



مهندسی نرم افزار ۱

Software Engineering1

آموزشگاه فنی و حرفه ای شماره دو ارومیه

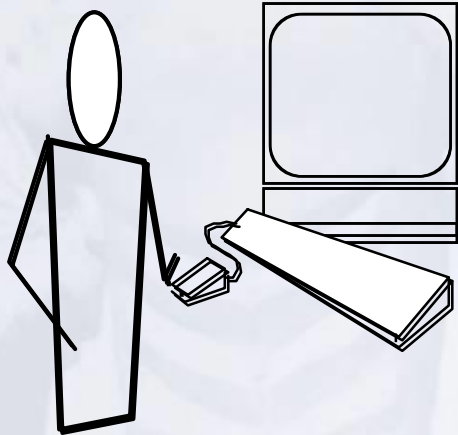
- مهندسی نرم افزار طراحی، برنامه نویسی، توسعه، مستندسازی و نگهداری نرم افزار با بکارگرفتن روش های فنی و عملی از علوم کامپیوتر، مدیریت پروژه، مهندسی، محدوده کاربرد، طراحی رابط، مدیریت تجهیزات دیجیتال و سایر زمینه ها است.
- اصطلاح مهندسی نرم افزار بعد از سال ۱۹۶۸ شناخته شد، طی کنفرانس مهندسی نرم افزار ناتو ۱۹۶۸ (که در گارمیش آلمان برگزار شد) توسط ریاست کنفرانس [F.L. Bauer](#) معرفی شد، و از آن پس به طور گسترده مورد استفاده قرار گرفت.

- تعریف سیستم
- تعریف نرم افزار، چرخه حیات سیستم
- مفاهیم تحلیل و طراحی سیستم‌ها
- سیستم‌های اطلاعاتی ساخت یافته
- امکان سنجی
- نمودارهای جریان داده
- توصیف داده‌ها
- توصیف فرآیندها
- معرفی ابزار: **Power Designer**
- مدیریت پروژه، برنامه ریزی و زمان بندی

تعریف نرم افزار

نرم افزار عبارت است از:

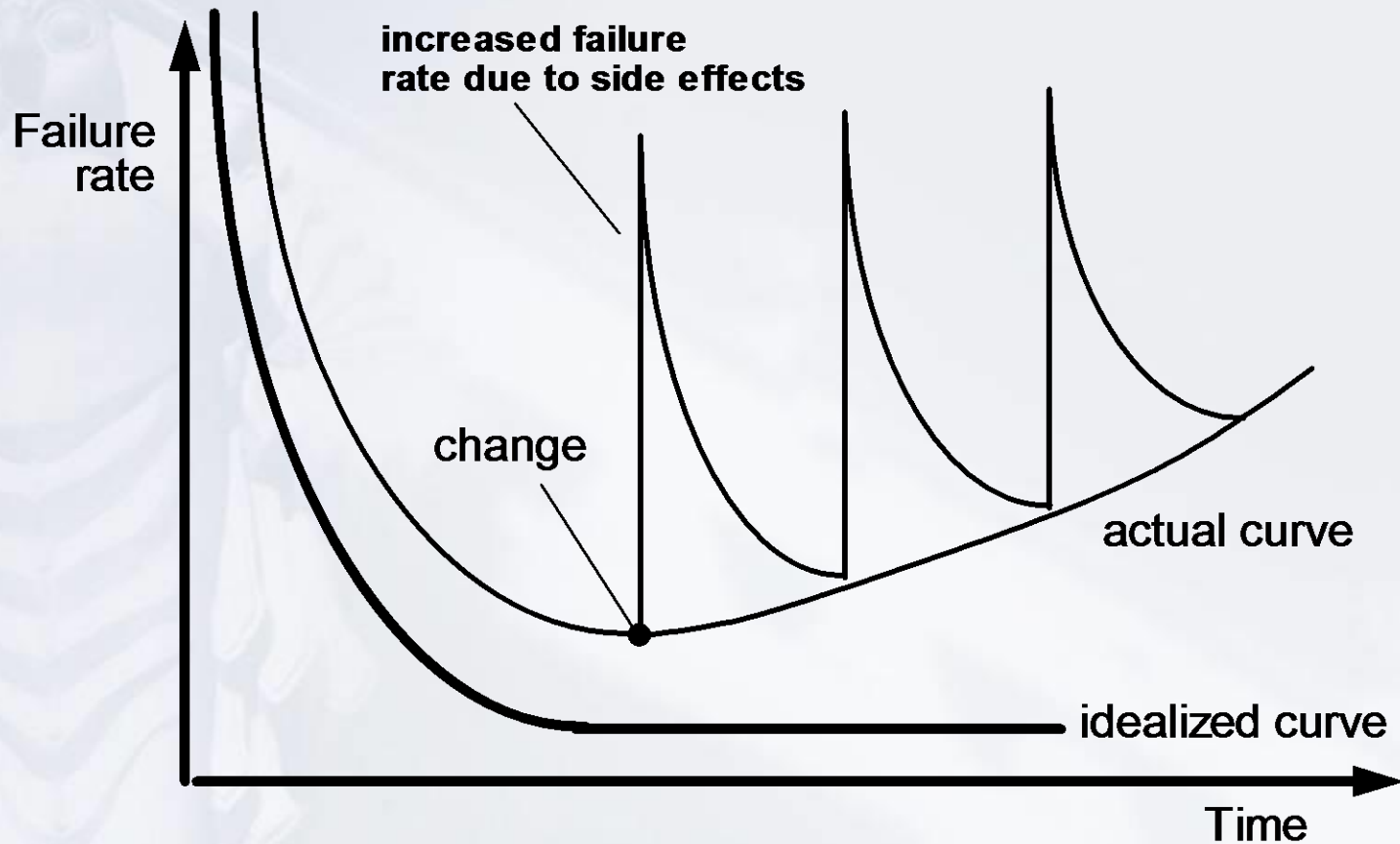
- دستورات (برنامه‌های کامپیوتری) که در صورت اجرا شدن باعث انجام عمل و کارهای خواسته شده می‌شوند،



- ساختمان داده‌هایی که باعث می‌شوند برنامه‌ها به طور مناسبی اطلاعات را دستکاری کنند، و

- مستنداتی که توصیف کننده عملکرد و چگونگی استفاده از برنامه‌ها می‌باشند.

نمودار نرخ شکست نرم افزار



مجموعه‌ای از برنامه‌هایی است که برای دادن سرویس به برنامه‌های دیگر نوشته شده‌اند. نمایش تحلیل و کنترل وقایع دنیای واقعی را در هنگام وقوع بر عهده دارند که اجزای آن عبارتند از ۱- مولفه جمع‌آوری داده‌ها از محیط خارجی ۲- مولفه تحلیل ۳- مولفه کنترل ورودی/خروجی ۴- مولفه نمایش دهنده. (پاسخ بین ۱ میلی ثانیه تا ۱ ثانیه)

پردازش اطلاعات تجاری با دسترسی به یک یا چند بانک اطلاعاتی بزرگ دارای الگوریتم‌های کار با اعداد مانند CAD

در حافظه فقط خواندنی قرار می‌گیرد و برای کنترل محصولات و سیستم‌های صنعتی و مشتری استفاده می‌شود (محصولات هوشمند/اتومبیل/تلفن همراه) کار محدود و مشخصی را انجام می‌دهد

پردازش کلمه، صفحات گسترده، گرافیک کامپیوتری، چندرسانه‌ای، بازی صفحات وب شامل دستورات اجرایی و داده‌ها که توسط مرورگرها بازیابی می‌شوند از الگوریتم‌های غیر عددی برای حل مسائل پیچیده استفاده می‌نمایند.
(سیستم خبره - تشخیص الگو)

طبقه‌بندی نرم‌افزارها :

- نرم‌افزار سیستمی
- نرم‌افزار بی‌درنگ (Real time)
- نرم‌افزار تجاری
- نرم‌افزار مهندسی و علمی
- نرم‌افزار جاسازی شده (Embedded)
- نرم‌افزار کامپیوتر شخصی
- نرم‌افزار وب
- نرم‌افزار هوش مصنوعی

- تغییر اهداف سازمان
- تغییر وظایف سازمان
- تغییر تکنولوژی (سخت افزار، نرم افزار، ارتباطات)
- تغییرات محیطی

نرم افزارهای موجود چرا باید تغییر کنند؟

- باید با محیط های محاسباتی و فن آوری های جدید مطابقت داشته باشند
- نیازمندی های جدید سازمان را برآورده کنند
- باید توسعه داده شوند تا قادر به همکاری با سیستم ها و بانک های اطلاعاتی جدید باشند
- باید در معماری نرم افزار تجدید نظر شود تا در محیط شبکه قابلیت سرویس دهی را داشته باشد

چرخه زیست سیستم

یک دوره زمانی از هنگام نیاز به سیستم تا پایان بهره‌برداری از آن

- چرخه زیست سیستم شامل

- تعریف (برنامه‌ریزی استراتژیک، تحلیل و طراحی مفهومی)
- توسعه (طراحی تفصیلی، ساخت، استقرار)
- اجرا
- نگهداری و بهبود می‌باشد.

نیاز به سیستم جدید	اجرا	توسعه	تعریف	نیاز به سیستم
--------------------	------	-------	-------	---------------



ضوابط ارزیابی نرم افزار :

هدف: تولید برنامه با کیفیت خوب

عوامل موثر در خوب بودن نرم افزار :

عوامل خارجی: توسط کاربر نرم افزار تشخیص داده می شود (اهداف)

عوامل داخلی: برای متخصصین کامپیوتر قابل درک است (ابزار رسیدن به اهداف)

عوامل خارجی :

- صحت برنامه (Correctness)
- استحکام (Robustness) : جوابگویی در شرایط غیرعادی
- قابلیت توسعه (Extendibility)
- قابلیت مصرف مجدد (Reusability)
- سازگاری (Compatibility) : رعایت استاندارد
- قابلیت حمل (Portability): اجرا در سیستم عامل و سخت افزارهای گوناگون
- کارایی (Efficiency): سرعت بالا و حافظه پایین

عوامل داخلی : واحدبندی

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

مدل‌های گوناگون در فرآیند تولید نرم‌افزار :

- برای بر طرف کردن بحران نرم‌افزار سه عامل کلیدی برای کنترل فرایند تولید نرم‌افزار و ایجاد یک پایه و اساس برای ساختن یک نرم‌افزار با کیفیت بالا عبارتند از
- روش‌های مهندسی نرم‌افزار (methods)
 - ابزار مهندسی نرم‌افزار (tools)
 - رویه‌های مهندسی نرم‌افزار: ابزار و روش‌ها را به هم پیوند می‌زنند، ترکیب به کاربردن روش‌ها/ کنترل کیفیت/ پیشرفت پروژه
- مهندسی نرم‌افزار شامل قدم‌هایی است که از این روش‌ها ابزار و رویه‌ها تشکیل شده است.
- مهندسی نرم‌افزار یک نظام مهندسی است که با تمام جنبه‌های نرم‌افزاری**
- محصول از مراحل اولیه تعیین مشخصات تا نگهداری سیستم سر و کار دارد.**

مدل‌های موجود در فرآیند تولید نرم‌افزار

متدلوژی :

به مجموعه‌ای از قواعد و رویه‌ها که به چرخه زیست سیستم ساختار می‌دهد متدلوژی می‌گویند .
فرایند مهندسی نرم‌افزار، مجموعه‌ای از قدم‌های قابل پیش‌بینی برای توسعه نرم‌افزار را مشخص می‌کند.

یک متدولوژی مجموعه‌ای از روش‌ها و توصیه‌ها (Guidelines) می‌باشد که به همراه راهبرد مشخص و طی مراحل مختلف از توسعه سیستم به کار گرفته می‌شود. متدولوژی در واقع یک چارچوب است که ترتیب گام‌به‌گام راهکارهایی را که به توسعه محصول نهایی کمک می‌کنند، مشخص می‌نماید.

یک متدولوژی دارای ابزار تعریف شده و مدل مفهومی می‌باشد و از یک گرامر مشخص استفاده می‌کند.

هر متدولوژی برای انجام کارها از تعدادی ابزار و تکنیک استفاده می‌نماید. برای مثال مدل شی‌گرا و یا مدل ساخت‌یافته در توسعه نرم‌افزار دو متدولوژی توسعه نرم‌افزار هستند.

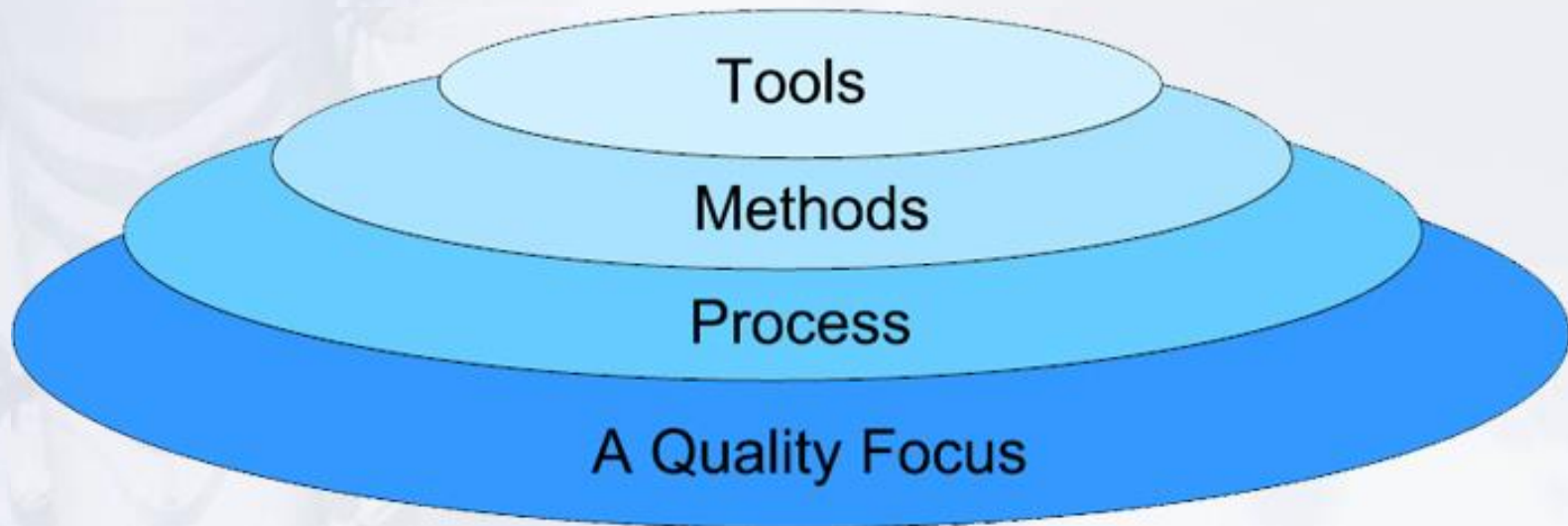
مدل فرآیند

فرآیند مهندسی نرمافزار مجموعه‌ای از قدم‌های قابل پیش‌بینی برای توسعه نرمافزار را مشخص می‌کند.

مدل فرآیند نرمافزار قدم‌ها، و استراتژی توسعه نرمافزار، فرآیند و روش می‌باشد.
از مدل‌های معروف می‌توان به مدل‌های
آبشاری،
افزایشی،
و چرخشی
اشاره کرد.

تفاوت‌های متدولوژی و مدل فرآیند

متدولوژی، روش طی کردن قدم‌هایی است که مدل فرآیند تعریف می‌کند. تکنولوژی مهندسی نرم‌افزار یک تکنولوژی لایه‌ای است و متدولوژی بر روی لایه فرآیند قرار دارد.

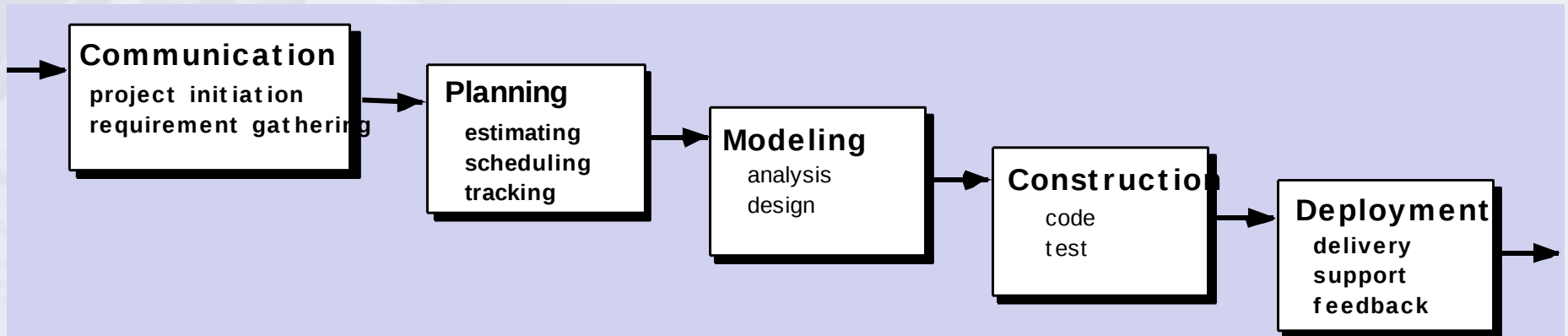


تکنیک‌ها و ابزارها

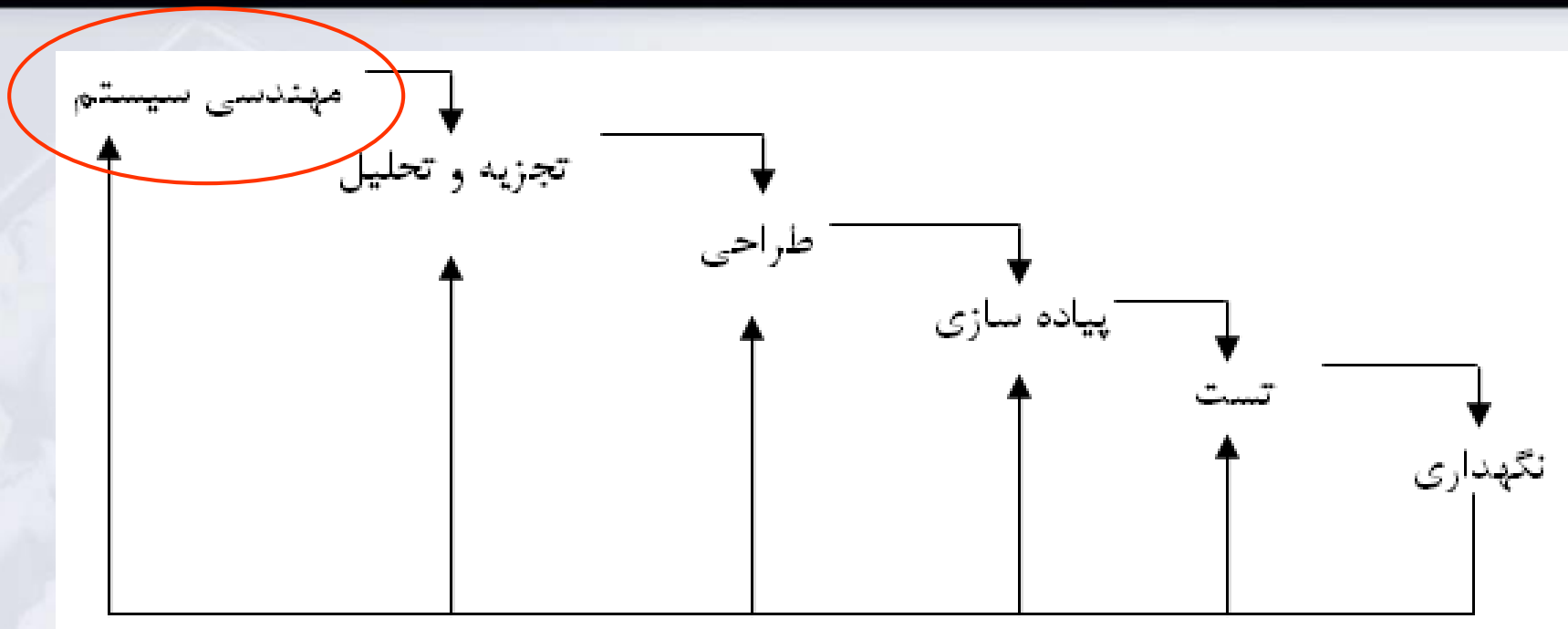
تکنیک‌ها روش‌هایی هستند که اعضای تیم با دنبال کردن آنها اطمینان خواهند داشت کار از ابتدا تا انتها به طور کامل و جامع انجام خواهد شد.

ابزارها برنامه‌های کامپیوتری هستند که استفاده از تکنیک‌های خاصی را آسان می‌نمایند. مانند ابزار Rational Rose که در تکنیک‌های تحلیل و طراحی بسیار مفید است و یا NUnit که مربوط به تکنیک‌های تست نرم‌افزار می‌شود و یا صدها برنامه کاربردی دیگر. غیر از ابزارهایی که به ما در استفاده از تکنیک‌ها یاری می‌بخشند، ابزارهایی وجود دارند که اهداف دیگری مانند سرعت بخشیدن به تولید محصول را دنبال می‌کنند. مانند تولید کننده کد (Code Generator)، ابزار مهندسی معکوس و ...

۱- مدل آبشاری



۱- مدل آبشاری

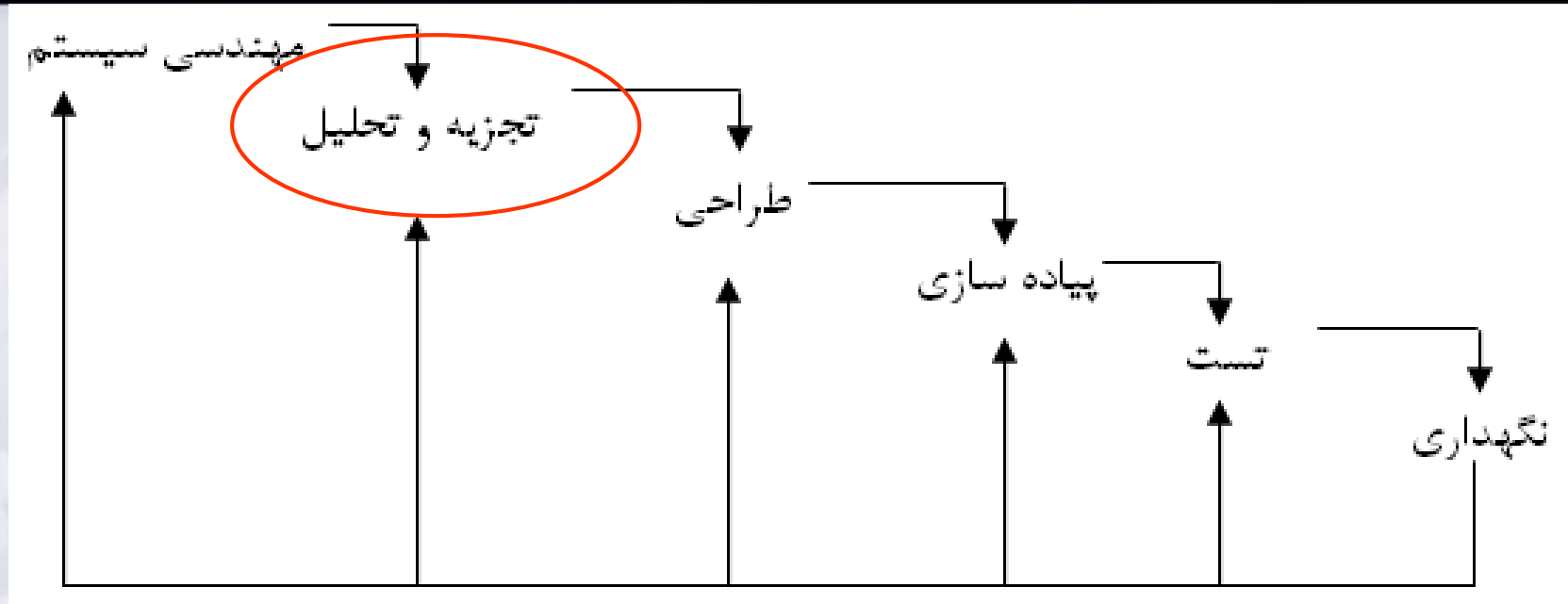


مهندسی سیستم:

از آنجاییکه نرم افزار همیشه قسمتی از یک سیستم بزرگتر است، کار از مشخص کردن نیازمندی های کل سیستم آغاز می شود و سپس زیرمجموعه ای از این نیازمندی ها 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 گاه پایگاه داده ها است، داشتن این دیدگاه از سیستم، یک امر اساسی است.

۱- مدل آبشاری



جمع آوری نیازمندی‌های مشخصاً مربوط به نرم‌افزار برای فهمیدن چگونگی برنامه‌هایی که باید ساخته شوند. مهندس نرم‌افزار (تحلیلگر) بایستی

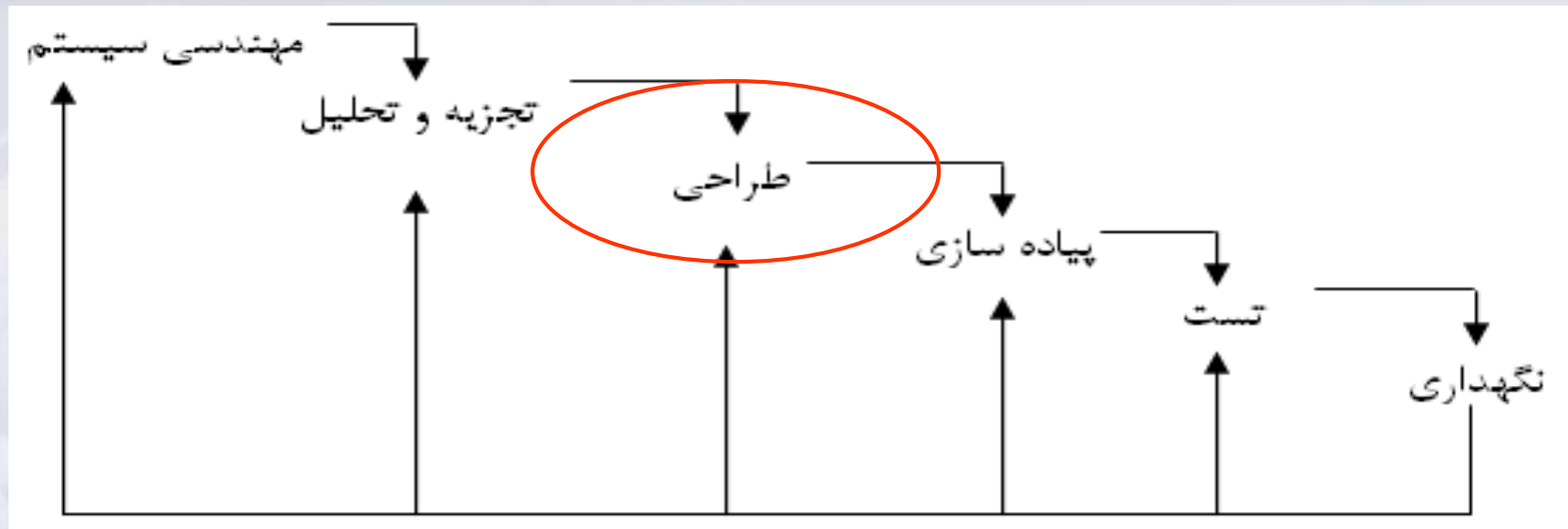
- دامنه اطلاعات

- عملیات مورد نظر (ورودی‌ها - خروجی‌ها و پردازش‌ها)

- واسط‌ها

را بداند. نیازمندی‌های سیستم و نرم‌افزار مستندسازی شده و با مشتری بازنگری می‌شود.

۱- مدل آبشاری

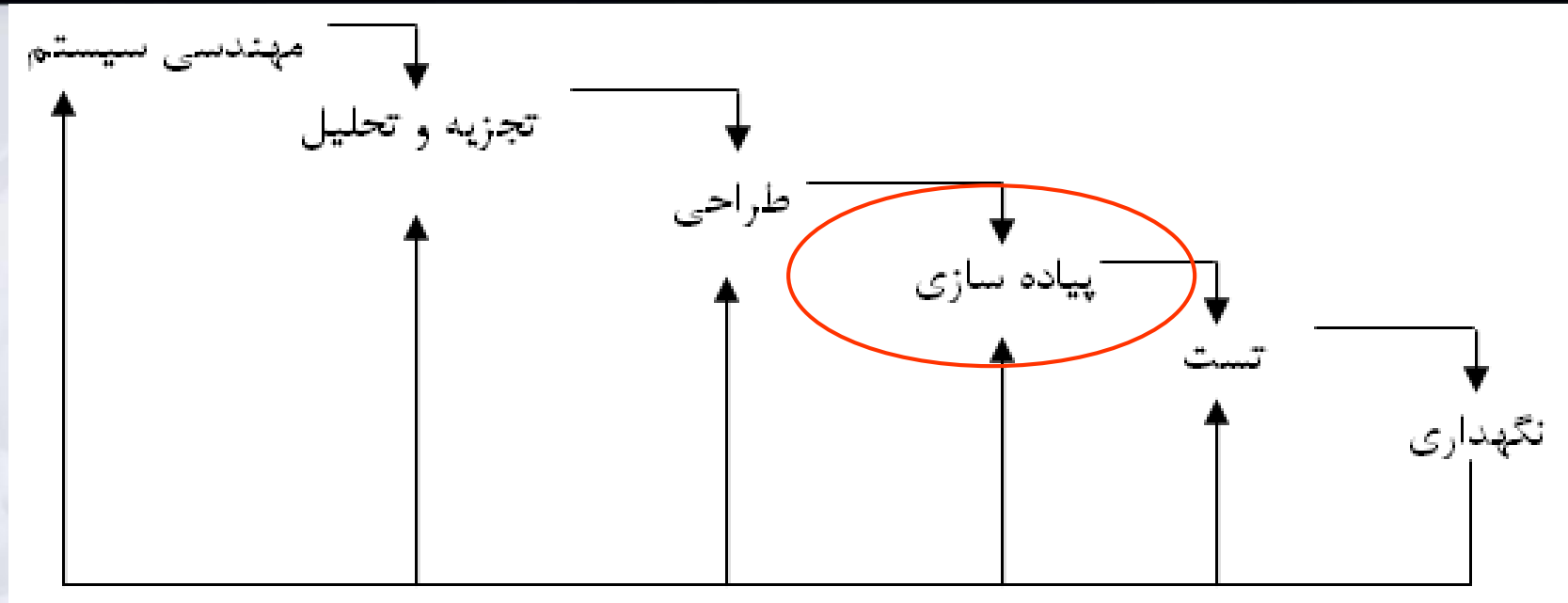


طراحی نرم افزار یک فرایند چند مرحله‌ای است که روی سه مشخصه متفاوت از برنامه تاکید دارد:

- ساختمان داده‌ها
- معماری نرم افزار
- جزئیات رویه‌ها

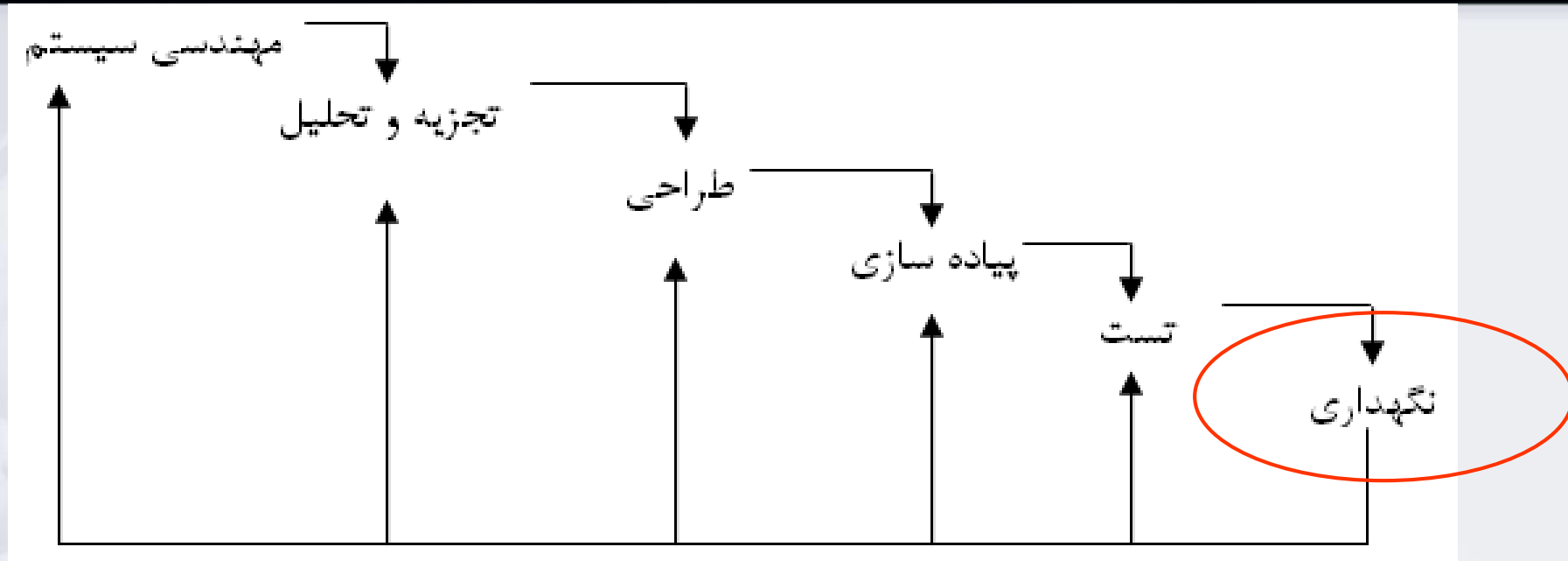
در فرایند طراحی نیازمندی‌ها تبدیل به نمایشی از نرم افزار می‌شوند تا قبل از به کد درآوردن، قابل ارزیابی باشند. طراحی نیز بایستی مانند نیازمندی‌ها مستندسازی گردد.

۱- 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.

۱- مدل آبشاری



نرم افزار بدون شک پس از تحویل به مشتری دچار تغییر می شود (به استثنای نرم افزارهای توکار). تغییرات به علت

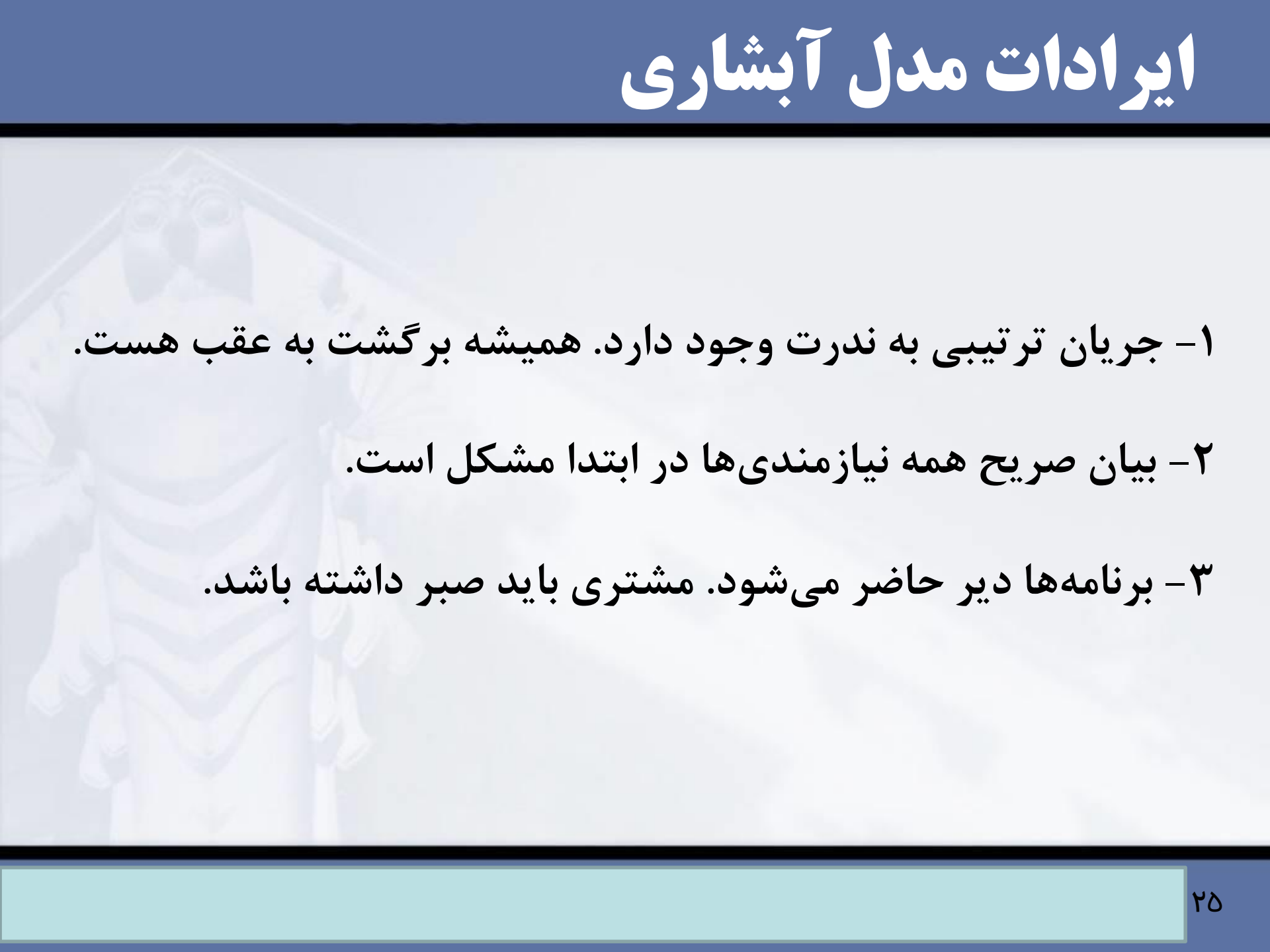
- خطاها

- 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- برنامه‌ها دیر حاضر می‌شود. مشتری باید صبر داشته باشد.

۲-تولید سریع نرم افزار ، RAD

Rapid Application Development

یک مدل تولید نرم افزار خطی ترتیبی است که روی چرخه کوتاه تولید تاکید زیادی دارد. نیازها بایستی کاملاً شناخته شده و حوزه کار محدود باشد.

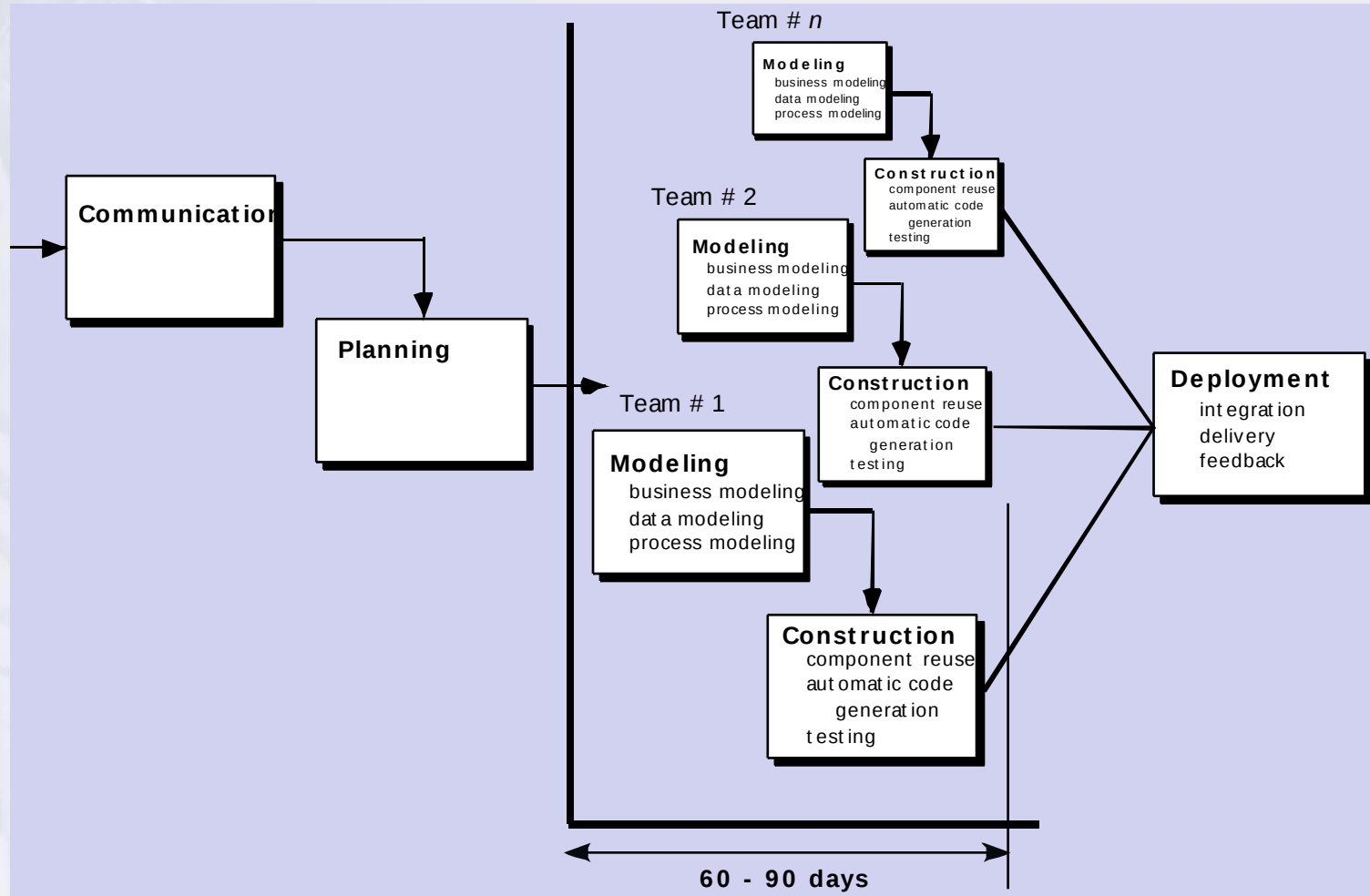
- از تکنیک های نسل چهارم و مؤلفه های آماده استفاده می گردد

- برای پروژه هایی که قابل تقسیم هستند به کار می آید

مشکلات :

- نیاز به نیروی انسانی کافی
- تعهد تولید کننده و مشتری برای کامل کردن سیستم در زمان کوتاه لازم است

٢- تولید سریع نرم افزار ، RAD



۲- تولید سریع نرم افزار ، RAD

موارد نامناسب برای استفاده از RAD:

- مواردی که نیاز به کارایی بالا می باشد.
- مواردی که امکان واحد بندی مناسب سیستم نباشد.
- مواردی که ریسک تکنیکی بالا باشد (هنگام استفاده از یک تکنولوژی جدید یا ...)

۳- نمونه سازی 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

paper prototype

(۱) یک نمونه روی کاغذ

working prototype

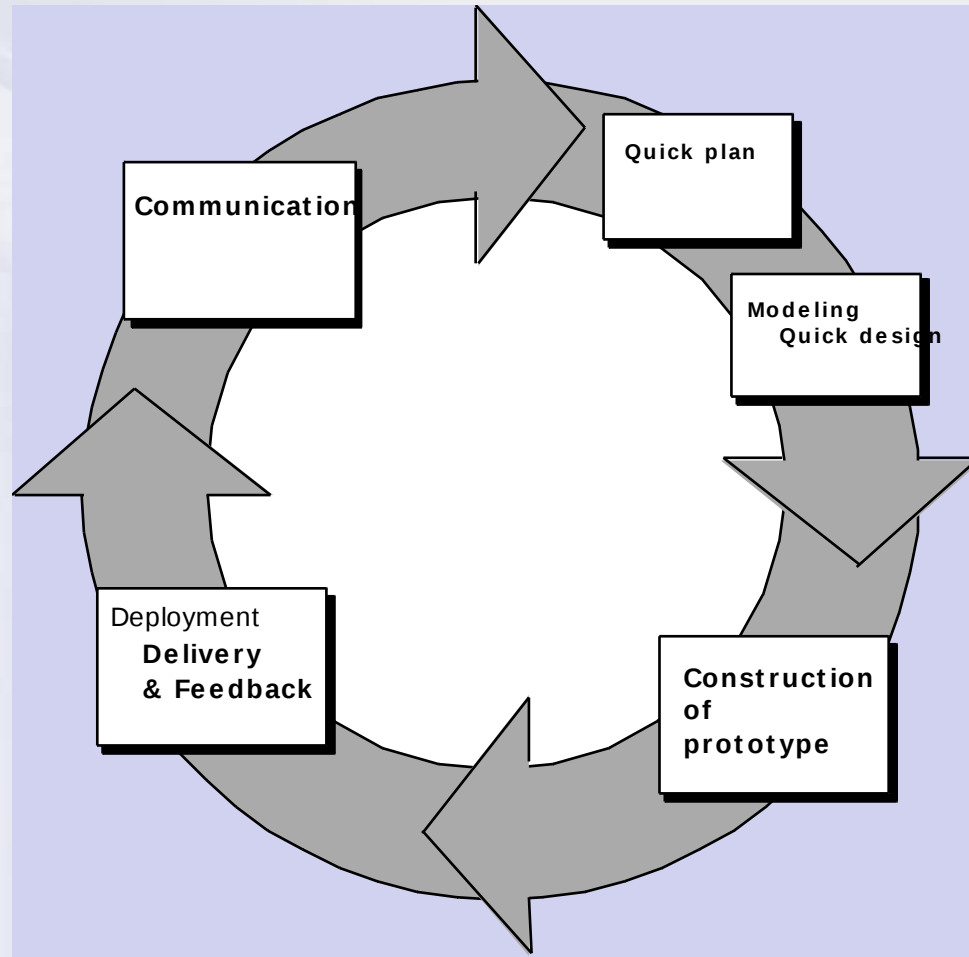
(۲) یک نمونه کاری

existing prototype

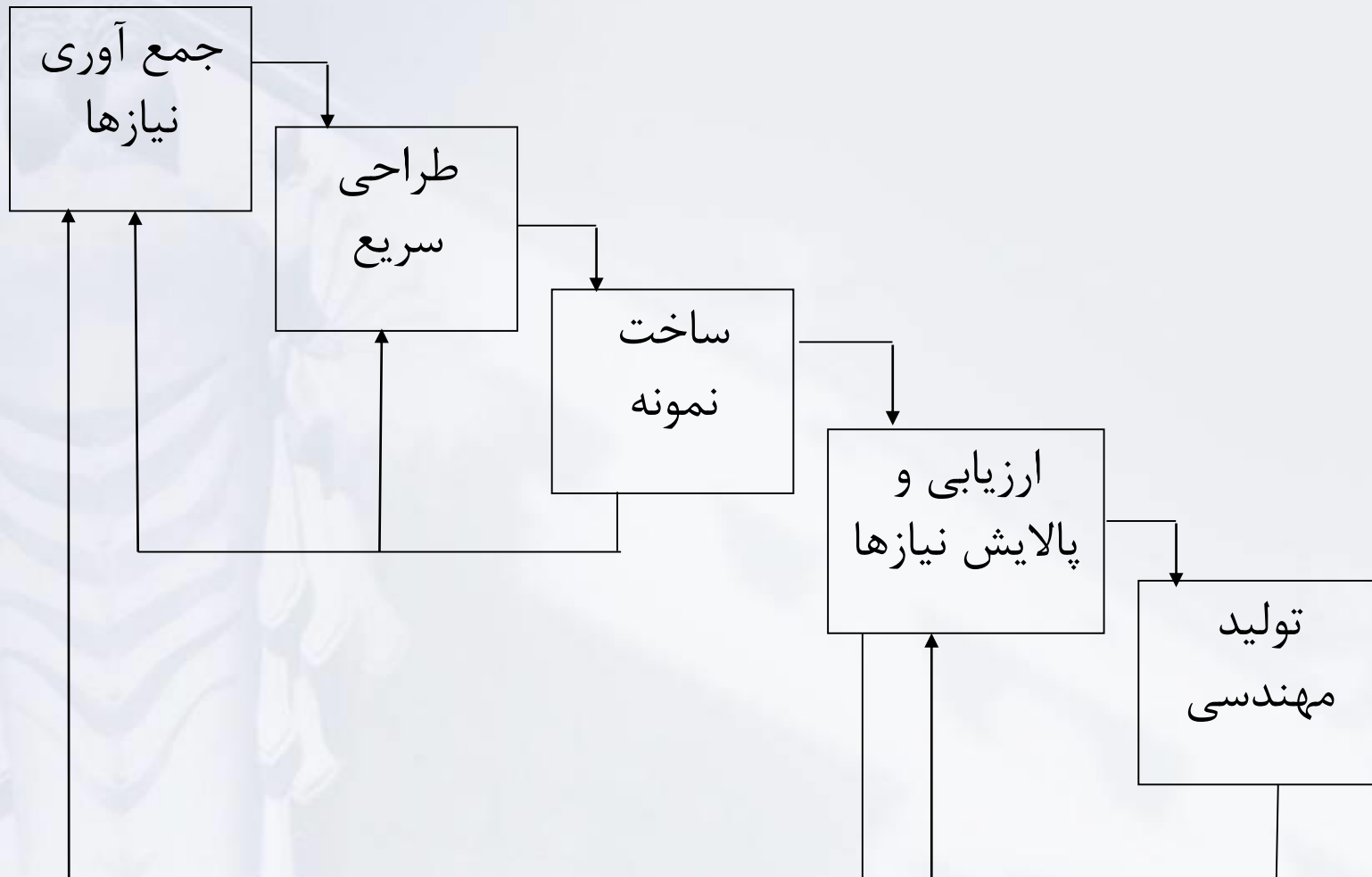
(۳) یک برنامه موجود

نمونه باید دور انداخته شود

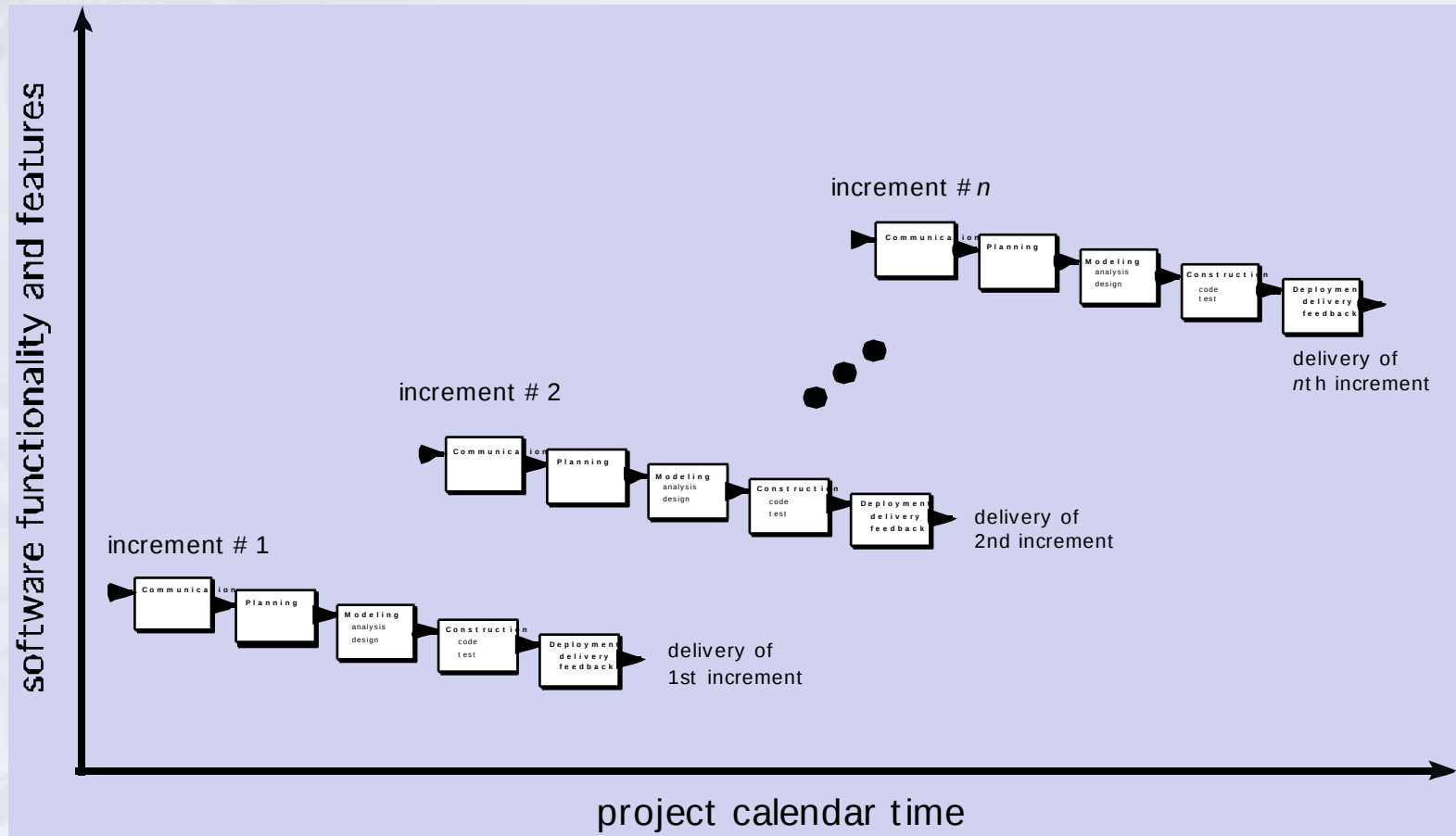
۳- نمونه سازی Prototyping



۳- نمونه سازی Prototyping

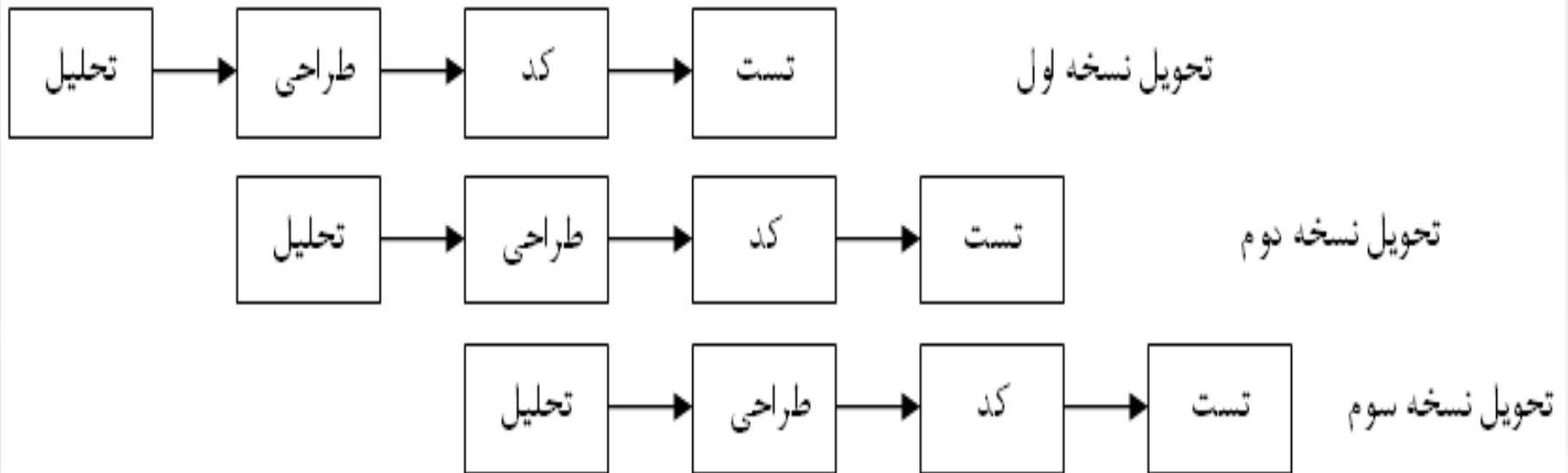


۴- مدل‌های تکاملی: مدل افزایشی



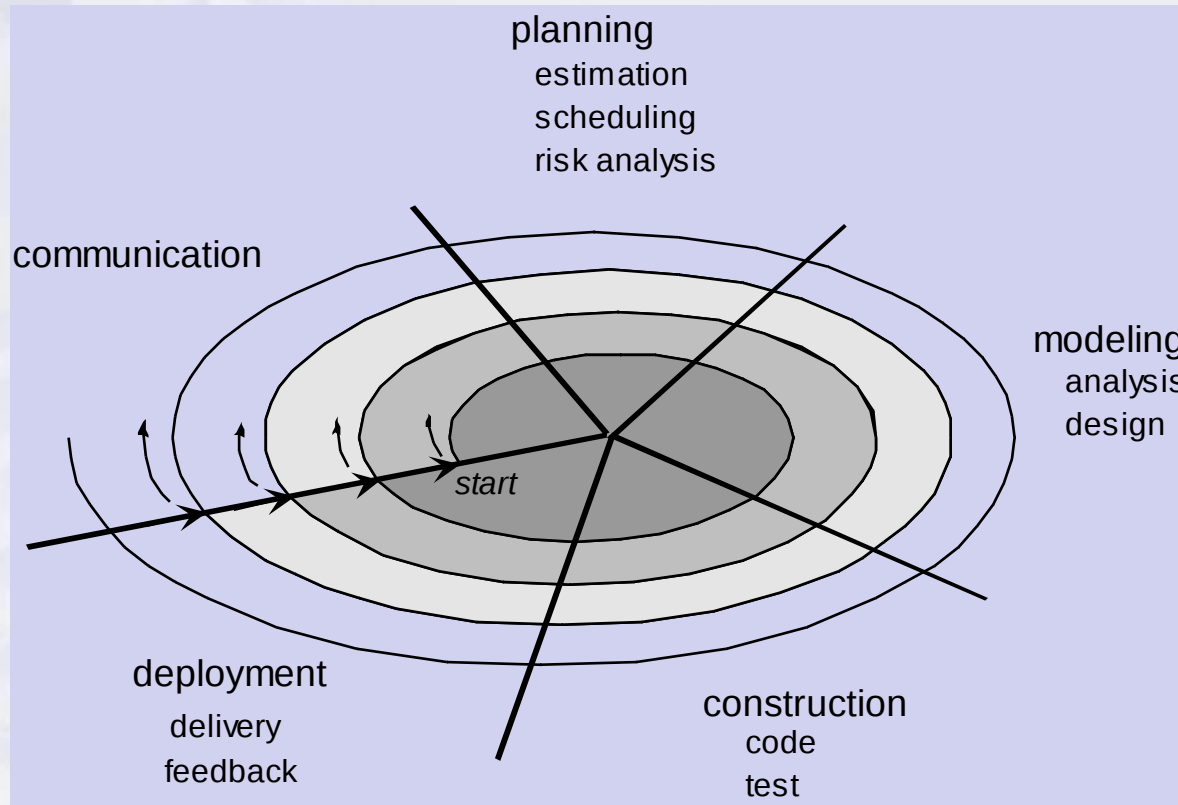
۴- مدل افزایشی

در این مدل مراحل تولید ترتیبی با فلسفه گردشی نمونه‌سازی ترکیب شده است. هر رشته ترتیبی عملیات، یک افزایش قابل تحویل را تولید می‌کند.

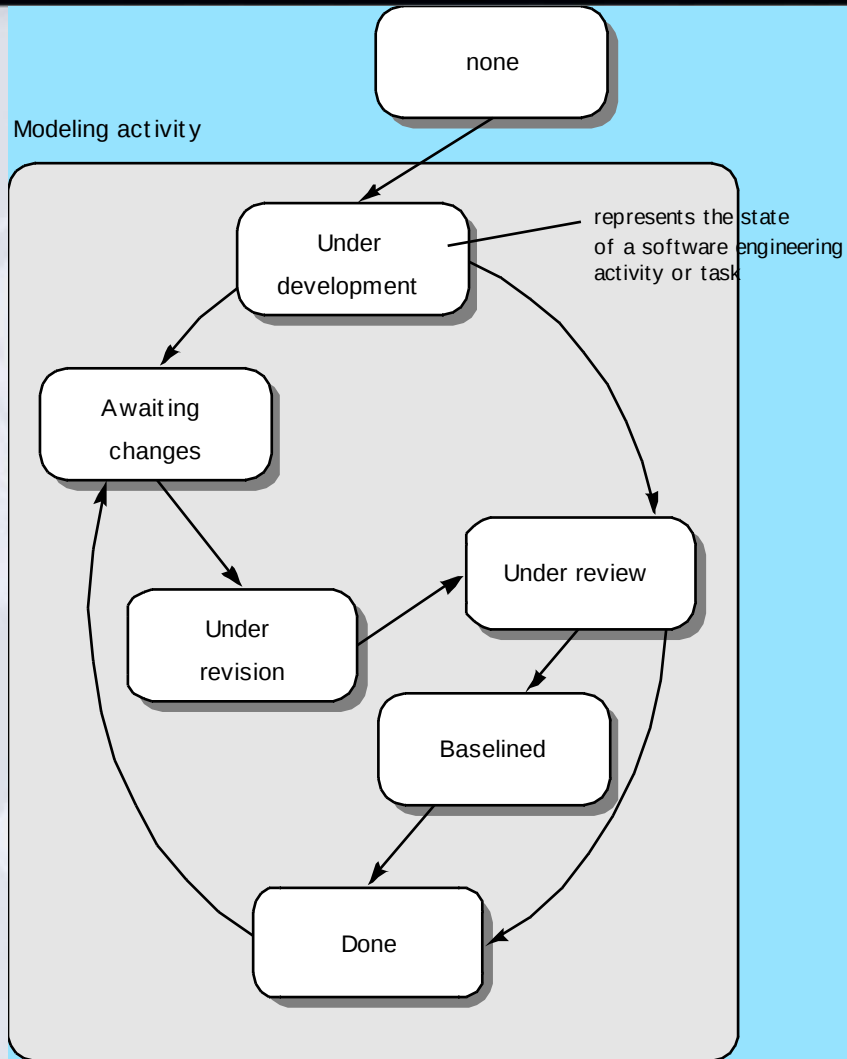


۵- مدل‌های تکاملی: مدل حلزونی

توسعه سیستم به صورت افزایشی و بازگشتی



۶- مدل‌های تکاملی: مدل تولید همزمان



توسعه به صورت یک فرایند سیستماتیک و مرحله‌بندی و برچسب‌گذاری هر بخش در هر مرحله، تقسیم سیستم به بخش‌های مختلف و تقسیم نیروها در بین پروژه‌های مختلف برای اجرای این بخش‌ها به صورت همزمان

مدل‌های دیگر

۷- مدل تولید با استفاده از مولفه‌ها

۸- مدل روش‌های رسمی (فرمال): بکارگیری مدل‌ها و مفاهیم و سیستم در توسعه سیستم

۹- تکنیک‌های نسل چهارم: بکارگیری ابزارهای گرافیکی و ابزارهای مهندسی نرم‌افزار

۱۰- Unified Process: با استفاده از UML

اصول محوری مهندسی نرم افزار

- ایجاد ارزش برای مشتری و کاربر
- ساده سازی
- چشم انداز نگهداری محصول و پروژه
- آنچه که شما تولید می کنید، دیگران مصرف خواهند کرد
- آینده نگر بودن
- طراحی پیشاپیش برای استفاده مجدد
- اندیشه

کارهای مهندسی نرم افزار

فرایندهای عمومی مهندسی نرم افزار

-ارتباط

- طرح ریزی

- مدل سازی

- جاری سازی (استقرار و اجراء سیستم)

کارهای ارتباطی – اصول

- گوش کردن
- آمادگی قبل از ایجاد ارتباط
- تسهیل ارتباط
- بهترین ارتباط، ارتباط رودررو است
- یادداشت برداری و ثبت تصمیمات
- تشریک مساعی با مشتری
- تمرکز داشتن
- ترسیم شکل زمانیکه موضوعات شفاف نیست
- حرکت داشتن
- برای برد طرفین بهترین کار مذاکره است

کارهای ارتباطی – مرحله آغازین

- طرفین باید به صورت فیزیکی با یکدیگر تعامل داشته باشند
- اطمینان از ارتباط فعال
- ایجاد سیستم‌های قوی و بومی شده
- استفاده از ساختار تیمی مناسب
- مجموعه وظایف کوتاه شده
- مشخص کنید که با چه کسی نیاز است صحبت شود
- بهترین مکانیزم ارتباطی را تعریف کنید
- اهداف فرعی و کلی را تعریف نموده و محدوده را مشخص کنید
- جزئیات بیشتری دریافت نماید
- دانستن سناریوهای تعریف شده توسط ذینفعان برای استفاده
- استخراج و اقتباس جنبه‌ها و کارکردهای اصلی
- بازنگری نتایج همراه با ذینفعان

کارهای طرح ریزی – اصول

- درک محدوده پروژه
- درگیر کردن مشتری (و سایر ذینفعان)
- مشخص کنید که طرح ریزی قابل تکرار است
- برآورد بر اساس دانسته‌ها
- در نظر گرفتن ریسک‌ها
- واقع گرائی - واقع گرا بودن
- تنظیم امور با توجه به طرح
- تعریف کنید که چگونه کیفیت قابل تحصیل خواهد بود
- تعریف کنید که چگونه تغییرات قابل تطبیق دادن است
- ردیابی آنچه که طراحی شده است

کارهای طرح ریزی – مرحله آغازین

(Bohem) پرسش‌های بوهم را مطرح کنید

- چرا سیستم باید توسعه یابد؟
- چه چیزی قرار است انجام شود؟
- چه زمانی این کار انجام خواهد شد؟
- چه کسی مسئول است؟
- کجا آنها استقرار یافته‌اند (بصورت سازمانی)؟
- چگونه کار از نظر فنی و مدیریتی انجام خواهد شد؟
- چه مقدار از هر منبعی مورد نیاز است؟

کارهای طرح ریزی

مجموعه وظایف اختصاری

- محدوده پروژه را دوباره ارزیابی کنید (ارزیابی دوباره محدوده پروژه)
- ارزیابی ریسک‌ها
- ارزش‌گذاری عملیات / جنبه‌ها
- ایجاد یک طرح مستحکم
- تعداد افزایشات نرم‌افزاری
- زمانبندی کلی
- زمان تحویل برای افزایشات
- ایجاد طرحی ساده برای اولین افزایش
- ردیابی پیشرفت (کنترل کیفیت)

کارهای مدل سازی

- ما مدل ها را ایجاد می کنیم تا درک بهتری از وضعیت واقعی که قرار است ساخته شود ایجاد نمائیم
- مدل های تحلیلی نیازهای مشتری را به وسیله نمایش نرم افزار در سه حوزه مختلف ارائه می دهند. این سه حوزه عبارتند از:
 - حوزه اطلاعات، حوزه عملیاتی و حوزه رفتاری
- مدل های طراحی ارائه دهنده ویژگی ها و خصوصیت های نرم افزار بوده که این خصوصیات به طراحان کمک می کنند تا به گونه ای موثر مدل ها را ایجاد نمایند. جزئیات آن عبارتند از:
 - معماری، واسطه کاربر و جزئیات سطح تجهیزات.

کارهای مدل سازی تحلیلی

اصول مدل سازی تحلیلی

- قلمرو اطلاعات را ارائه می کند
- عملکرد نرم افزار را ارائه می کند
- رفتار نرم افزار را ارائه می کند
- تفکیک این ارائه ها

مولفه های مدل تحلیلی

- مدل داده
- مدل جریان
- مدل نوع
- مدل رفتار

کارهای مدل سازی طراحی

اصول

- در مدل تحلیلی، طراحی باید قابل ردیابی باشد.
- معماری همواره باید مد نظر باشد.
- تمرکز بر طراحی داده ها باشد.
- برنامه های واسطه باید طراحی شده باشند (هر دو کاربر و داخلی).
- اجزاء باید نشان دهنده استقلال عملکرد باشد.
- اجزاء باید به گونه ای ضعیف جفت شوند.

کارهای مدل سازی طراحی

طراحی ارائه شده باید به راحتی درک شود

- مدل طراحی شده باید به گونه ای قابل تکرار توسعه یابد

- مؤلفه های مدل طراحی

- طراحی داده

- طراحی معماری

- طراحی اجزا

- طراحی واسط

کارهای ساختاری

اصول آماده‌سازی: قبل از اینکه حتی یک خط کدنویسی انجام شود باید از موارد زیر مطمئن شد:

-درک مسئله‌ای که شما سعی در حل آن دارید (رجوع شود به ارتباط و مدل‌سازی)

-درک مفاهیم و اصول پایه‌ای

-گزینش زبان برنامه‌نویسی مناسب، که نیازهای نرم‌افزار و محیط آنرا که قرار است ایجاد شده و در آن کار کند برآورده سازد

انتخاب محیط برنامه‌نویسی که ابزار ارائه شده توسط آن موجب سادگی در کار گردد

ایجاد مجموعه‌ای از تست‌های واحد که اجزاء کامل شده کدگذاری را برای اولین اجراء تست خواهند کرد

کارهای مربوط به ساخت سیستم

اصول کدگذاری: با شروع به کدنویسی باید از موارد زیر اطمینان حاصل کرد:

- انتخاب ساختارهای داده که می‌خواهد نیازهای طراحی را برآورده سازد
- درک معماری نرم‌افزار و ایجاد واسطه‌هایی که با آن سازگار است
- حفظ شرایط منطقی تا جایی که ممکن است
- ایجاد حلقه‌های تو در تو به گونه‌ای که قابلیت تست آسانی داشته باشند
- انتخاب نام‌های متغیر معنی‌دار، و پیروی از دیگر استانداردهای کدگذاری محلی (بومی)
- نوشتن کدهایی که خودشان قابلیت مستند شدن داشته باشند
- ایجاد یک طرح دیداری (به عنوان مثال: ایجاد فضای خالی در شروع متن و ایجاد خطوط خالی که به درک مطلب کمک می‌کند)

کارهای مربوط به ساخت سیستم

اصول معتبرسازی:

- پس از تکمیل اولین مرحله کدگذاری، باید اطمینان حاصل نمائید:
- در زمان مقتضی جلسهای برای بررسی کد برگزار نمایند
- اجرای تستهای واحد و تصحیح خطاهایی که پوشش داده نشده‌اند
- دوباره عمل کردن به کد

کارهای مربوط به ساخت سیستم

اصول تست:

- همه تست‌ها باید قابل ردیابی نیازمندی‌ها باشند
- تست‌ها باید طرح ریزی شده باشند
- اصل پارتو باید در مورد تست کردن رعایت شوند
- ۸۰ درصد خطاها در ۲۰ درصد از کدها یافت می‌شوند
- تست‌ها از کم به زیاد باید انجام شود
- تست فراگیر ممکن نیست

امور مربوط به جاری سازی سیستم

اصول

- مدیریت انتظارات مشتری برای هر جزء اضافی
- یک بسته کامل باید ایجاد، تست و تحویل شود
- یک روش پشتیبانی از محصول باید ایجاد شود
- موضوعات آموزشی می بایست تهیه و در اختیار کاربر قرار گیرد
- ابتدا باید اشکالات نرم افزار مرتفع شده، سپس متحول شود

مبانی تجزیه تحلیل نیازها

تجزیه و تحلیل نیازها

عملیات نرم افزار و کارآیی لازم برای عملیات، واسطه های نرم افزار با مولفه های دیگر سیستم و محدودیت های طراحی که نرم افزار بایستی رعایت کند را مشخص می کند.

تجزیه و تحلیل نیازها

برای طراح نرم افزار نمایش اطلاعات و عملیات را مشخص می کند که منجر به طراحی داده ها، معماری و رویه ها می شود و برای مشتری و تولیدکننده نحوه ارزیابی کیفیت نرم افزار را بعد از ساخت مشخص می کند.

مراحل تهیه طرح جامع سیستم های اطلاعات مدیریت

خصوصیات تحلیلگر (Analyst):

۱- توانایی فهمیدن مفاهیم مجرد.

سازماندهی مجدد آنها در گروه‌های منطقی

ایجاد راه‌حلهایی برای هر گروه منطقی

۲- توانایی جذب حقایق صحیح و مناسب از منابع گیج کننده و متناقض.

۳- توانایی فهم محیط‌های کاربر/ مشتری.

۴- توانایی بکارگیری مولفه‌های سخت‌افزاری/نرم‌افزاری برای محیط‌های کاربر/ مشتری.

۵- توانایی برقراری ارتباط خوب به صورت شفاهی و کتبی.

اصول تحلیل :

دامنه اطلاعات بایستی شناخته شده و نمایش داده شود. (جریان اطلاعات، محتوای اطلاعات و ساختار اطلاعات)

۱. مدل‌هایی که نشانگر اطلاعات، عملیات و رفتار سیستم هستند.

۲. مسأله بایستی تقسیم‌بندی شود تا جزئیات مشخص گردد. (کاهش پیچیدگی)

۳. نمایش منطقی و فیزیکی سیستم بایستی ایجاد گردد.