فصل دوم

فرآیند توسعه نرم افزار

مقدمه

- مهندسی نرم افزار یک فناوری لایه ای است که مبتنی بر لایهٔ کیفیت است.
- مدیریت کیفیت به ارتقای مداوم فرایند توسعه کمک می کند، به طوری که همواره بهبود مستمر در فرایند مهندسی نرم افزار انجام شود.
- اساس مهندسی نرم افزار لایه های فرایند آن است.
- فرایند مهندسی نرم افزار همراه با لایه های فن آوری (روش ها و ابزارها) باعث توسعهٔ تدریجی نر م افزار های رایانه ای می شوند

مقدمه

- **فرایند نرم افزار چیست؟**
  - ***چارچوبی برای کارها و فعالیت های مورد نیاز برای توسعهٔ نرم افزاری با کیفیت بالا را فرایند نامند***
- **ضوابط ارزیابی نرم افزار**
- **ارزیابی نرم افزار از طریق عوامل داخلی و خارجی، با در نظر داشتن پارامترهای متفاوت صورت می گیرد.**

ضوابط ارزیابی نرم افزار

- **عوامل خارجی (کاربران)**
  - ***کسانی که از نرم افزار فقط استفاده می کنند و در فرایند توسعهٔ آن نقشی ندارند. معیارهای ارزیابی نرم افزار از نظر این افراد عبارت اند از:***
    - **صحت برنامه (Correctness)**
    - **استحکام از دید کاربر (Robustness)**
    - **قابلیت توسعه (Extendability)**
    - **قابلیت مصرف مجدد (Reusability)**
    - **سازگاری (Compatibility)**
    - **قابلیت حمل (Portability)**
    - **کارایی (Efficiency)**

ضوابط ارزیابی نرم افزار

- **عوامل داخلی (توسعه دهندگان نرم افزار)**
  - ***عوامل داخلی به توسعه دهندگان نرم افزار اطلاق می شود. مهندسین نرم افزار از دیدگاه فنی به ارزیابی نرم افزار می پردازند و به دنبال ارائهٔ پاسخ مشخص برای سؤالات زیر هستند:***
    - **از چه ابزارهایی(Tools) استفاده شده است؟**
    - **چه روش هایی(Methods) به کار برده شده است؟**
    - **چطور پردازش(Process) انجام شده است؟**
    - **مستندات کیفیت(Quality) چیست؟ (تمرکز بر کیفیت)**

لایه های مهندسی نرم افزار

مراحل فرایند مهندسی نرم افزار

- **فرایند مهندسی نرم افزار شامل چهار مرحلهٔ تعریف، توسعه، نگهداری و پشتیبانی نرم افزار است.**
- **تعریف**
  - ***فعالیت های مرحلهٔ تعریف***
    - **مهندسی سیستم یا اطلاعات** (System/Information Engineering)
    - **برنامه ریزی توسعه پروژهٔ نرم افزاری** (Software Development Project Planning)
    - **تحلیل و مدل سازی نیازها** (Requirments Analysis and Modeling)

مراحل فرایند مهندسی نرم افزار

- **نگهداری**
  - ***نگهداری اصلاحی(Correction):***
    - **اشکالات در چرخهٔ زندگی پروژه، شناسایی، بررسی و رفع می شود (رفع خطا).**
  - ***نگهداری تطبیقی(Adaptation):***
    - **تغییر به مرور زمان با توجه به نیازهای محیطی و کاربر صورت می گیرد (تطبیق با نیازهای جاری).**
  - ***نگهداری ارتقایی(Enhancement):***
    - **ارتقای نرم افزار پس از راه اندازی که توسط کاربر درخواست می شود (افزایش امکانات).**
  - ***نگهداری پیشگیرانه(Prevention):***
    - **ایجاد تغییراتی در برنام هها که منجر به ارتقا و بهبود عملکرد نرم افزار می شود (آینده نگری).**

مراحل فرایند مهندسی نرم افزار

- **فعالیت های پشتیبانی فرایند مهندسی نرم افزار**
  - ***فعالیت های چتری یا پشتیبانی (Umbrella Activities) در تمامی مراحل زیر انجام می گیرد***
    - **کنترل و نظارت بر برنامه ریزی پروژه**(Project Monitoring and Control)
    - **بازنگری های مستمر فنی رسمی** (Formal Technical Review)
    - **اطمینان مرغوبیت نرم افزار** (Quality Assurance)
    - **مدیریت پیکربندی نرم افزار** (Configuration/Change Management)
    - **تهیه و تدوین مستندات فنی** (Technical Documentation)
    - **اندازه گیری نرم افزار با مقیا سهای اندازه گیری** (Software Measurements)
    - **مدیریت ریسک نرم افزار** (Risk Management)

مدل بلوغ توانایی(Capability Maturity Model)

- مؤسسهٔ مهندسی نرم افزار(SEI) آمریکا مدل جامعی ارائه داده است که مجموع های از توانایی های مهندسی نرم افزار برای دسترسی سازما نها به سطوح مختلف بلوغ فرایند در آن پیش بینی شده است. این مدل تعیین کنندهٔ میزان مؤثر بودن فعالیت های مهندسی نرم افزار در یک سازمان نرم افزاری بوده و دارای پنج سطح تکامل فرایندی است.

مدل بلوغ توانایی(Capability Maturity Model)

- **سطح ۱: اولیه(Initial)**

فرایند نرم افزار به صورت موقتی و حتی بعضی اوقات بسیار درهم و برهم توصیف شده است. چند فرایند ساده تعریف می شود و موفقیت به تلاش های فردی بستگی دارد.

از نظر تضمین کیفی و مدیریت پروژه، وظیفه خاصی وجود ندارد. در این سطح تیم پروژه می تواند برای توسعه و ارائه نرم افزار هر راهی انتخاب کند. روش ها، استانداردها و رویه هایی که از کیفیت خوب تا خیلی ضعیف دارند، می توانند در این انتخاب قرار بگیرند.

مدل بلوغ توانایی(Capability Maturity Model)

- **سطح ۲: قابل تکرار(Repeatable)**

فرایندهای اولیه مدیریت پروژه برای مشخص کردن هزینه ، زمان بندی و کارایی، در این سطح صورت می گیرد. این سطح نمایانگر این حقیقت است که ارائه دهنده نرم افزار، فعالیت های خاصی را همچون گزارش تکمیل کار و گزارش زمان بندی و فعالیت های انجام شده را تعریف کرده است.

مدل بلوغ توانایی(Capability Maturity Model)

- **سطح ۳: تعریف شده(Defined)**

فرایند نرم افزار برای مدیریت فعالیت های مهندسی در طول فرایند توسعهٔ نرم افزار در سازمان ثبت شده، استاندارد و منسجم است. این سطح شامل مشخصه های تعریف شده در سطح ۲ نیز است. این سطح نمایانگر این است که ارائه دهندهٔ نرم افزار، فرایندهای فنی و مدیریتی را تعریف کرده است. این سطح از سطوحی است که اکثر تولیدکنندگان نرم افزار از طریق استانداردهایی مثل ایزو ۹۰۰۱ آن را مد نظر دارند.

مدل بلوغ توانایی(Capability Maturity Model)

- **سطح ۴: مدیریت شده(Managed)**

اقدامات دقیق صورت گرفته در فرایند نرم افزار و کیفیت محصول، همگی کنترل می شوند. فرایند نرم افزار و محصولات از لحاظ کمی شناسایی شده و با استفاده از اقداماتی دقیق کنترل می شوند. این سطح دربرگیرندهٔ مفهوم اندازه گیری و استفاده از معیارهای سنجش برای فرایند توسعهٔ نرم افزار و محصول نرم افزاری است. این سطح شامل تمام مشخصه های تعریف شده در سطح ۳ نیز است.

مدل بلوغ توانایی(Capability Maturity Model)

- **سطح ۵: بهینه سازی(Optimizing)**

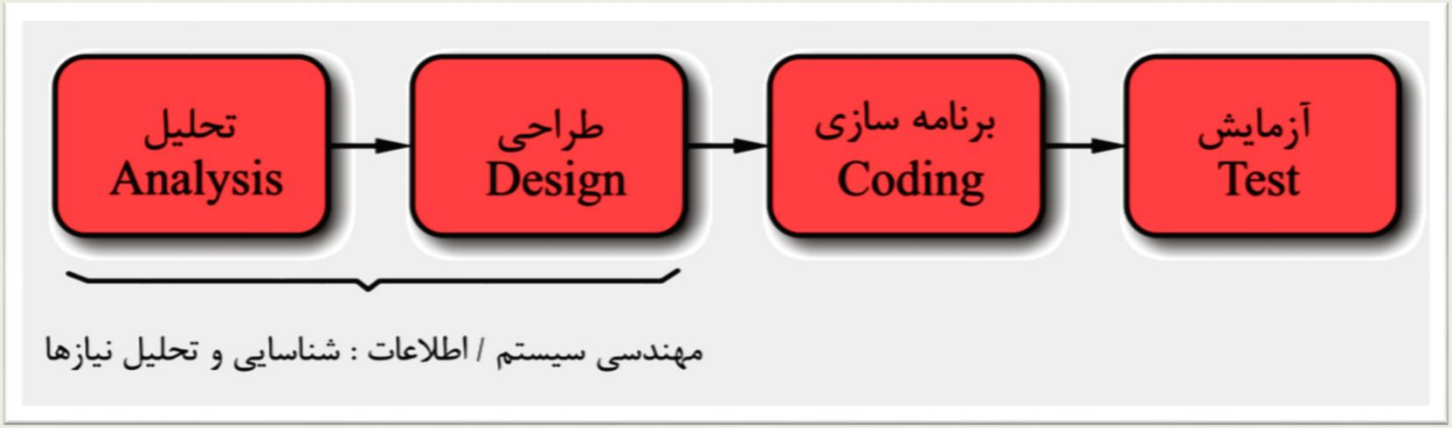
بهبود مکرر فرایند با توجه به بازخورد کمّی فرایند و از طریق آزمون ایده های نوآور و فن آوری های نوین و بروز ، مقدور است. این سطح شامل تمام مشخصه های تعریف شده برای سطح ۴ است.

مدل های پایه در فرایند توسعهٔ نرم افزار

مدل ترتیبی خطی(The Linear Sequential Model)

این مدل گاهی مدل پایه نیز نامیده می شود. مدل خطی ترتیبی یک سری فعالیت متوالی برای توسعهٔ نرم افزار پیشنهاد می کند که از سطح تحلیل آغاز و به آزمایش نرم افزار ختم می شود.

این مدل برای پروژه های کوچک کارا است زیرا در آن تعریف صورت مسئله به راحتی انجام می گیرد.

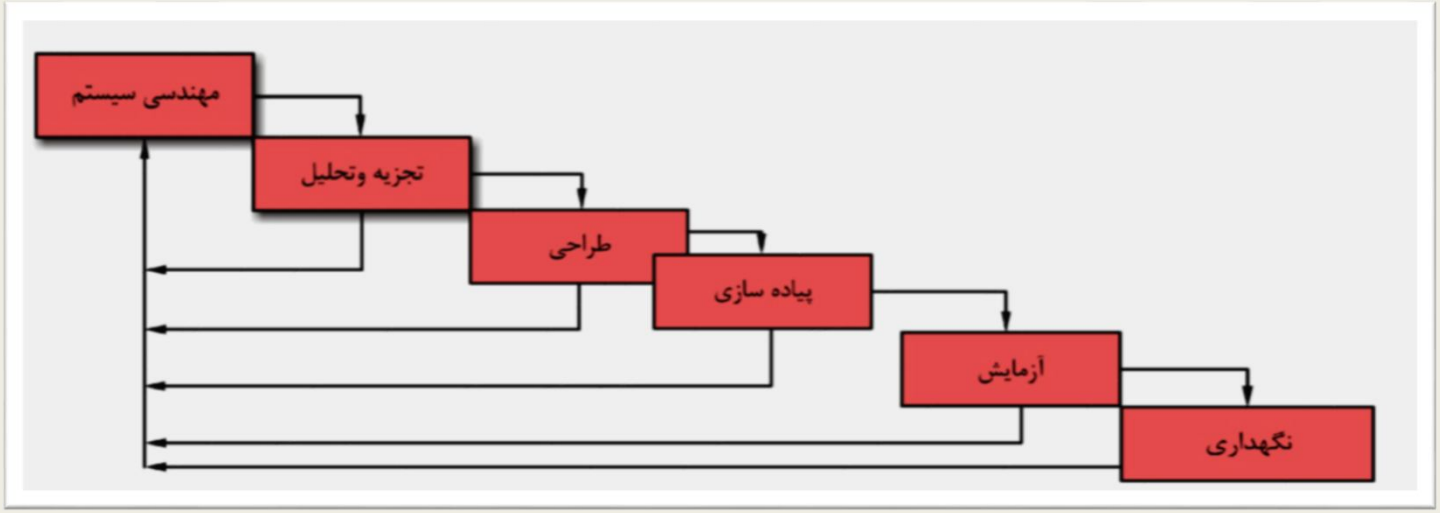


مدل های پایه در فرایند توسعهٔ نرم افزار

مدل آبشاری(WaterFall Model)

مدل آبشاری نیز اغلب با نام مدل دورهٔ زندگی کلاسیک معرفی می شود. در این مدل فعالیت توسعه نرم افزار از مهندسی سیستم شروع و به نگهداری نرم افزار ختم می شود.

مدل آبشاری ساده است. در پروژه های بزرگ، مدت زمان انجام پروژه طولانی است و بنابراین نیازها تغییر می یابد؛ ازاین رو بدیهی است از این مدل در پروژه های بزرگ استفاده نشود.



مدل های پایه در فرایند توسعهٔ نرم افزار

مدل آبشاری(WaterFall Model)

مدل آبشاری ساده است. در پروژه های بزرگ، مدت زمان انجام پروژه طولانی است و بنابراین نیازها تغییر می یابد؛ ازاین رو بدیهی است از این مدل در پروژه های بزرگ استفاده نشود.

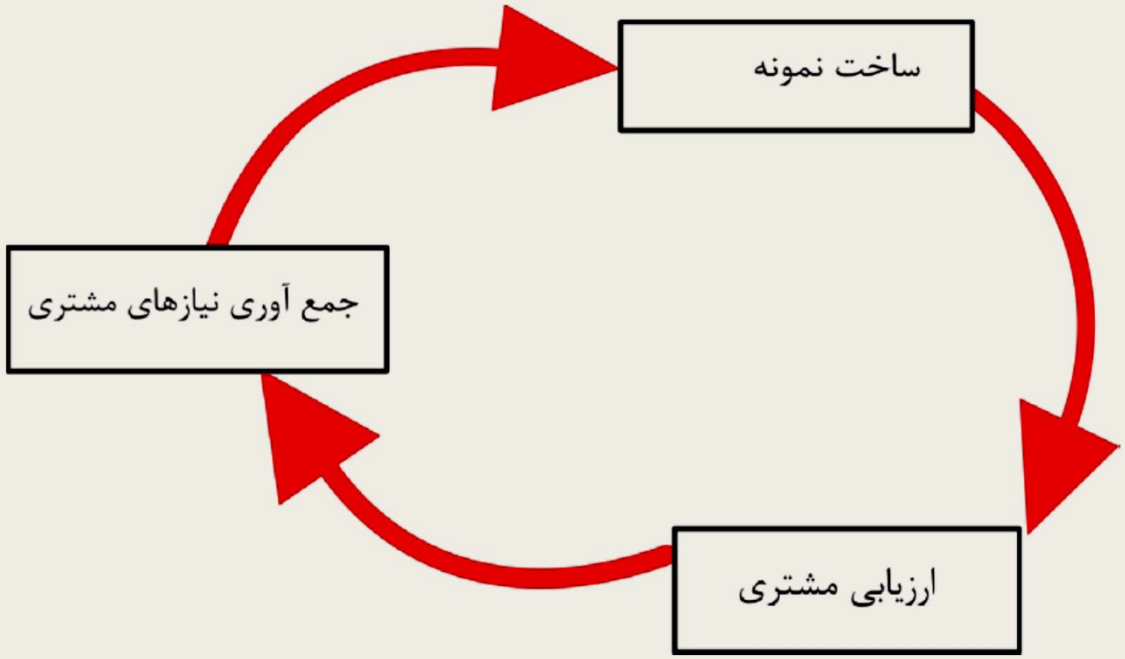
مشکلات مدل

- ***مشکل برگشت به عقب در نتیجهٔ بروز تغییرات***
- ***عدم بیان نیازها توسط مشتری در ابتدای پروژه***
- ***طولانی منتظر ماندن مشتری برای دستیابی به نرم افزار***

مدل های پایه در فرایند توسعهٔ نرم افزار

مدل نمونه سازی(Prototyping)

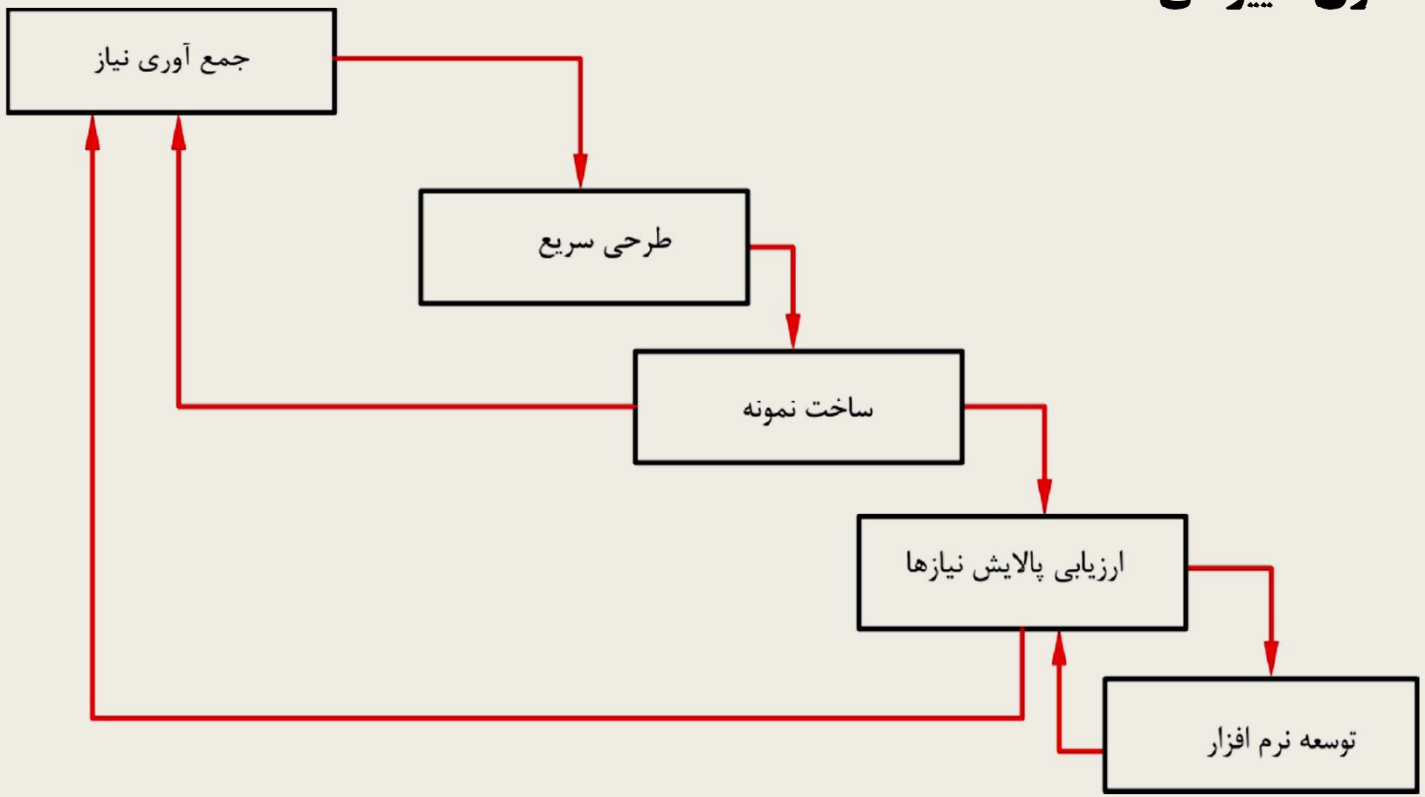
بخش های مختلف سیستم به صورت مرحله ای شناسایی و نمونه سازی می شود. مدل نمونه سازی، نوعی تولید مجدد محسوب می شود (دوباره کاری). در حالت ایده آل، نمونه سازی روشی برای مشخص کردن نیازهای نرم افزار محسوب می شود.



مدل های پایه در فرایند توسعهٔ نرم افزار

مدل نمونه سازی(Prototyping)

در ابتدا برنامه خیلی ساده در اختیار مشتری قرار می گیرد و سپس آن را طبق نیازهای مشتری تغییر می دهند.



مدل های پایه در فرایند توسعهٔ نرم افزار

- **مدل نمونه سازی(Prototyping)**

مزایا

- **ارتباط با مشتری در تمام طول اجرای پروژه برقرار است.**
- **تغییر نیازها به راحتی امکان پذیر است.**

معایب

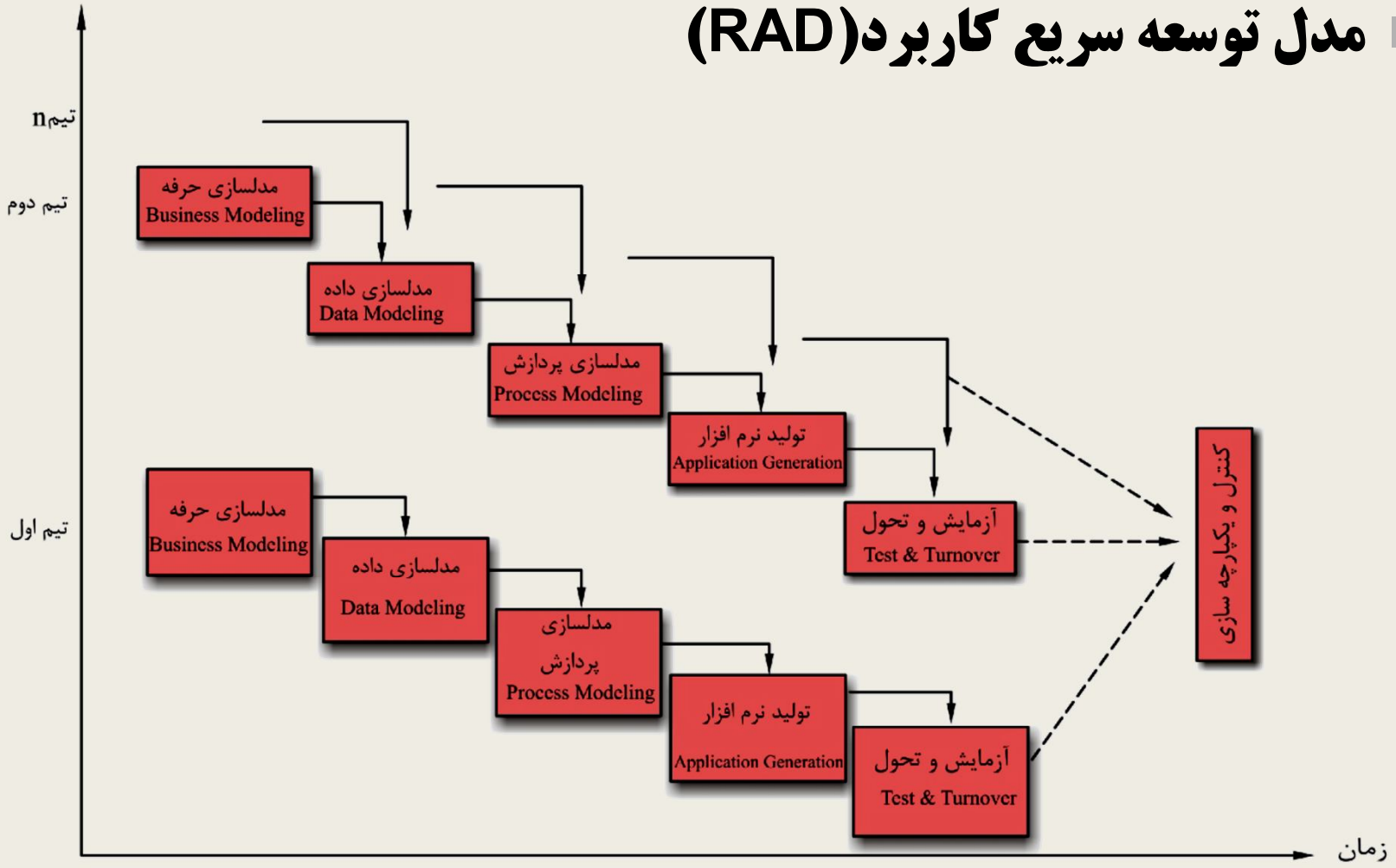
- **مشتری، کل نرم افزار را نمی بیند و ملاحظات کلی از نرم افزار را نمی تواند داشته باشد (مثل قابلیت نگهداری).**
- **تغییر زیاد ممکن است باب میل مشتری نباشد (مانند نمونه های ساخته شدهٔ دورانداختنی).**
- **سرعت انجام کار باعث بروز اثرات جانبی می شود (مثل تغییر و محدودیت های سخت افزاری یا نرم افزاری).**

مدل های پایه در فرایند توسعهٔ نرم افزار

- مدل توسعه سریع کاربرد (Rapid Application Development)
- تاکید بر توسعه سریع نرم افزار (RAD) در مدت زمان کوتاه، حداکثر در سه ماه است.
- در هر قسمت، روش خطی و یا هر مدل دیگر قابل استفاده است.
- در این روش سیستم در قالب کارکردهای (پیمانه های) مختلف شکسته می شود و تیم های مختلف، همزمان روی هر یک از آنها فعالیت می کنند و درنهایت نتیجه کار با هم ترکیب و یکپارچه می شود.

مدل های پایه در فرایند توسعهٔ نرم افزار

- مدل توسعه سریع کاربرد(RAD)



مدل های پایه در فرایند توسعهٔ نرم افزار

- **مدل توسعه سریع کاربرد(RAD)**

مزایا

- **زمان توسعه و تولید نرم افزار کوتاه است.**
- **تأکید بر قابلیت استفادهٔ مجدد مؤلفه های نرم افزاری است.**

معایب

- **مدیریت تیم ها و هماهنگی آنها خیلی مشکل است.**
- **در سیستم هایی می توان این کار را کرد که قابلیت پیمانه ای(Modularity) وجود داشته باشد که اغلب مواقع این طور نیست.**
- **در پروژه های بزرگ نیاز به افراد متخصص زیادی است.**
- **عدم اطمینان از تعهد کاری و پایبندی افراد در تیم های پروژه به عنوان ریسک، همواره وجود دارد**

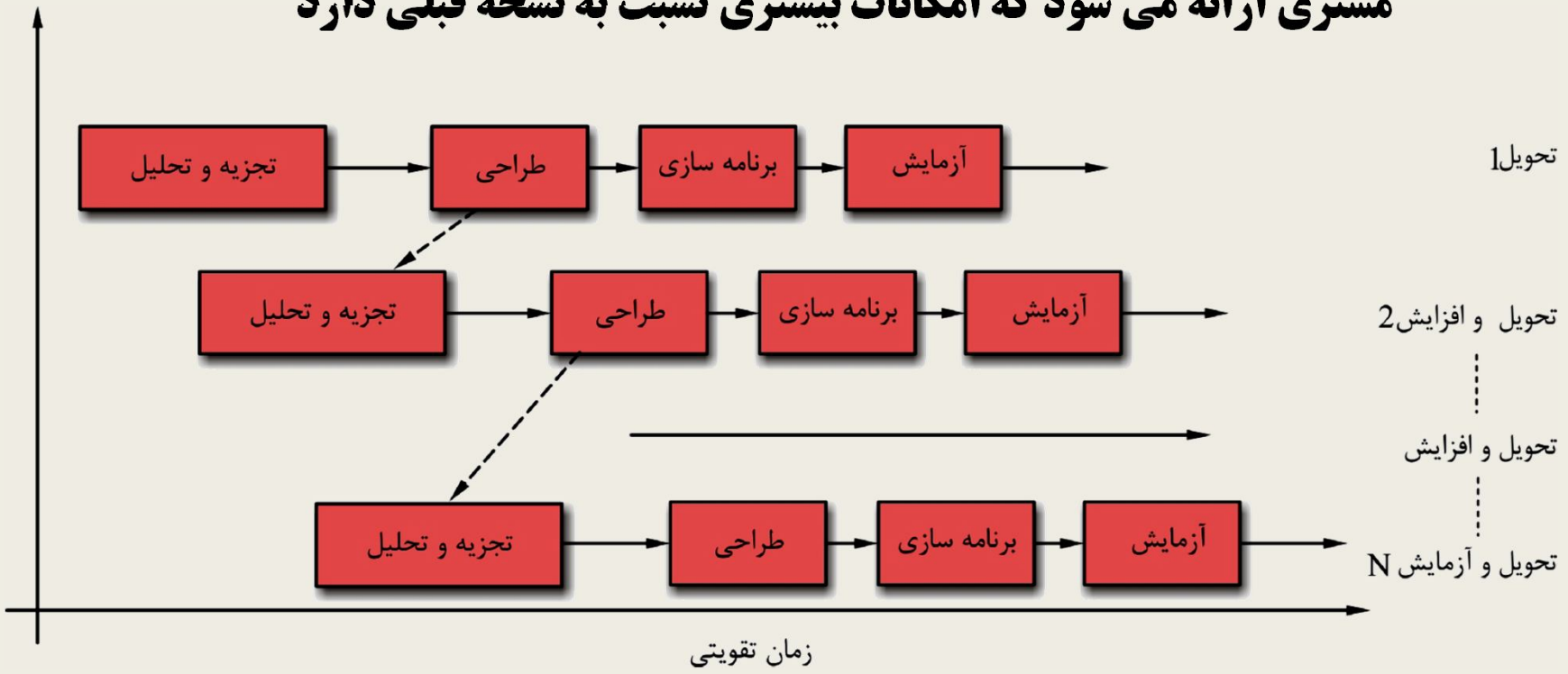
مدل های تکاملی در فرایند توسعهٔ نرم افزار

- ایده گسترش و بلوغ نرم افزار مانند سایر سیستم های پیچیده همواره مطرح است.
- با رشد ابزارها و فن آوری مربوطه، فرایند های توسعهٔ نرم افزار نیز تکامل می یابند.
- مدل های تکاملی ماهیت تکراری دارند، یعنی مهندسین نرم افزار را قادر می کند تا به صورت افزایشی نسخه های کامل تری از نرم افزار را توسعه و عرضه کنند.

مدل های تکاملی در فرایند توسعهٔ نرم افزار

مدل افزایشی(Incremental Model)

ابتدا محصول پایه ای تولید می شود و سپس در هر مرحله، نسخه ای از نرم افزار به مشتری ارائه می شود که امکانات بیشتری نسبت به نسخه قبلی دارد



مدل های تکاملی در فرایند توسعهٔ نرم افزار

- **مدل افزایشی(Incremental Model)**

مزایا

- **برعکس روش نمونه سازی، نسخه های نرم افزار دور ریخته نمی شود.**
- **افزودن امکانات جدید به نرم افزار، ضمن به هنگام سازی آن، هزینهٔ کمتری دربردارد.**

معایب

- **نیاز به نیروی انسانی متخصص و یا سخت افزار کافی دارد.**

مدل های تکاملی در فرایند توسعهٔ نرم افزار

- مدل حلزونی(Spiral Model)

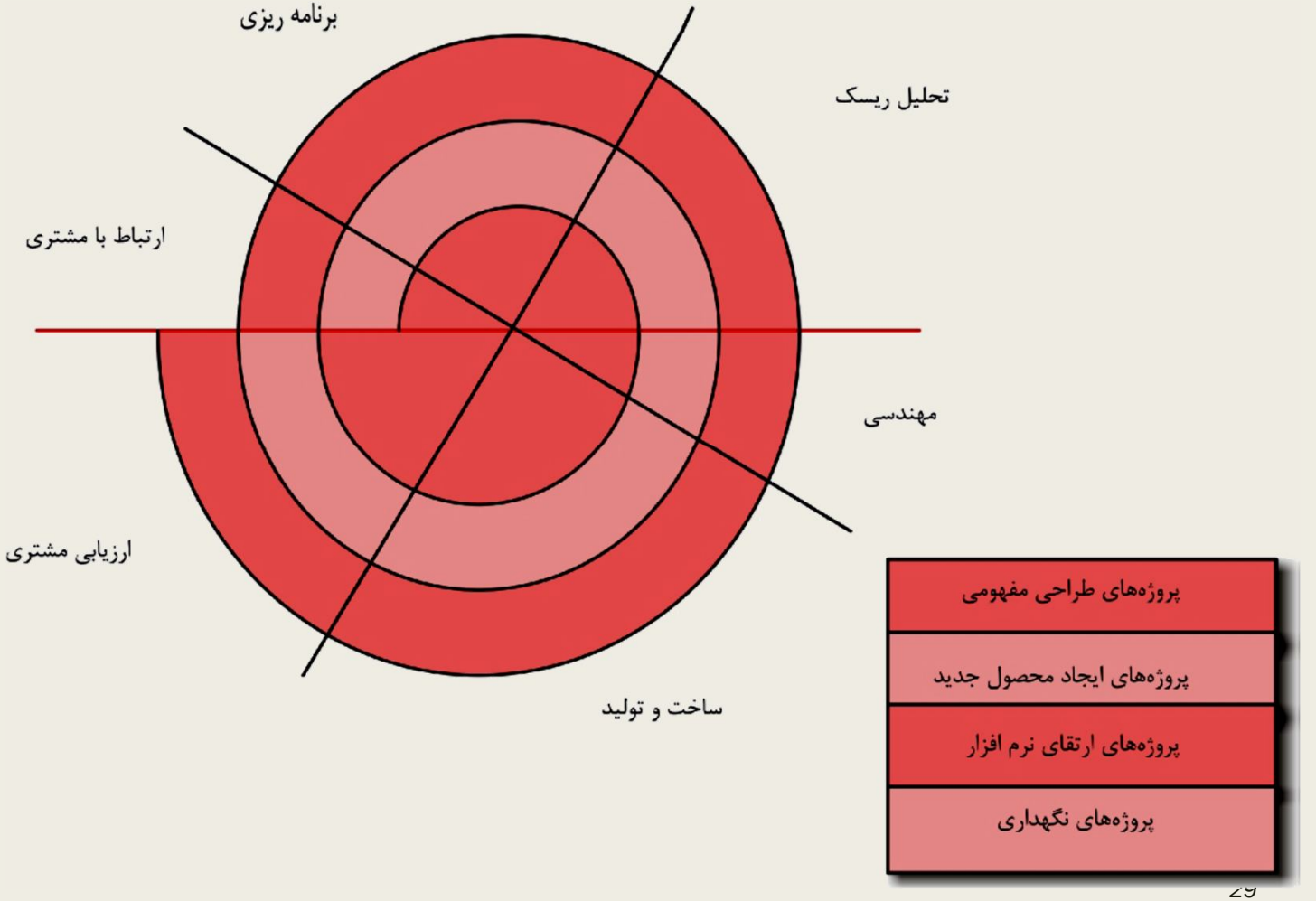
ترکیب روش خطی با فلسفهٔ نمونه سازی جهت استفاده در پروژه های بزرگ

فعالیت های مربوط به مهندسی نرم افزار به شش زمینه کاری تقسیم می شود:

ارتباط با مشتری، برنامه ریزی، تحلیل ریسک، مهندسی و طراحی، ساخت و تحویل، ارزیابی مشتری

مدل های تکاملی در فرایند توسعهٔ نرم افزار

مدل حلزونی(Spiral Model)



مدل های تکاملی در فرایند توسعهٔ نرم افزار

- **مدل حلزونی(Spiral Model)**

مزایا

- **در هر دور از مدل، تحلیل ریسک برای بهبود کیفیت نرم افزار صورت می گیرد. ·**
- **تعداد دورهای این روش با توافق مشتری و هزینه ای که پرداخت می کند تعیین می شود. ·**
- **مدیریت پروژه همواره بر پروژه اعمال می شود. ·**
- **کنترل ریسک همواره وجود دارد.**

مدل های تکاملی در فرایند توسعهٔ نرم افزار

- مدل مارپیچ WinWin

استخراج نیازهای نرم افزار وابسته به مذاکرات است. مذاکرات موفق یعنی توفیق و موفقیت طرفین.

این مدل یک سری مذاکرات را در آغاز هر مرحله از مارپیچ بیان می کند و تمامی فعالیت های زیر تعریف می شوند:

شناسایی سیستم یا دارندگان اصلی سیستم های فرعی، تعیین شرایط برد طرفین، مذاکره در مورد شرایط برد طرفین.

مدل های تکاملی در فرایند توسعهٔ نرم افزار

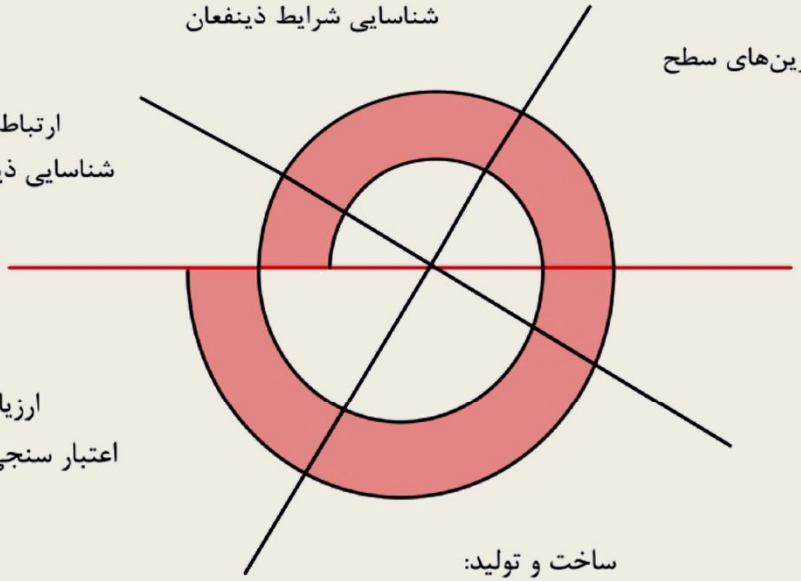
مدل مارپیچ WinWin

تحلیل ریسک :

- تعیین شرایط برد
- برقراری اهداف اصلی، محدودیت‌ها و جایگزین‌های سطح

شناسایی شرایط ذینفعان

ارتباط با مشتری:
شناسایی ذینفعان سطح بعد



ارزیابی مشتری:
اعتبار سنجی محصول و فرآیند

مهندسی:

ارزیابی فرآیند، محصول و حل مشکلات

ساخت و تولید:
طراحی و توسعه محصول مهندسی

مدل های تکاملی در فرایند توسعهٔ نرم افزار

- **مدل مونتاژ مؤلفه ها(Component Assembly Model)**

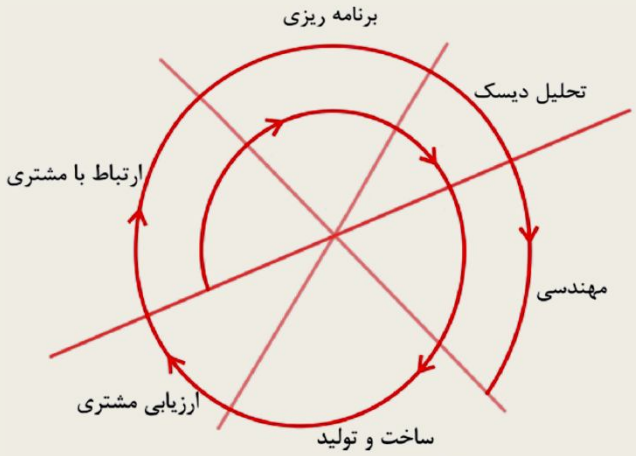
اجزای سیستم به صورت مؤلفه هایی در نظر گرفته می شوند که از قابلیت ساخت، آزمون و استفادهٔ مناسبی برخوردار هستند.

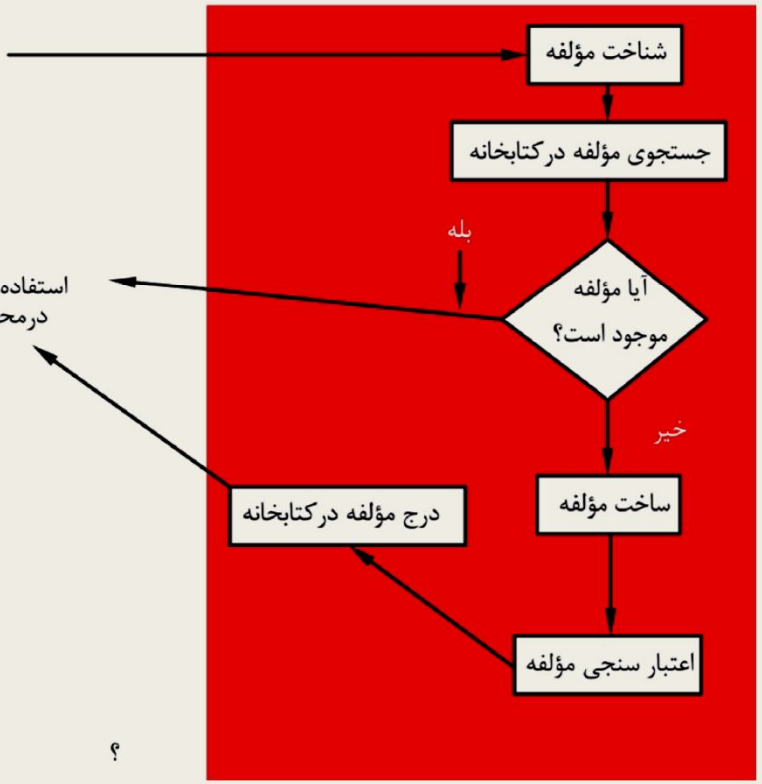
هر مدل یا روشی در توسعهٔ نرم افزار در اینجا استفاده شود، فرقی نمی کند.

در این مدل هدف و تأکید بر قابلیت استفاده مجدد از مؤلفه هاست.

مدل های تکاملی در فرایند توسعهٔ نرم افزار

مدل مونتاژ مؤلفه ها(CAM)





مدل های تکاملی در فرایند توسعهٔ نرم افزار

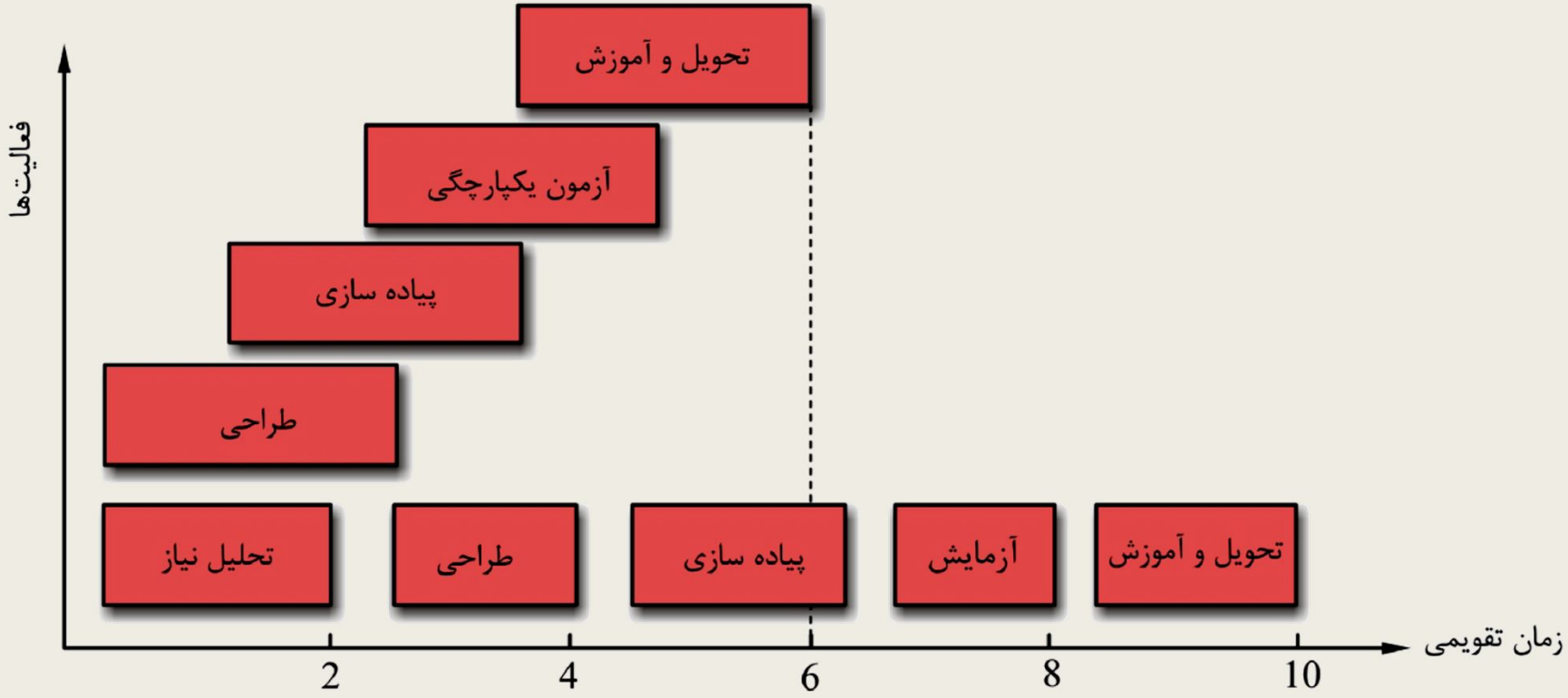
- **مدل توسعهٔ هم روند(Concurrent Development Model)**

در مدل توسعه هم روند فعالیت ها به منظور کاهش زمان ایجاد نرم افزار به طور موازی انجام می گیرند.

در حالت موازی، زمان کاهش می یابد ولی ممکن است بعضی پارامترهای دیگر مثل دوباره کاری و ... افزایش یابد. در این حالت مدیریت پروژهٔ نرم افزاری پیچیدگی زیادی دارد.

مدل های تکاملی در فرایند توسعهٔ نرم افزار

مدل توسعهٔ هم روند(CDM)



مدل های تکاملی در فرایند توسعهٔ نرم افزار

مدل روش های رسمی(Formal Methods Model)

مجموعه ای از فعالیت ها جهت طراحی و بیان نرم افزار رایانه ای در قالب روابط ریاضی

مزایا

- بدیع، سازگار، بدون ابهام، کاربرد در سیستم های حساس و ...

معایب

- وقت گیر و پرهزینه، کاربرد محدود، نیاز به آموزش جامع، مشکل ارتباط با مشتری و ...

مدل های تکاملی در فرایند توسعهٔ نرم افزار

- **ابزارهای نسل چهارم(4GT-Forth Generation Tool)**

شامل محدودهٔ وسیعی از ابزارهای نرم افزاری است که قادر به تعیین و مشخص کردن بعضی از خصوصیات و ویژگی های نرم افزار در سطح بالاست و می تواند حتی کُدی بر مبنای مشخصات تولیدشده به صورت خودکار ایجاد کند.

در این ابزارها قابلیت های زیر وجود دارد:

قابلیت استفاده برای مستندسازی و مدل سازی سیستم، تولید گزارشات، استفادهٔ آسان از امکانات محیط برنامه سازی، مدیریت آسان پایگاه داده ها ، قابلیت تولید کد برنامه ·

عیب

تکنیک ها و ابزارهای موجود در حال حاضر خیلی عام نیستند و کاربردهای خاص دارند.

مدل های تکاملی در فرایند توسعهٔ نرم افزار

- **فرایند مدل سازی یکنواخت(Rational Unified Process)**
- **ایجاد یک بستر یکسان برای انجام فرایندهای مهندسی نرم افزار شامل ۴ فاز اصلی و ۹ نظام کاری**
- **تدوین خروجی های هر فاز در چارچوب مستندات مشخص و جریان مشخص شدن کار نظام های کاری نیز**
- **در این فرایند ایده تکراری(Iterative) بودن فرایند مهندسی نرم افزار نیز مد نظر بوده که در هر فاز تکرارهای متفاوتی را می توان اجرا کرد.**

مدل های تکاملی در فرایند توسعهٔ نرم افزار

فرایند مدل سازی یکنواخت(RUP)