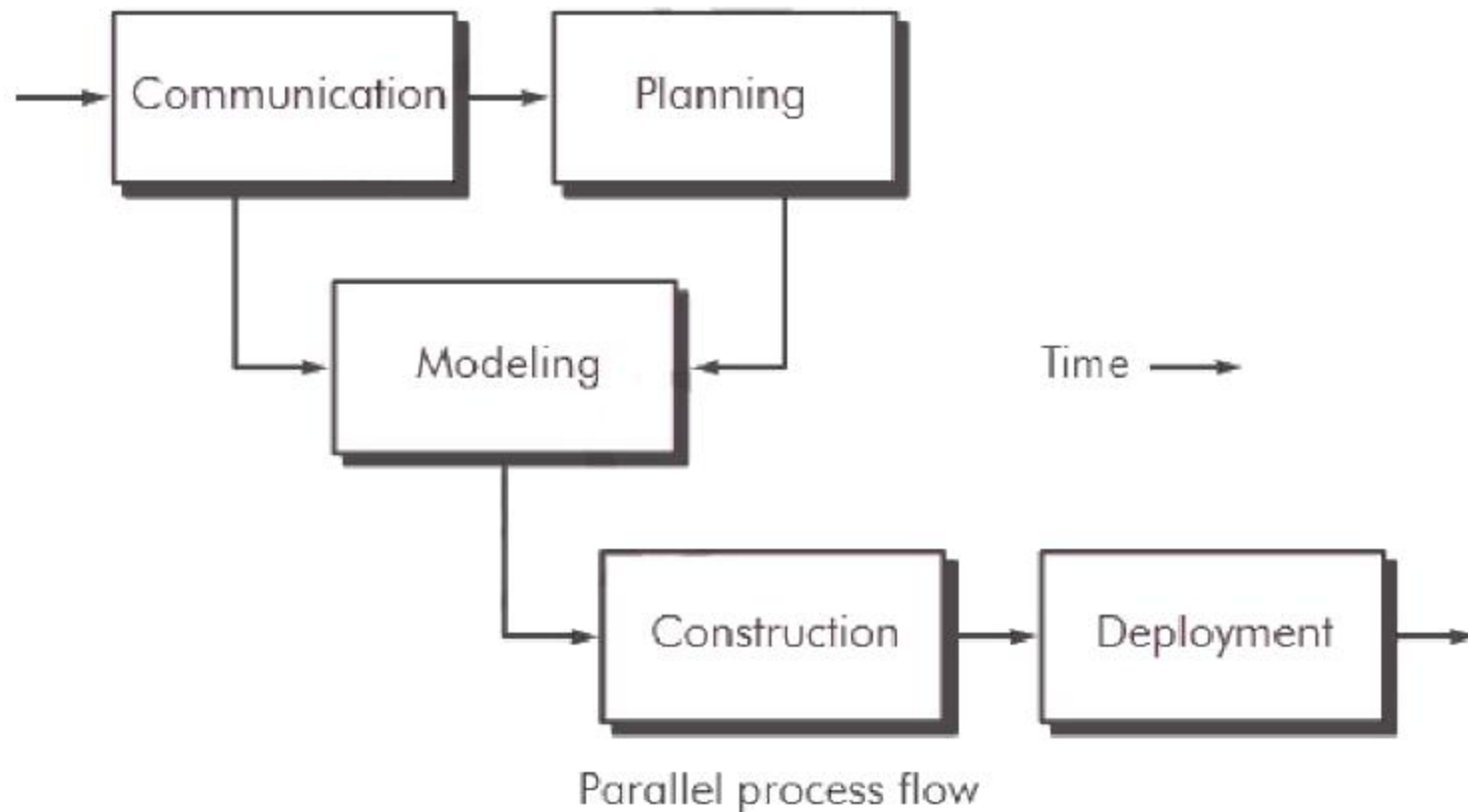


Evolutionary process flow

جریان فرآیند موازی

یک یا چند فعالیت به موازات سایر فعالیت ها انجام می شود .

مثلاً مدل سازی برای یک جنبه از نرم افزار ممکن است به موازات ساختار جنبه ی دیگری از نرم افزار اجرا گردد.



مجموعه وظایف (مثال)

برای یک پروژه نرم افزاری کوچک که یک نفر با خواسته های صریح و ساده در مکانی دوردست درخواست می کند ، فعالیت برقراری **ارتباط** ممکن است شامل چیزی در حد یک تماس با ذینفع مورد نظر باشد.

پس تنها کنش لازم ، مکالمه تلفنی است.

مجموعه وظایف :

- ۱- بحث درباره خواسته ها و یادداشت برداشتن
- ۲- سازمان دهی یادداشت ها در یک بیان مختصر از خواسته ها
- ۳- ارسال ایمیل برای بازبینی و تصویب

الگوی های فرآیند

Process Patterns

الگوی فرآیند، مشکلی مرتبط با آن فرآیند را توصیف می کند که طی کار مهندسی نرم افزار با آن مواجه شده باشند ،

محیطی که در آن مشکل مشاهده شده را مشخص می کند

و

یک یا چند راهکار برای آن پیشنهاد می کند.

قالب برای توصیف الگوی فرآیند

نام الگو: دادن نام با معنی به الگو نامی .

نیروها: محیطی که الگو در آن مشاهده می شود و مواردی که مسئله را پدیدار می کند و ممکن است بر راهکار آن تأثیر بگذارند.

نوع: نوع الگو مشخص می شود. (مرحله ای - وظیفه ای - فازی)

حیطه ی اولیه: توصیف شرایطی که الگو در آن کاربرد دارد.

مسئله: مسئله ی خاصی که قرار است توسط این الگو حل شود.

راهکار: توصیف چگونگی پیاده سازی موفق الگو

حیطه ی حاصل: شرایطی را توصیف می کند که نتیجه ی پیاده سازی موفق الگو هستند.

الگوهای مرتبط: فهرستی از همه ی الگوهای فرآیند تهیه کنید که با این الگو ارتباط مستقیم دارند.

الگوهای پیشنهادی توسط امبلر

۱- **الگوی مرحله ای:** مسئله ای مرتبط با یک مسئله ی چارچوبی را برای یک فرآیند تعریف می کند.

۲- **الگوی وظیفه ای:** مسئله ای مرتبط با یک کنش یا وظیفه ی کاری نرم افزاری را تعریف می کند که کار مهندسی نرم افزار موفق به آن بستگی دارد.

۳- **الگوی فازی:** یک سری فعالیت های چارچوبی را تعریف می کند که درون فرآیند رخ می دهند حتی هنگامی که جریان کلی فعالیت ها ماهیتی تکراری داشته باشند.

روش های ارزیابی فرایند و بهبودی

CMMI برای بهسازی (SCAMPI)

یک مدل ارزیابی فرآیند ۵ مرحله ای فراهم می آورد که شامل فازهای زیر می باشد:

(۱) شروع (۲) عیب یابی (۳) ساخت (۴) عملیات (۵) یادگیری

ارزیابی مبتنی بر CMM برای بهبود بخشیدن به فرایندهای داخلی (CBA IPI):

یک تکنیک عیب یابی برای ارزیابی بلوغ نسبی یک سازمان نرم افزاری فراهم می آورد .

: SP/CE (ISO/IEC15504)

استانداردی که مجموعه ای از خواسته ها را برای ارزیابی فرآیند نرم افزار تعریف می کند .

هدف این استاندارد کمک به سازمان ها در توسعه ی یک ارزیابی عینی از بازدهی هرگونه فرآیند نرم افزار تعریف شده است.

ISO 9001:2000 برای نرم افزار:

یک استاندارد عمومی که برای هر سازمانی که مایل به بهسازی کیفیت کلی محصولات، سیستم ها یا سرویس های ارائه شده اش باشد قابل استفاده است.

مدل های فرآیند تجویزی (سنتی)

PRESCRIPTIVE PROCESS MODELS

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

The Waterfall Model

۱- مدل آبشاری

Incremental Process Models

۲- مدل های فرآیند افزایشی

Evolutionary Process Models

۳- مدل های فرآیند تکاملی

Concurrent Models

۴- مدل توسعه همروند

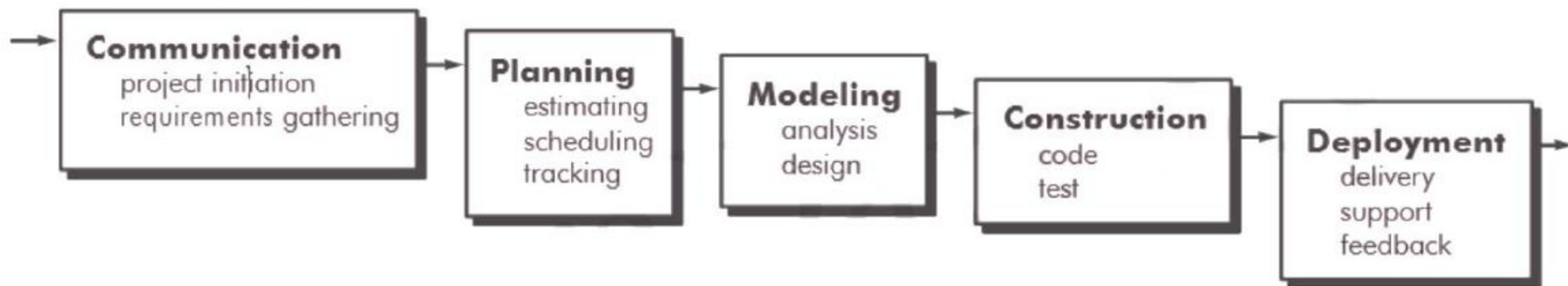
تجویز موارد زیر برای هر پروژه

- مجموعه ای از عناصر فرآیند
- فعالیت های چارچوبی
- عملیات مهندسی نرم افزار
- وظایف
- محصولات کاری
- تضمین کیفیت
- ساز و کارهای کنترل تغییرات

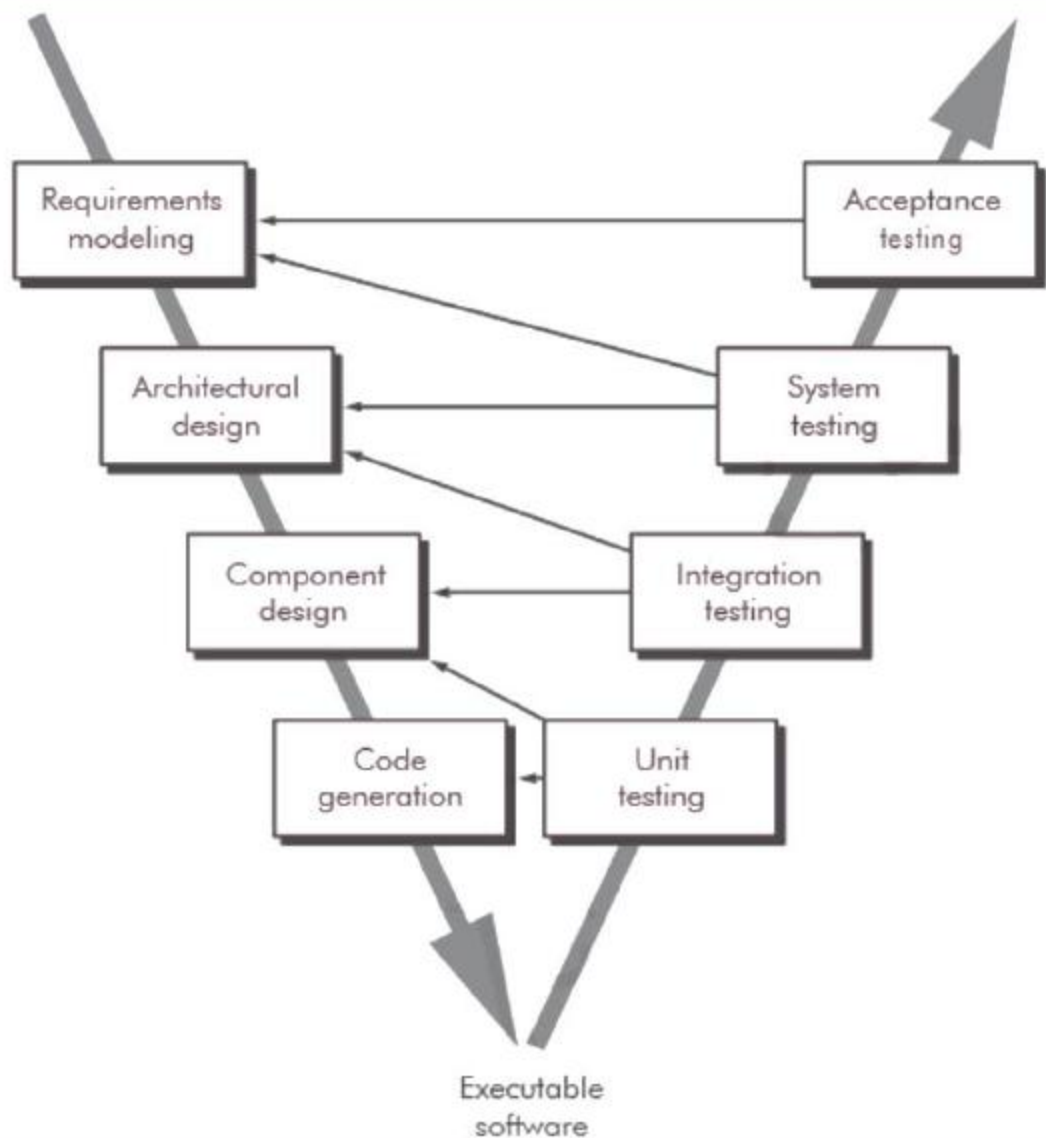
مدل آبشاری

The Waterfall Model

وقتی خواسته های مربوط به یک مسئله به خوبی شناخته شده اند.
هنگامی که کار به طریق خطی از برقراری ارتباط تا استقرار جریان پیدا می کند.



مدل V



شکل دیگر در نمایش مدل آبشاری .

تصویر رابطه تضمین کیفیت با :

- واکنش های مرتبط با ارتباط،
- مدلسازی
- فعالیت های ساخت اولیه.

با حرکت تیم نرم افزاری به طرف پایین و سمت چپ V خواسته های اساسی مسئله رفته رفته پالایش شده و جزئیات بیشتری از آنها تعیین می شود و مسئله و راهکار آن بهتر نشان داده می شود .

هنگامی که کدها نوشته شد، تیم در طرف راست V به طرف بالا حرکت می کند و اساساً یک سری آزمون اجرا می کند تا هرکدام از مدل های ایجادشده در مدت حرکت تیم به طرف پایین را واریسی کند.

مدل V راهی برای تجسم بخشیدن به چگونگی واریسی و اعتبارسنجی در ابتدای کار نرم افزار را فراهم می آورد.

مشکلاتی که به هنگام اجرای مدل ترتیبی خطی پیش می آید:

مدل ترتیبی خطی، قدیمی ترین و پرکاربردترین الگو برای مهندسی نرم افزار است.

مشکلات :

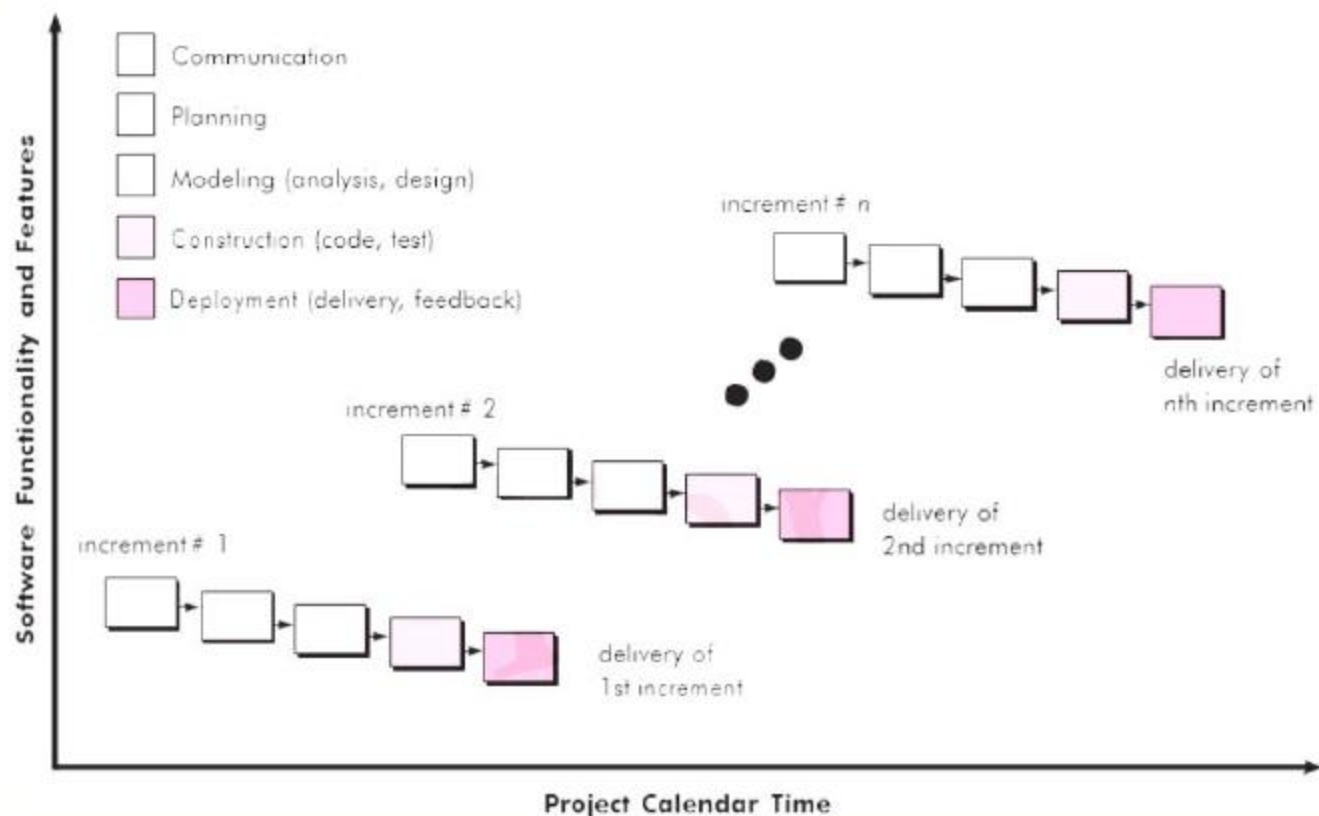
۱- پروژه های واقعی به ندرت جریان ترتیبی پیشنهادشده توسط این مدل را دنبال می کنند.

۲- غالباً برای مشتری دشوار است که همه نیاز خود را به وضوح بیان کند.

۳- مشتری باید حوصله داشته باشد.

موارد انتخاب یک مدل فرآیند برای تولید نرم افزار به شیوه ی افزایشی :

- وضعیت هایی که در آنها خواسته های اولیه ی نرم افزار به خوبی تعریف شده اند ولی حوضه ی کلی تلاش های به عمل آمده در توسعه ی نرم افزار مانع از یک فرآیند خطی محض می شود.
- ضرورت نیاز به فرآهم کردن سریع مجموعه محدودی از عملکردهای نرم افزار برای کاربران و سپس پالایش و بسط براساس آن عملکردها در نتیجه های بعدی نرم افزار .



نرم افزار **واژه پردازی** که با استفاده از الگوی افزایشی توسعه یافته است، ممکن است اعمالی از قبیل:

مدیریت فایل، تولید و ویرایش مستندات در **نسخه ی اول**

قابلیتهای پیچیده تر ویرایشی و تولید مستندات در **نسخه دوم**

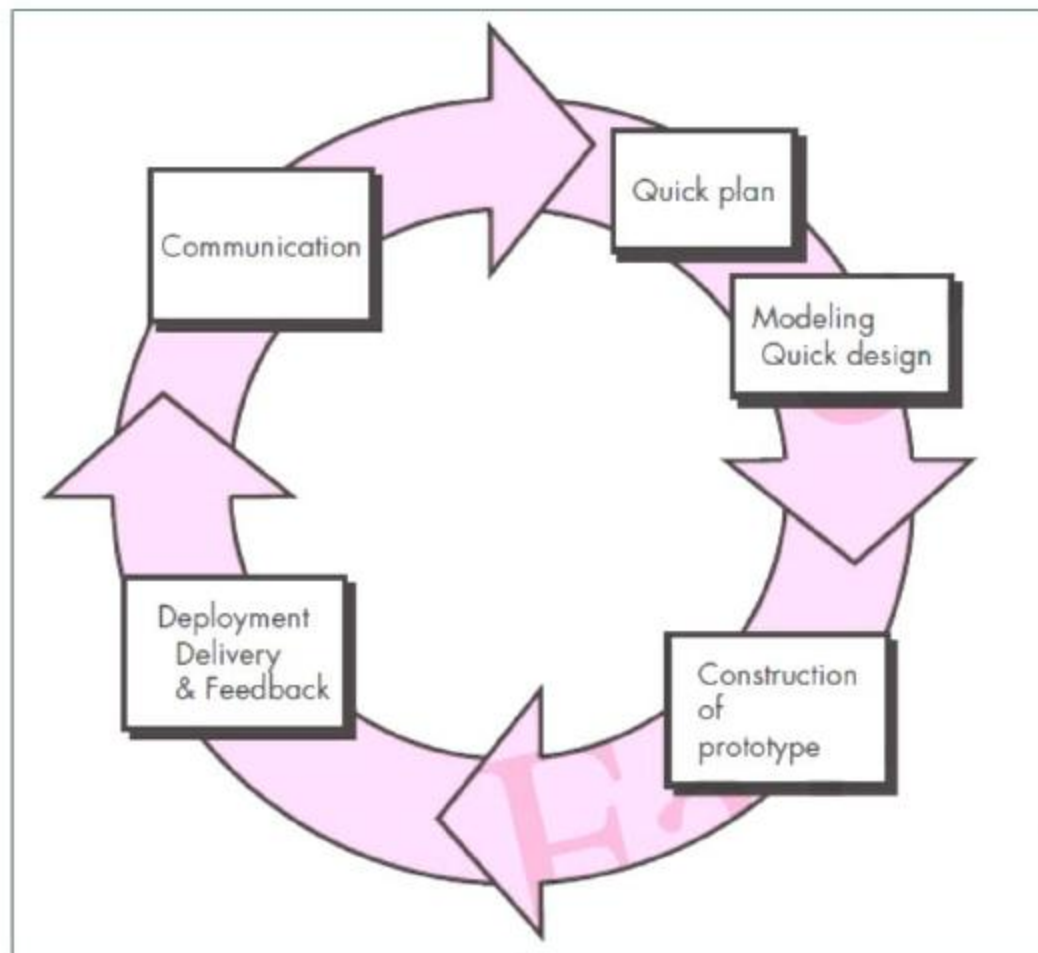
چک کردن املا و دستور در **نسخه سوم**

قابلیت های پیشرفته ی صفحه بندی را در **نسخه چهارم**

تحويل دهد.

مدل های فرآیند تکاملی

ساخت نمونه اولیه



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 پروژه ها نخستین سیستمی که ساخته می شود 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 چاره ای جز شروع دوباره وجود ندارد و باید نسخه دیگری ساخت که این مشکلات در آن حل شده باشد.

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

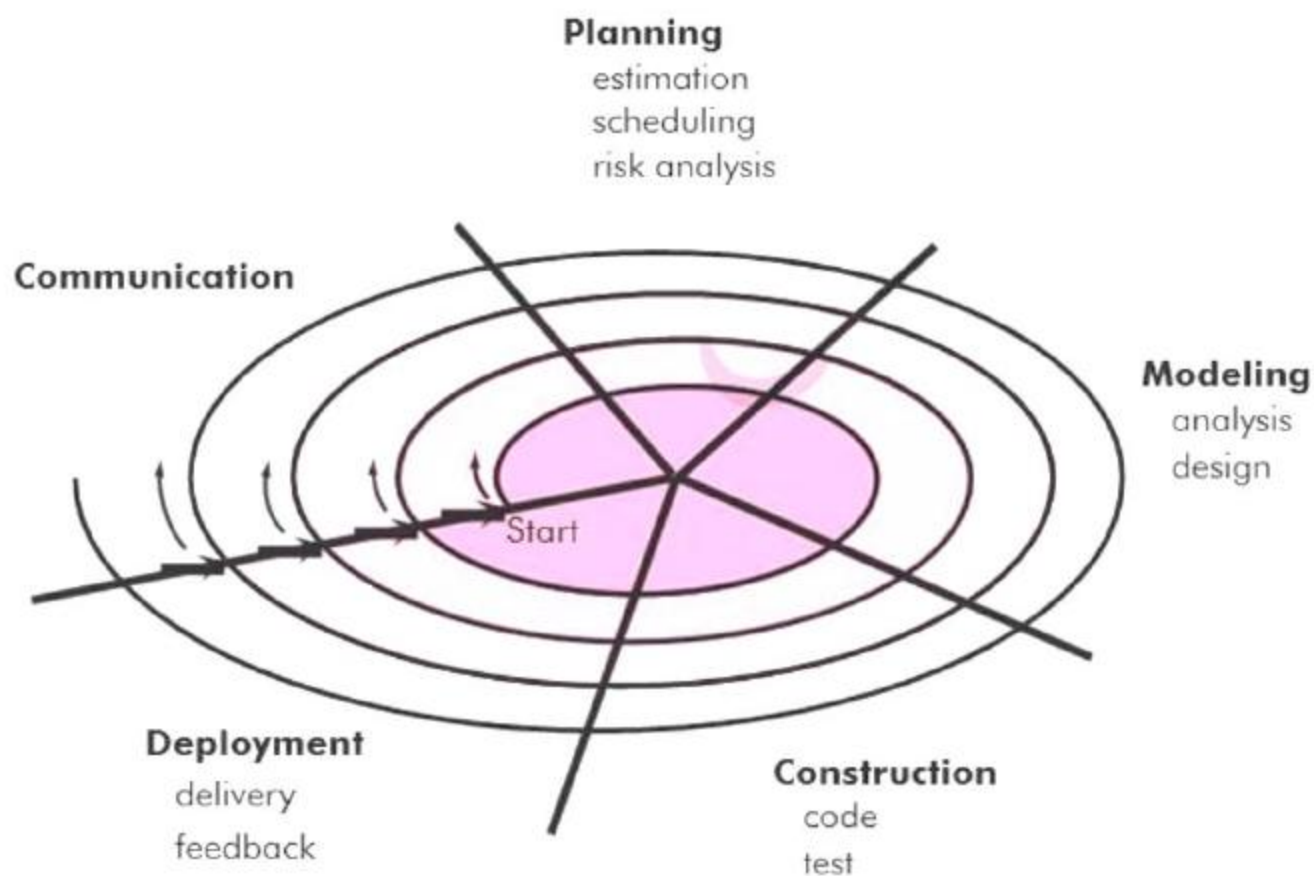
بعضی نمونه های اولیه طبیعتی تکاملی دارند و به تدریج به سیستم واقعی تکامل می یابد.

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.

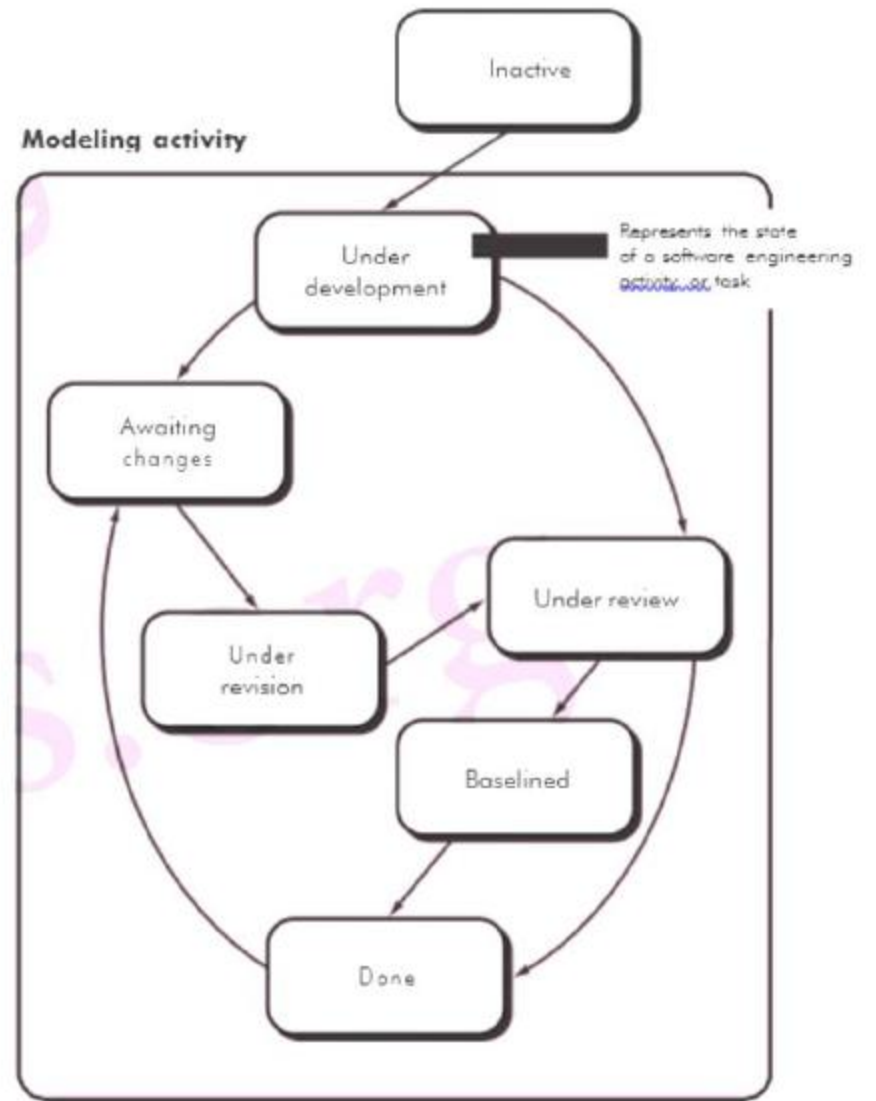
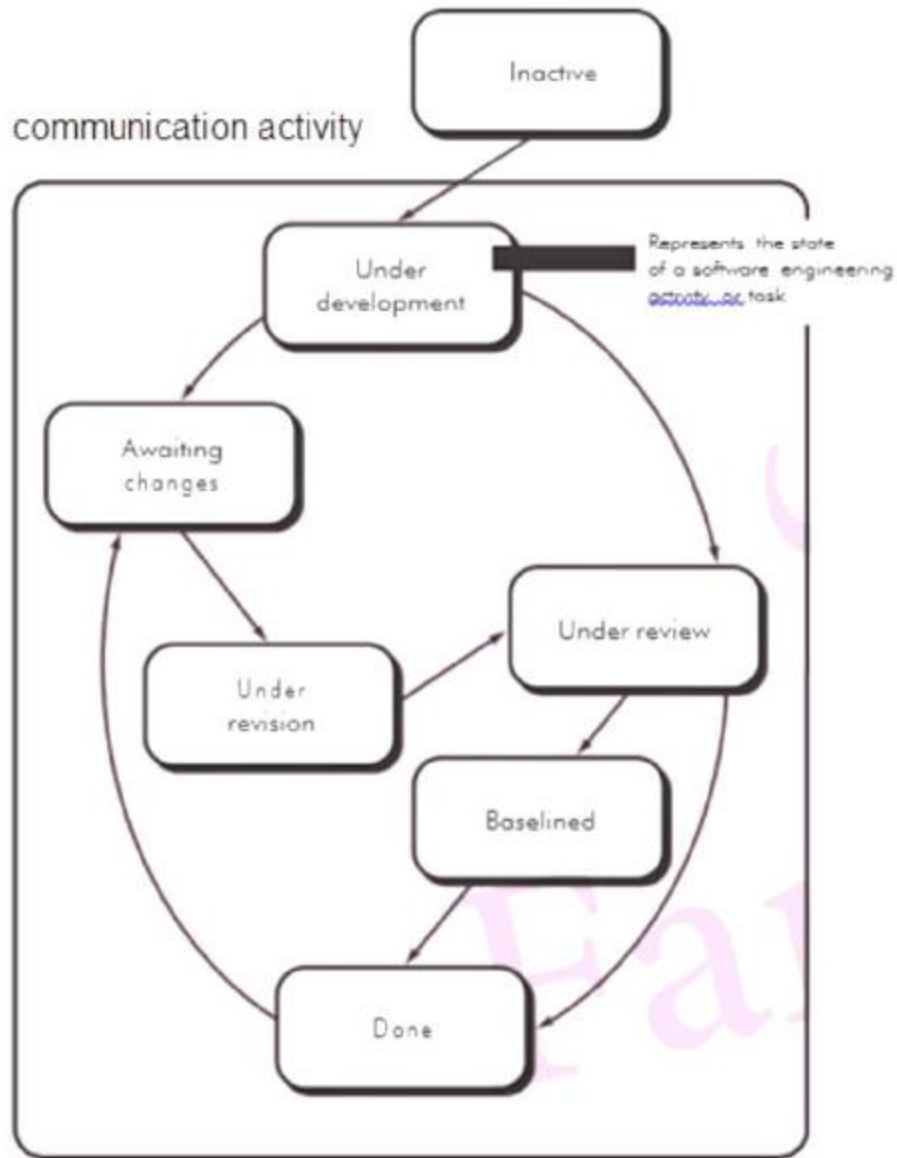
برخلاف سایر مدل های فرآیند کلاسیک که با تحویل نرم افزار پایان می یابند، مدل مارپیچی را می توان طوری تطبیق داد که در سرتاسر عمر نرم افزار کامپیوتری قابل به کارگیری باشد.

مدل مارپیچی یک روش واقع گرا برای توسعه نرم افزارها و سیستم هایی در مقیاس انبوه است.

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

مدل توسعه 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.

نقاط ضعف فرایندهای تکاملی

۱- تهیه نمونه اولیه به دلیل قطعی نبودن تعداد چرخه های لازم برای ساخته شدن محصول، برای برنامه ریزی پروژه ایجاد مشکل می کند.

۲- حداکثر سرعت تکامل را تعیین نمی کنند.

۳- در فرایندهای نرم افزاری ، **انعطاف پذیری** و **بسط پذیری** باید بیش از **کیفیت بالا** مورد توجه قرار گیرد.

شکل طرحی از یک فعالیت با مدل توسعه همروند ارائه می دهد . 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 می شود ولی اگر مشتری متذکر شود که تغییراتی در خواسته ها باید صورت گیرد فعالیت تحلیل از حالت تحت توسعه به حالت انتظار تغییرات می رود.

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

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

مدل های فرایند تخصصی

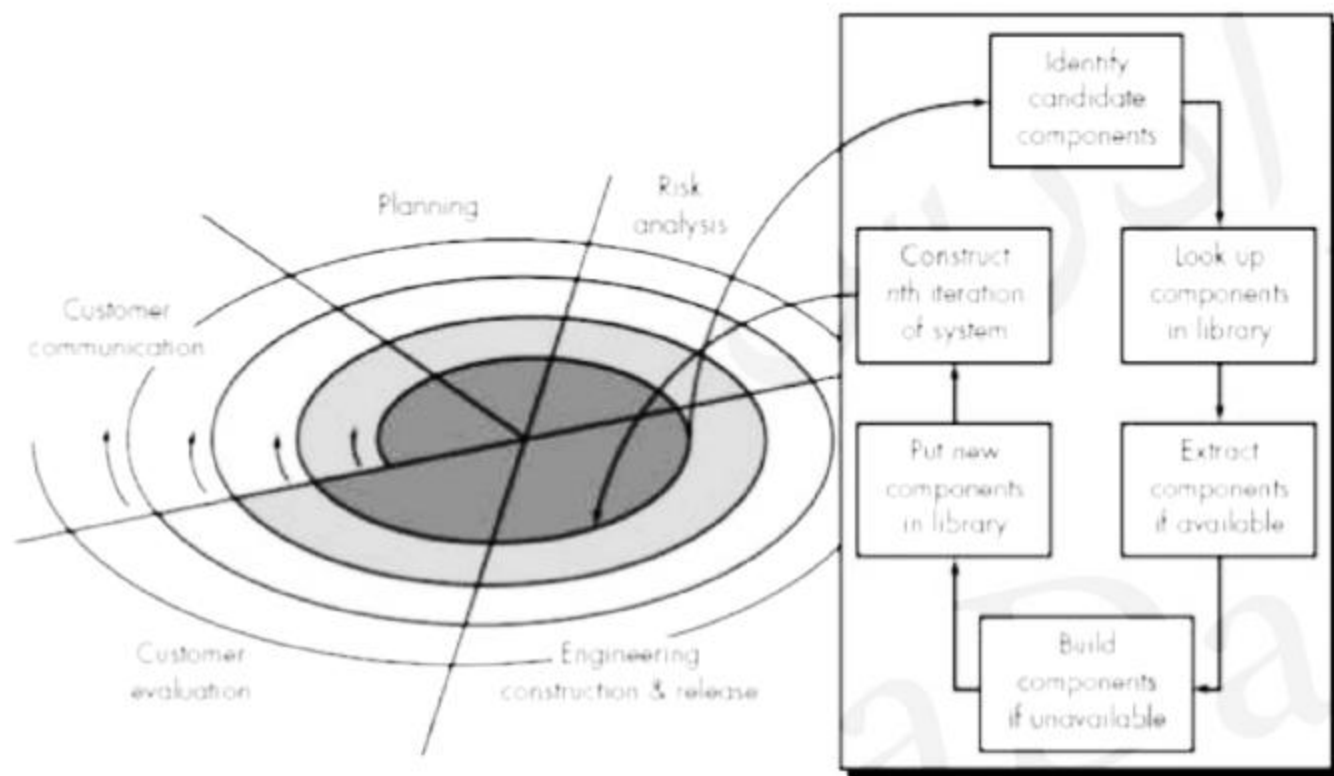
SPECIALIZED PROCESS MODELS

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 معمولاً 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



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.

توسعه مبتنی بر مؤلفه شامل موارد زیر است:

- ۱- محصولات مبتنی بر مؤلفه های موجود از نظر دامنه ی کاربرد مورد نظر بررسی و ارزیابی می شوند.
- ۲- مسائل مربوط به انسجام مؤلفه ها در نظر گرفته می شود.
- ۳- برای چیدمان مؤلفه ها یک معماری نرم افزار طراحی می شود.
- ۴- مؤلفه ها در این معماری قرار داده می شوند.
- ۵- آزمونی جامع برای حصول اطمینان از عملکرد درست بدست می آید.

The Formal Methods Model مدل روش های 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.

توسعه ی 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.

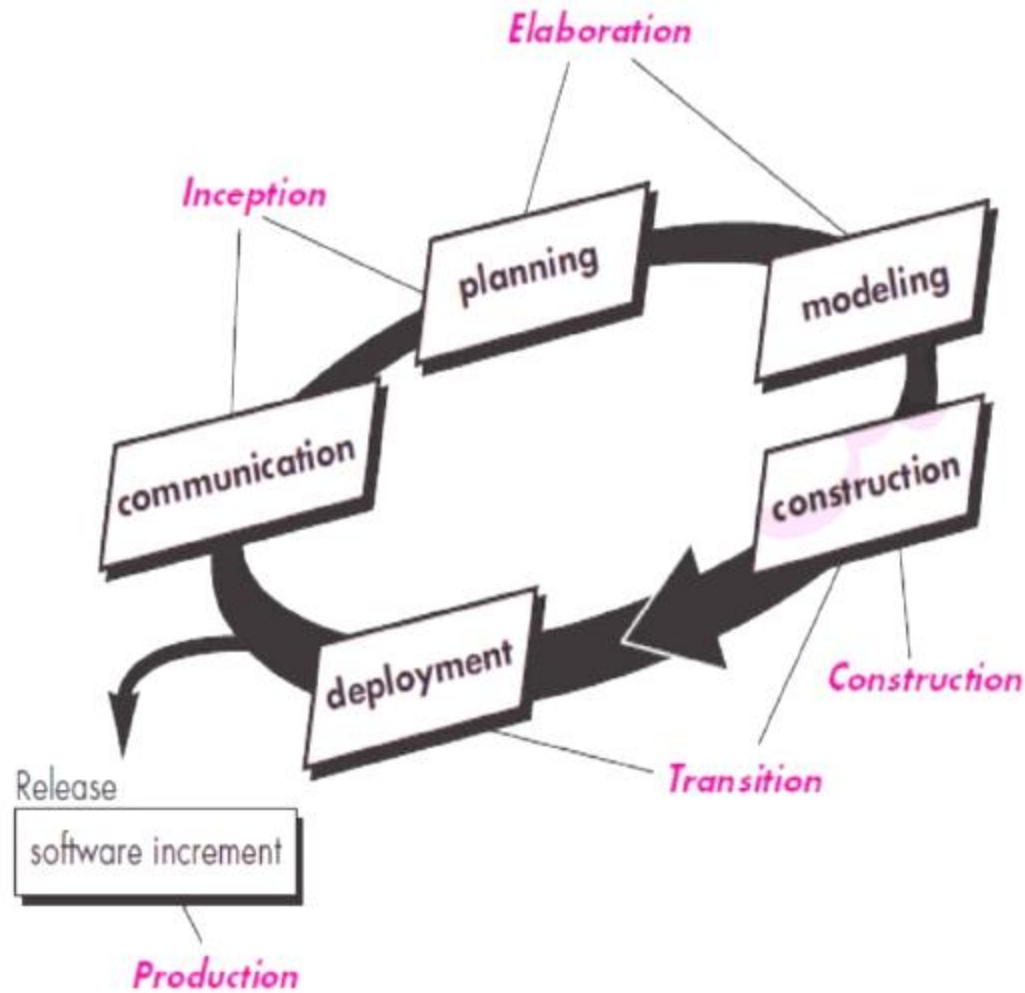
فرآیند یکپارچه (UP) از جهاتی تلاشی برای گرد هم آوردن بهترین ویژگی ها و خصوصیات مدل های فرآیند سنتی است، ولی آنها را به شیوه ای مشخص می کند که بسیاری از بهترین اصول توسعه نرم افزار چابک را پیاده سازی می کند.

در فرایند یکپارچه اهمیت برقراری ارتباط با مشتریان و روش های ساده برای توصیف دیدگاه مشتریان (use case) یک سیستم به خوبی درک می شود.

در این فرایند بر اهمیت نقش معماری نرم افزار تاکید می شود.

در این فرایند، یک جریان فرایند مبتنی بر تکرار و افزایشی پیشنهاد می شود.

در اوایل دهه ۱۹۹۰، کار روی یک روش یکپارچه آغاز شد که بهترین ویژگی های هر کدام از روش های طراحی و تحلیل شی گرا را تلفیق می کرد و ویژگی هایی از سایر کارشناسان در مدل سازی شی گرا بر آن منطبق می ساخت. نتیجه، زبان مدل سازی یکپارچه (UML) است که حاوی یک نمادگذاری قدرتمند برای مدلسازی و توسعه سیستم های شی گرا است.



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 یوزکیس 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.

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 (PSP) Personal Software Process

فرایند نرم افزار تیمی (TSP) Team Software Process

فرآیند نرم افزار شخصی

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**

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.

فعالیت های چارچوبی تعریف شده در TSP

۱- آغاز پروژه

۲- طراحی سطح بالا

۳- پیاده سازی

۴- انسجام دهی

۵- آزمون

انتخاب یک مدل فرایند

SAFEHOME



Selecting a Process Model, Part 1

The scene: Meeting room for the software engineering group at CPI Corporation, a (fictional) company that makes consumer products for home and commercial use.

The players: Lee Warren, engineering manager; Doug Miller, software engineering manager; Jamie Lazar, software team member; Vinod Raman, software team member; and Ed Robbins, software team member.



انتخاب یک مدل فرایند (بخش ۲)

SAFEHOME



Selecting a Process Model, Part 2

The scene: Meeting room for the software engineering group at CPI Corporation, a company that makes consumer products for home and commercial use.

The players: Lee Warren, engineering manager; Doug Miller, software engineering manager; Vinod and Jamie, members of the software engineering team.



پایان فصل ۲