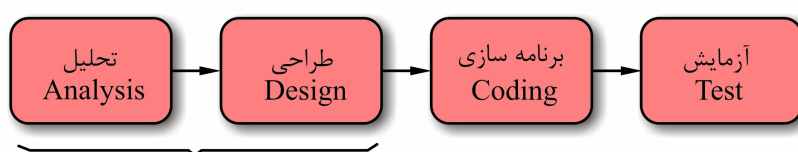


## ۵-۲ مدل‌های پایه در فرایند توسعه نرم‌افزار

### ۱-۵-۲ مدل ترتیبی خطی

این مدل گاهی مدل پایه نیز نامیده می‌شود. مدل خطی ترتیبی<sup>۵</sup> یک سری فعالیت متوالی برای توسعه نرم‌افزار پیشنهاد می‌کند که از سطح تحلیل آغاز و به آزمایش نرم‌افزار ختم می‌شود.



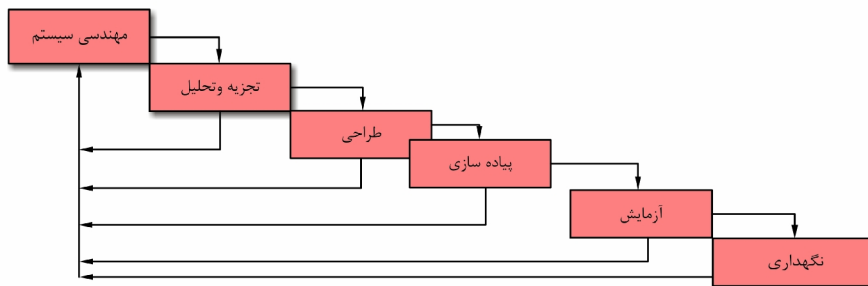
مهندسی سیستم / اطلاعات : شناسایی و تحلیل نیازها

شکل ۲-۲ مدل ترتیبی خطی

این مدل برای پروژه‌های کوچک کارا است زیرا در آن تعریف صورت مسئله به راحتی انجام می‌گیرد.

### ۲-۵-۲ مدل آبشاری

مدل آبشاری<sup>۶</sup> نیز اغلب با نام مدل دوره زندگی کلاسیک معرفی می‌شود. در این مدل فعالیت توسعه نرم‌افزار از مهندسی سیستم شروع و به نگهداری نرم‌افزار ختم می‌شود.



شکل ۳-۲ مدل آبشاری

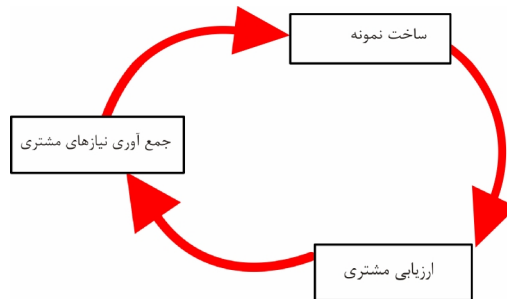
مدل آبشاری ساده است. در پروژه‌های بزرگ، مدت زمان انجام پروژه طولانی است و بنابراین نیازها تغییر می‌یابد؛ از این رو بدیهی است از این مدل در پروژه‌های بزرگ استفاده نشود.

**مشکلات مدل**

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

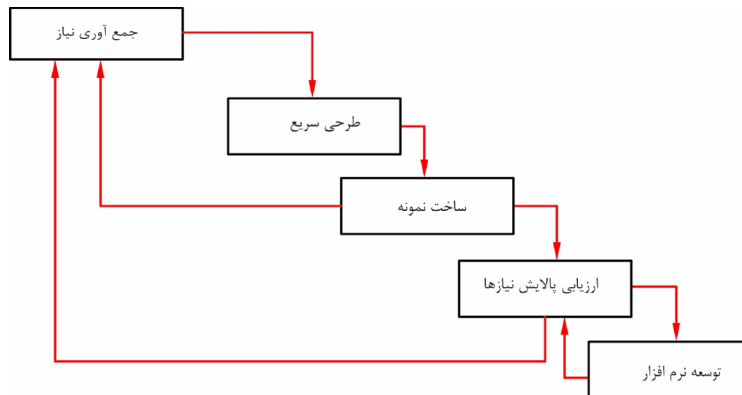
**۳-۵-۲ مدل نمونه سازی**

بخش‌های مختلف سیستم به صورت مرحله‌ای شناسایی و نمونه سازی می‌شود. مدل نمونه سازی<sup>۱</sup>، نوعی تولید مجدد محسوب می‌شود (دوباره کاری). در حالت ایده آل، نمونه سازی روشی برای مشخص کردن نیازهای نرم افزار محسوب می‌شود.



شکل ۴-۲ روش نمونه سازی

در ابتدا برنامه خیلی ساده در اختیار مشتری قرار می‌گیرد و سپس آن را طبق نیازهای مشتری تغییر می‌دهند.



شکل ۵-۲ نمونه سازی در مدل آبشاری

مزیت

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

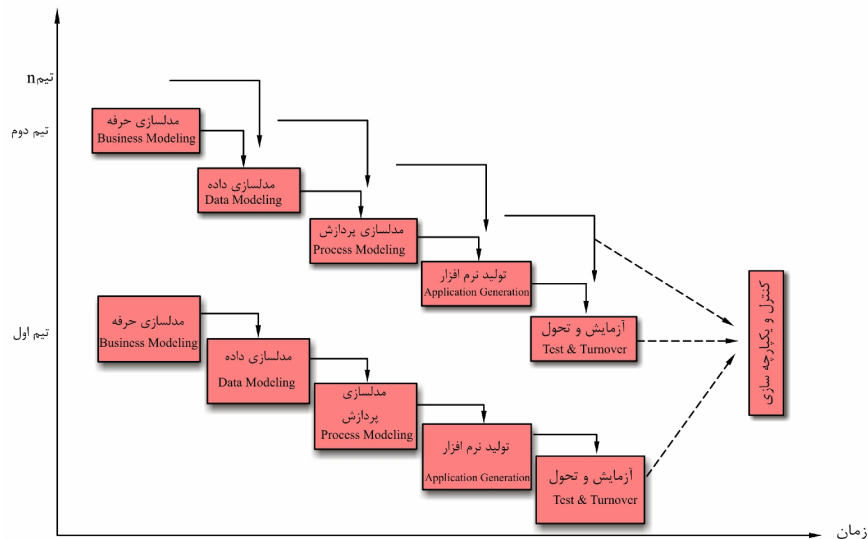
عیب

- 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 شده دوراندختنی).
- سرعت انجام کار باعث بروز اثرات جانبی می شود (مثل تغییر و محدودیت های سخت افزاری یا نرم افزاری).

۴-۵-۲ مدل توسعه سریع کاربرد (RAD)

در این مدل تأکید بر توسعه سریع نرم افزار<sup>۱</sup> 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 در نهایت نتیجه کار با هم ترکیب و یکپارچه می شود.



شکل ۶-۲ مدل توسعه سریع کاربرد

مزیت

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

عیب

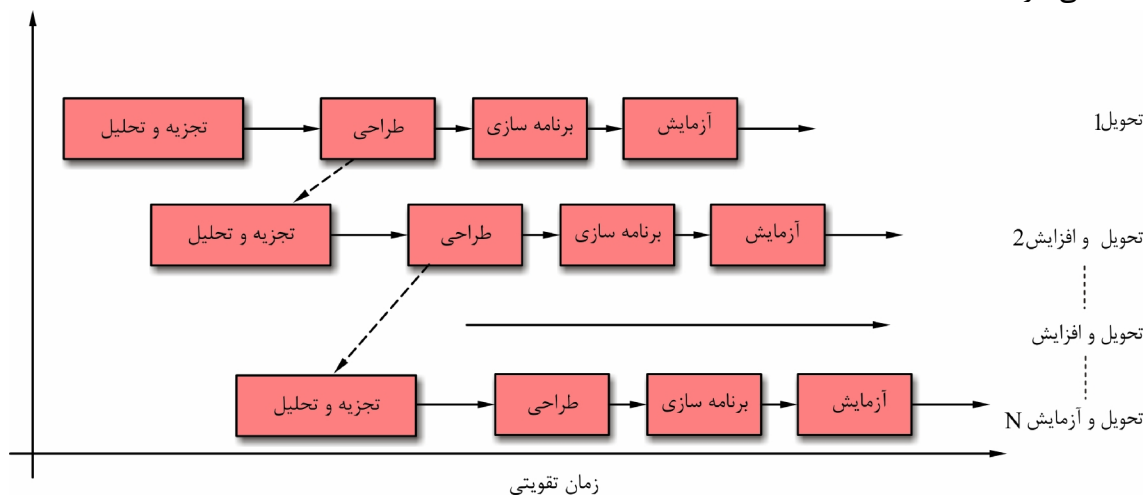
- مدیریت تیم ها و هماهنگی آنها خیلی مشکل است.
- در سیستم هایی می توان این کار را کرد که قابلیت پیمانه ای<sup>۲</sup> وجود داشته باشد که اغلب مواقع این طور نیست.
- در پروژه های بزرگ نیاز به افراد متخصص زیادی است.
- عدم اطمینان از تعهد کاری و پایبندی افراد در تیم های پروژه به عنوان ریسک، همواره وجود دارد.

## ۶-۲ مدل های تکاملی فرایند توسعه نرم افزار

ایده گسترش و بلوغ نرم افزار مانند سایر سیستم های پیچیده همواره مطرح است. با رشد ابزارها و فن آوری مربوطه، فرایندهای توسعه نرم افزار نیز تکامل می یابند. در این بخش مدل های تکاملی فرایند توسعه نرم افزار مورد بررسی قرار می گیرند. مدل های تکاملی ماهیتاً تکراری هستند، یعنی مهندسين نرم افزار را قادر می کند تا به صورت افزایشی نسخه های کامل تری از نرم افزار را توسعه و عرضه کنند.

### ۱-۶-۲ مدل افزایشی

در مدل افزایشی<sup>۱</sup> 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.



شکل ۷-۲ مدل افزایشی

#### مزیت

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

#### عیب

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

### ۲-۶-۲ مدل حلزونی

مدل حلزونی<sup>۲</sup> (معروف به مدل Bohem) روش خطی را با فلسفه نمونه سازی ترکیب می کند که در پروژه های بزرگ مورد استفاده است. در این مدل فعالیت های مربوط به مهندسی نرم افزار به شش زمینه کاری تقسیم می شود:

ارتباط با مشتری<sup>۳</sup>: قراردادهای و کل توافقات انجام گرفته و اطلاعات کلان و کلی از پروژه جمع آوری می شود.

برنامه ریزی<sup>۴</sup>: بر اساس نتایج به دست آمده از شناخت کلان، اعضای تیم شناسایی و تیم های کاری تشکیل و برای هر تیم، تخصیص کار صورت می گیرد و برنامه کنترل و نظارت پروژه تدوین می شود.

تحلیل ریسک<sup>۵</sup>: پیش بینی مخاطرات غیر منتظره مثل تغییرات فن آوری سخت افزار و نرم افزار، عدم تعهد کارفرما یا نیروی انسانی، تغییرات مدیریتی، تغییر خواسته مشتری (عوامل محیطی) و ... انجام می شود.

مهندسی و طراحی<sup>۱</sup>: فعالیت های مهندسی مورد نیاز برای طراحی و ساخت نمونه هایی از برنامه کاربردی انجام می گیرد.

1 - Incremental Model

2 - Spiral Model

3 - Customer Communication

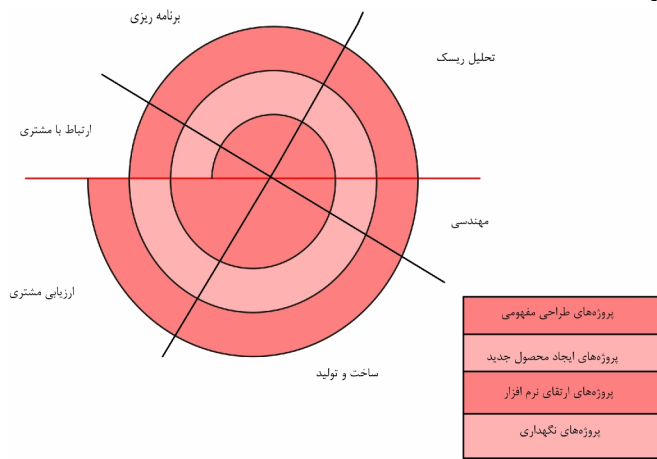
4 - Planning

5 - Risk Analysis

ساخت و تحویل<sup>۲</sup>: برنامه نویسی، آزمون، کنترل کیفیت، نصب و آموزش از عمده فعالیت های این بخش است.  
 ارزیابی مشتری<sup>۳</sup>: اخذ نظرات مشتری و کاربران در مورد محصولات تحویلی به آنان و ارزیابی و اعمال تغییرات در صورت نیاز در این مرحله صورت می گیرد.

### 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.

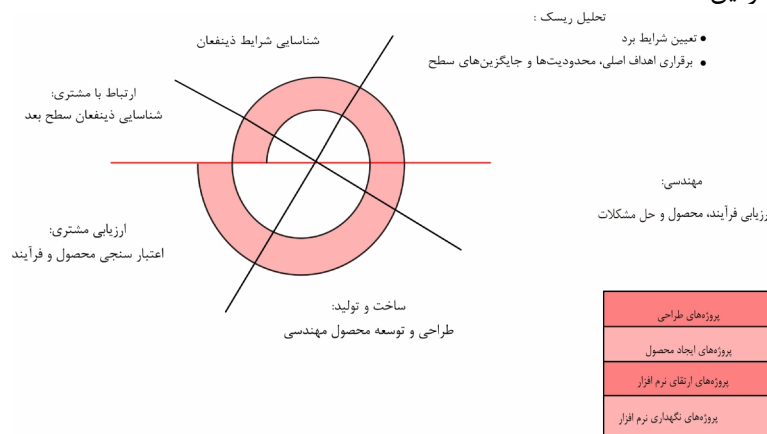


شکل ۲-۸ مدل حلزونی (مارپیچی)

### ۲-۶-۳ مدل حلزونی WinWin

استخراج نیازهای نرم افزار وابسته به مذاکرات است. مذاکرات موفق یعنی توفیق و موفقیت طرفین. مدل حلزونی بوهم یک سری مذاکرات را در آغاز هر مرحله از مارپیچ بیان می کند و تمامی فعالیت های زیر تعریف می شوند:

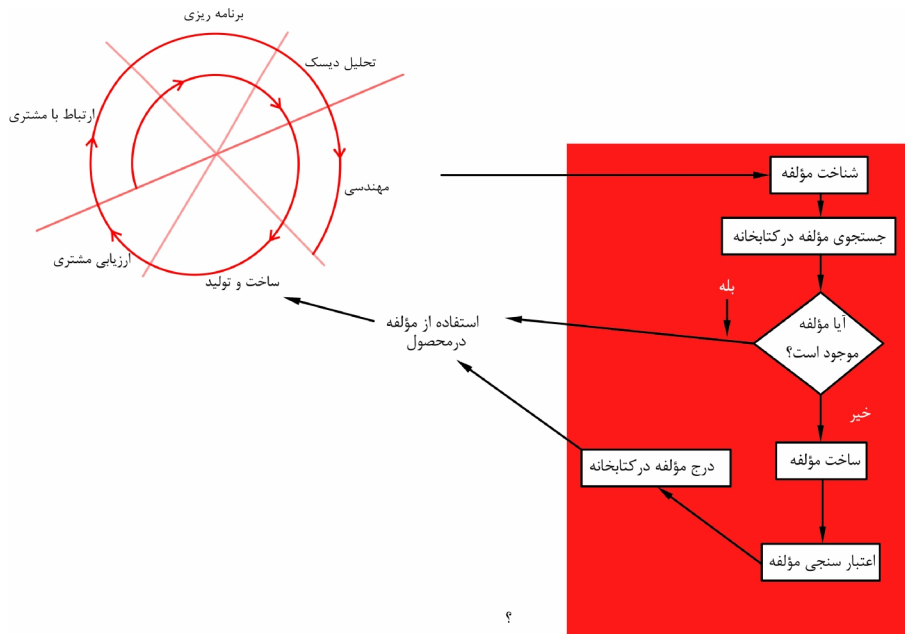
- ۱- شناسایی سیستم یا دارندگان اصلی سیستم های فرعی.
- ۲- تعیین شرایط برد طرفین.
- ۳- مذاکره در مورد شرایط برد طرفین.



شکل ۲-۹ مدل حلزونی WinWin

### ۴-۶-۲ مدل مونتاز مؤلفه‌ها<sup>۱</sup>

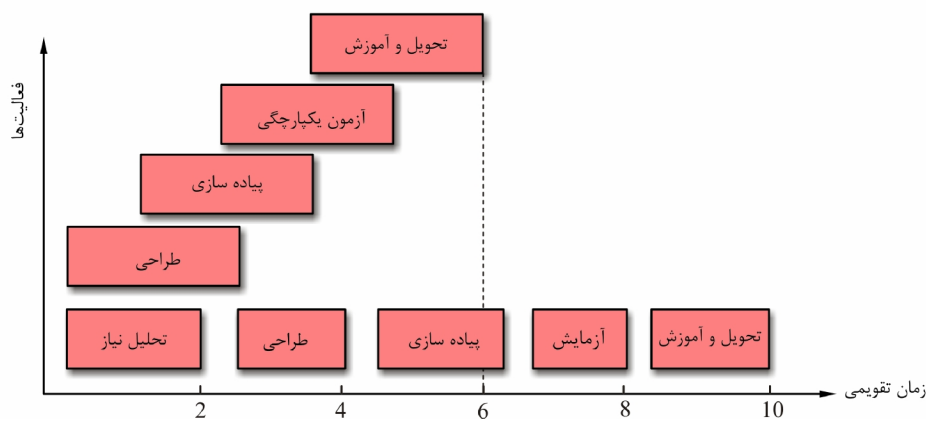
اجزای سیستم به صورت مؤلفه‌هایی در نظر گرفته می‌شوند که از قابلیت ساخت، آزمون و استفاده مناسبی برخوردار هستند. هر مدل یا روشی در توسعه نرم افزار در اینجا استفاده شود، فرقی نمی‌کند. در این مدل هدف و تأکید بر قابلیت استفاده مجدد از مؤلفه‌هاست.



شکل ۱۰-۲ مدل توسعه مبتنی بر مؤلفه

### ۵-۶-۲ مدل توسعه همروند<sup>۲</sup>

در مدل توسعه همروند<sup>۲</sup> فعالیت‌ها به منظور کاهش زمان ایجاد نرم افزار به طور موازی انجام می‌گیرند.



شکل ۱۱-۲ مدل توسعه همروند

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

1 - Component Assembly Model  
2 - Concurrent Development Model

## ۲-۶-۶ مدل روش‌های رسمی<sup>۱</sup>

مجموعه‌ای از فعالیت‌ها که نرم‌افزار رایانه‌ای را در قالب روابط ریاضی طراحی و بیان می‌کند.

مزایا	معایب
✓ بدیع	✓ وقت‌گیر و پرهزینه
✓ سازگار	✓ کاربرد محدود
✓ بدون ابهام	✓ نیاز به آموزش جامع
✓ کاربرد در سیستم‌های حساس	✓ مشکل ارتباط با مشتری
✓ و ...	✓ و ...

## ۲-۶-۷ ابزارهای نسل چهارم<sup>۲</sup>

شامل محدوده وسیعی از ابزارهای نرم‌افزاری است که قادر به تعیین و مشخص کردن بعضی از خصوصیات و ویژگی‌های نرم‌افزار در سطح بالاست و می‌تواند حتی کدی بر مبنای مشخصات تولیدشده به صورت خودکار ایجاد کند. در این ابزارها قابلیت‌های زیر وجود دارد:

- قابلیت استفاده برای مستندسازی
- قابلیت استفاده برای مدل‌سازی سیستم
- قابلیت تولید گزارشات
- قابلیت استفاده آسان از امکانات محیط برنامه‌سازی
- قابلیت مدیریت آسان پایگاه داده‌ها
- قابلیت تولید کد برنامه
- و ...

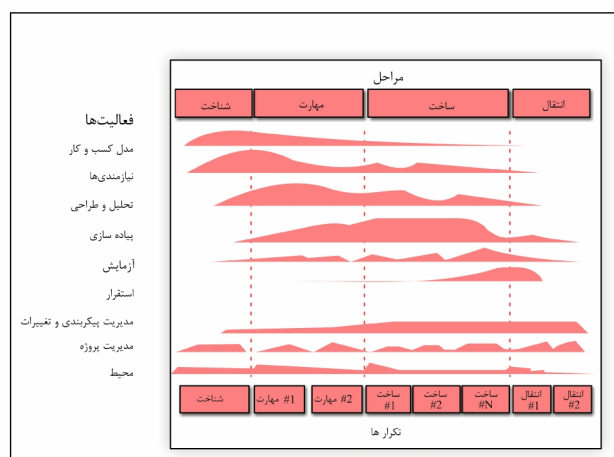
عیب

- تکنیک‌ها و ابزارهای موجود در حال حاضر خیلی عام نیستند و کاربردهای خاص دارند.

## ۲-۶-۸ فرایند مدل‌سازی یکنواخت<sup>۳</sup>

در اواخر دهه ۹۰ میلادی به منظور ایجاد یک بستر یکسان برای انجام فرایندهای مهندسی نرم‌افزار چارچوبی توسط شرکت Rational Rose تحت عنوان RUP<sup>۴</sup> ارائه شد که در آن چهار فاز اصلی و نه نظام کاری تعریف شد.

خروجی‌های هر فاز در چارچوب مستندات مشخص تدوین و جریان کار نظام‌های کاری نیز مشخص شده است. در این فرایند ایده تکراری<sup>۵</sup> بودن فرایند مهندسی نرم‌افزار نیز مد نظر بوده که در هر فاز تکرارهای متفاوتی را می‌توان اجرا کرد. چارچوب RUP به صورت زیر است.



شکل ۲-۱۲ چارچوب RUP

1 - Formal Methods Model  
 2 -4GT-Forth Generation Tool  
 3 -Unified Modeling Process  
 4 -Rational Unified Process  
 5 -Iterative