



# مهندسی نرم افزار

سید کاظم شکفته

موسسه آموزش عالی فردوس مشهد

نیمسال دوم ۸۹-۱۳۸۸

## فصل دوم

مهندسی نرم افزار و فرایندهای توسعه نرم افزار

# فرایندهای توسعه نرم افزار

- فرایند نرم افزار: مجموعه فعالیت هایی که هدفشان توسعه و تکامل نرم افزار می باشد.
- تعریف مهندسی نرم افزار از دیدگاه IEEE: کاربرد روش های اصولی، دارای نظم و قابل سنجش در توسعه، عملکرد و نگهداری سیستم های نرم افزاری.
- تعریفی دیگر از مهندسی نرم افزار: مهندسی نرم افزار انضباط تولید بدون نقص نرم افزار است مشروط بر آنکه به موقع با بودجه مشخص و برآوردن نیازهای کاربر همراه باشد.

# فرایندهای توسعه نرم افزار

- شامل فعالیت های اساسی زیر است :

۱- تعیین مشخصات نرم افزار

۲- توسعه نرم افزار

۳- اعتبار سنجی نرم افزار

۴- تکامل نرم افزار

# عمر نرم افزار

- نرم افزا شبیه انسان است.
- فاصله زمانی بین تصور ایجاد نرم افزار تا زمان منسوخ شدن نرم افزار.

# مدل‌های عمر نرم‌افزار

- شامل فرایندها، روش‌ها و ابزارهای لایه و فازهای عمومی است.

1. مدل عمر ساخت و ترمیم (Build & Fix)
2. مدل عمر آبشاری (Waterfall)
3. مدل عمر نمونه سازی
4. مدل عمر افزایشی
5. مدل عمر توسعه سریع نرم‌افزار (Rapid Application Development / RAD)
6. مدل عمر مارپیچی (Spiral)
7. مدل توسعه بر مبنای مولفه
8. مدل روش‌های رسمی (Formal Methods)

# ۱. مدل عمر ساخت و ترمیم (Build & Fix)

- مبتنی بر پیاده‌سازی است.
- در آن هیچ تحلیل و طراحی صورت نمی‌گیرد.
- معمولاً با تست‌های مختلف، مشکلات آن مشخص و رفع می‌گردد.
- اغلب برای پروژه‌های کوچک مناسب است.

## • مزیت

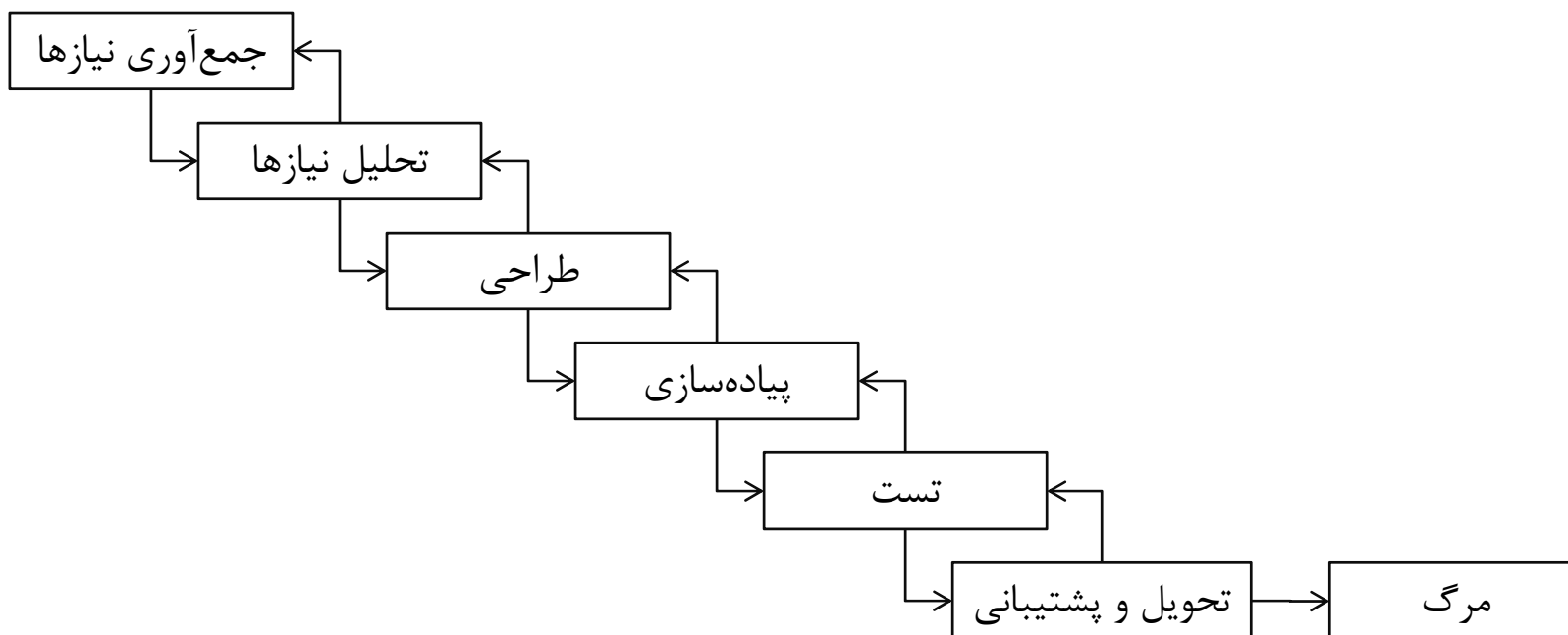
- در زمان توسعه برای پروژه‌های کوچک ممکن است صرفه‌جویی زمانی داشته باشد.

## • عیب

- عدم آشنایی کامل با صورت مسئله ممکن است باعث طولانی شدن زمان پیاده‌سازی گردد.

## ۲. مدل عمر آبشاری (Waterfall / ترتیبی خطی)

- مدل عمر کلاسیک یا مدل خطی سری نیز نامیده می شود.
- یک روش خطی و سیستماتیک است.



## ۲. مدل عمر آبخاری (Waterfall / ترتیبی خطی) ...

### • مزیت

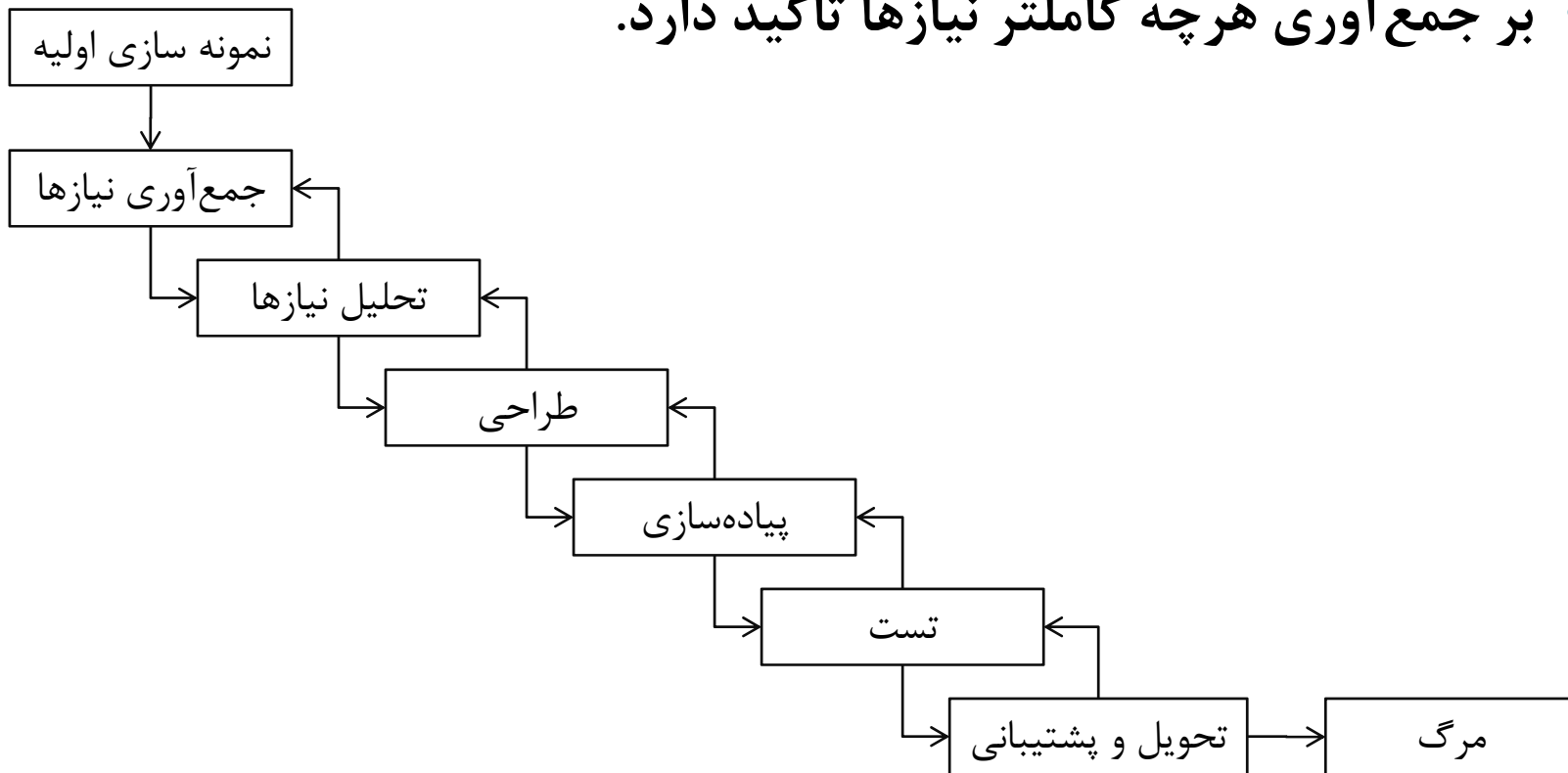
- در مقایسه با مدل قبل، مراحل توسعه نرم افزار به مراحل کوچکتر شکسته شده.

### • معایب

- معمولاً پروژه‌های واقعی به ندرت جریان ترتیبی ارائه شده توسط این مدل را دنبال می‌کنند.
- معمولاً مرحله جمع‌آوری نیازها به طور کامل انجام نمی‌گیرد. دلیل؟
- به دلیل خطی بودن انجام فعالیت‌ها، مشتری باید تأخیرهای احتمالی را بپذیرد.
- زمانبندی دقیق مشکل است.

## ۳. مدل عمر نمونه سازی

- گاهی از کارایی الگوریتم خاص، یا توانایی یک سیستم عامل، یا شکلی که انسان و ماشین با هم ارتباط برقرار می کنند، اطمینان نیست.
- بر جمع آوری هرچه کاملتر نیازها تأکید دارد.



## ۳. مدل عمر نمونه‌سازی ...

### • مزیت

- در مقایسه با مدل‌های قبل، جمع‌آوری نیازهای توسعه نرم‌افزار به شکل کاملتری انجام می‌شود.

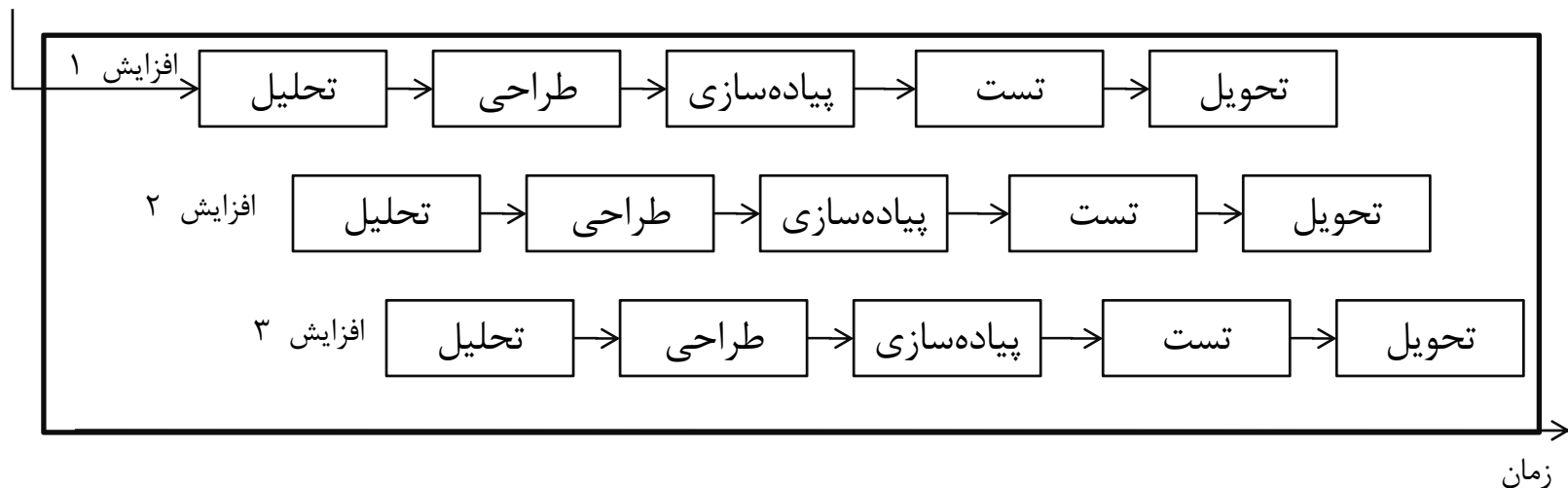
### • معایب

- مشتری شکل اجرایی نرم‌افزار را می‌بیند و آگاهی ندارد که این نمونه ظاهری و موقتی است و نکات کیفیت نرم‌افزار و قابلیت نگهداری برای مدت طولانی برای آن در نظر گرفته نشده است.
- توسعه دهنده اغلب نکاتی را در پیاده‌سازی نادیده می‌گیرد تا یک نمونه اجرایی را به سرعت ایجاد کند.

## ۴. مدل عمر افزایشی

- اجزاء مدل آبشاری را به صورت تکراری همراه با فلسفه تکرار نمونه‌سازی ترکیب می‌کند.
- هر گام توسعه نرم‌افزار تا زمان تحویل نهایی ادامه می‌یابد.

جمع‌آوری نیازها



## ۴. مدل عمر افزایشی ...

- مزیت

- مراحل توسعه نرم افزار به مراحل که ریسک کمتری دارند شکسته می شود.

- معایب

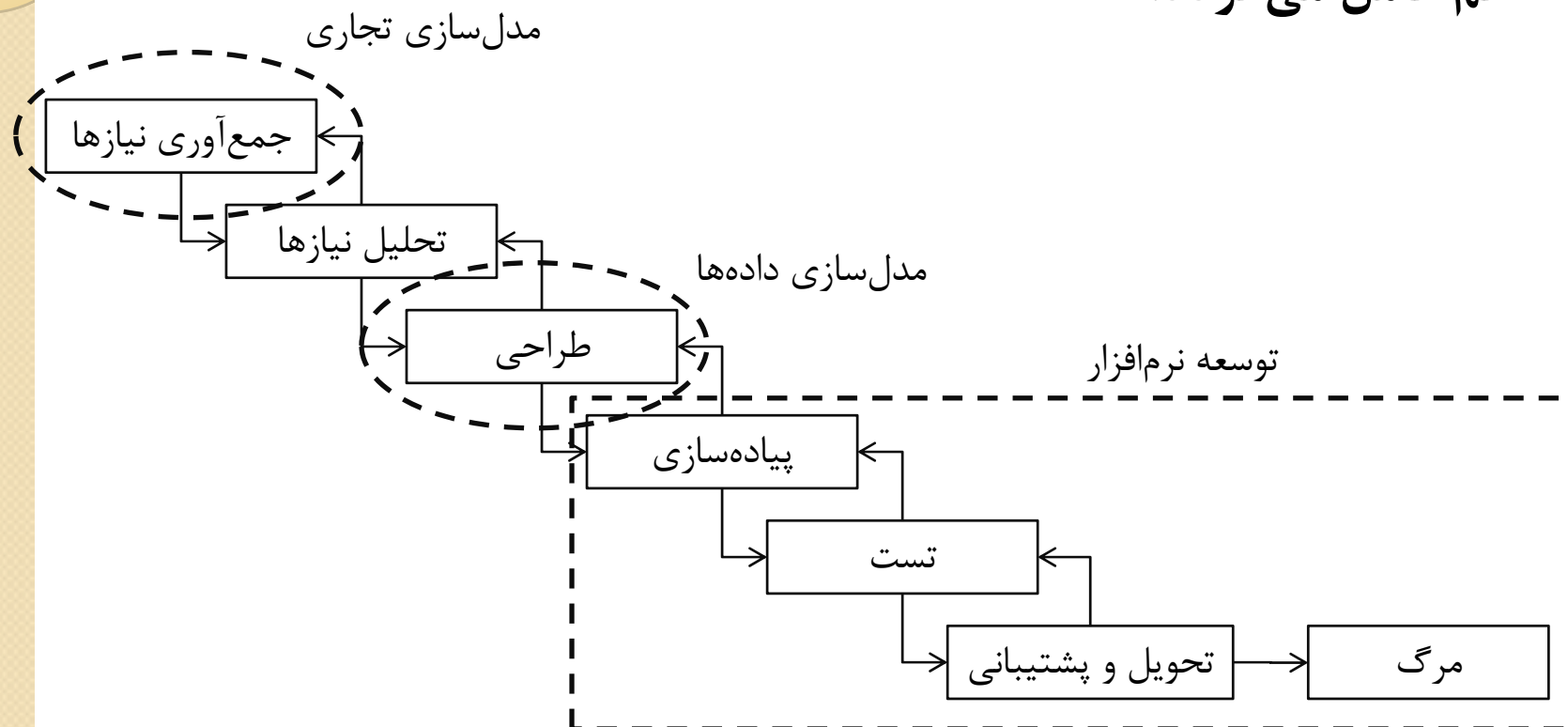
- هزینه جانبی انتقال از یک فاز به فاز دیگر زیاد می شود.
- مشخص نیست تعداد مراحل چندتا باشد تا هزینه اضافی نداشته باشد.
- مدیریت باتجربه نیاز دارد.

## ۵. مدل عمر توسعه سریع نرم افزار (RAD)

- یک فرایند توسعه افزایشی است که تأکید زیاد بر روی دوره کوتاه توسعه دارد.
- برای توسعه نرم افزارهای بانک اطلاعاتی کاربرد دارد.
- اگر لوازم کاملاً مشخص باشد و محدوده پروژه تعیین شده باشد، تیم توسعه، یک سیستم کامل را در دوره زمانی کوتاه (۶۰ تا ۹۰ روز) ایجاد می نماید.
- مراحل این مدل
  - مدل سازی تجاری
  - مدل سازی داده
  - مدل سازی فرایند
  - توسعه نرم افزار

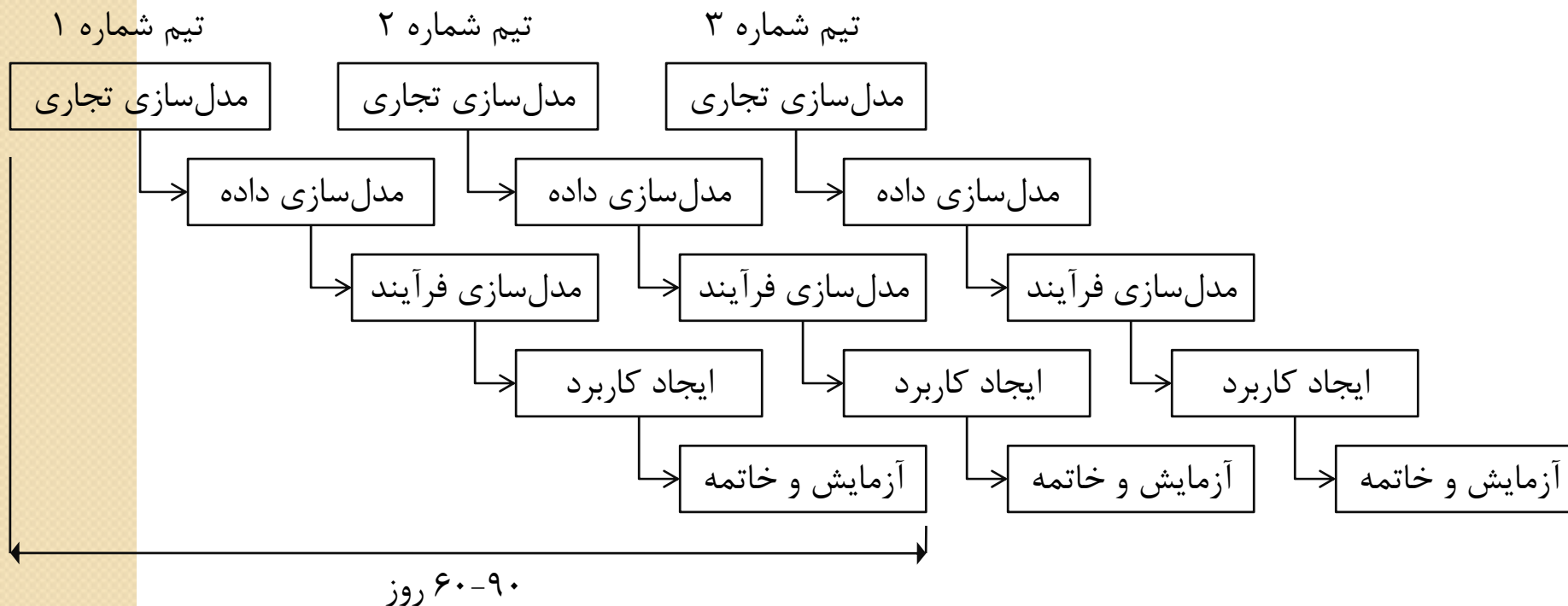
## ۵. مدل عمر توسعه سریع نرم افزار (RAD) ...

- معمولاً هر مرحله از ساخت در ۶۰ تا ۹۰ روز انجام می‌گیرد و نرم‌افزار کم‌کم کامل می‌گردد.



# ۵. مدل عمر توسعه سریع نرم افزار (RAD) ...

- تیم‌های تولیدکننده نرم افزار غالباً به صورت موازی همکاری می کنند.



## ۵. مدل عمر توسعه سریع نرم افزار (RAD) ...

### • مزایا

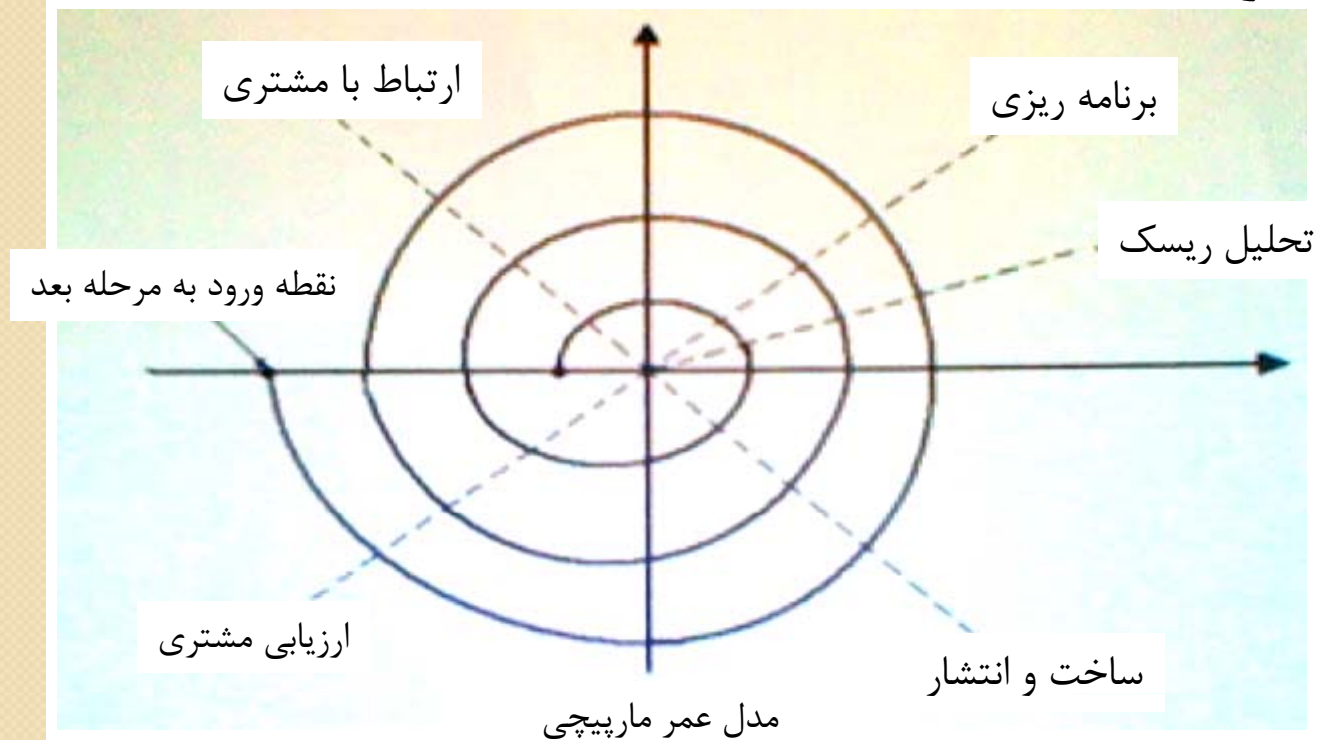
- سعی می شود تا حد امکان از قطعات پیش ساخته استفاده شود.
- نرم افزارهای متوسط و کوچک با استفاده از این روش توسعه می یابند.

### • معایب

- برای پروژه های بزرگ ولی قابل تغییر، نیازمند منابع انسانی کافی برای سازماندهی تعداد تیمهای مورد نیاز خود است.
- نیازمند توسعه دهنده ها و مشتری هایی است که بر انجام فعالیت های صحیح مورد نیاز برای تکمیل سیستم در کمترین زمان ممکن توافق داشته باشند.
- تمام کاربردها برای RAD مناسب نیستند.
- RAD در زمانی که ریسک های تکنیکی بالا وجود دارد مناسب نیست.

## ۶. مدل عمر مارپیچی (Spiral / حلزونی)

- فرآیند توسعه ماهیت تکرارپذیر و ترتیبی دارد.
- ممکن است یک سری از مراحل چندین بار تکرار شوند.
- توسط Boehm ارائه شد.



## ۶. مدل عمر مارپیچی (Spiral / حلزونی) ...

### • مراحل

- ارتباط با مشتری
- برنامه ریزی
- تحلیل ریسک
- پیاده سازی یا توسعه (ساخت و انتشار)
- ارزیابی مشتری

## ۶. مدل عمر مارپیچی (Spiral / حلزونی) ...

- مزایا

- توجه خاص به تحلیل ریسک دارد.

- معایب

- معمولاً تحلیل ریسک در هر دوره تکامل نرم افزار هزینه زیادی دارد.

- استفاده از این روش نیاز به تجربه کافی دارد.

- به جای بررسی کافی در مراحل توسعه نرم افزار به مباحث خاص مدیریتی می پردازد.

## ۷. مدل توسعه بر مبنای مولفه (CBD)

- روشی تکراری-افزایشی که سعی در استفاده از مولفه‌های از پیش ساخته برای تولید نرم‌افزار دارد.
- بسیاری از خصوصیات مدل مارپیچی را داراست.
- ماهیتی تکاملی دارد و روشی متفاوت برای ایجاد نرم‌افزار می‌باشد.
- زبان‌های مختلفی برای مدلسازی سیستم در این روش وجود دارند: UML (زبان مدلسازی یکپارچه).

## ۷. مدل توسعه بر مبنای مولفه (CBD) ...

- مزایا

- زمان توسعه با استفاده از مولفه‌های آماده کاهش پیدا می‌کند.
- **Component**ها مونتاژ می‌شوند در نتیجه زمان کاهش می‌یابد.

# ۸. مدل عمر روش‌های رسمی

- مجموعه‌ای از فعالیت‌ها که به مشخصه‌ی ریاضی نرم‌افزار و بیان ریاضی آن می‌پردازد.
- روش کلی نیست اما نرم‌افزار بدون خطا را تولید می‌کند.
- از جمله زبان‌های مورد استفاده در این روش: زبان Z.
- از جمله موارد استفاده: سیستم‌های کنترل بسیار حساس، کنترل خطوط هوایی، ریل راه‌آهن و سیستم‌هایی با محدودیت‌های زیاد.

# ۱. مدل عمر روش‌های رسمی ...

- مزایا

- دقت فوق العاده در کیفیت نرم‌افزار.

- معایب

- توسعه بسیار گران بر اساس این مدل.

- نیاز به تخصص زیاد در این روش.

- از این روش‌ها برای ارتباط با مشتری به سختی می‌توان استفاده کرد.

# نیاز

- نیاز یا خواسته به معنای شرط، قابلیت یا رفتاری سیستماتیک و الگوریتمی که سیستم باید داشته باشد.

خواسته  
نیاز

<b>If (x == 1) then</b>	شرط
<b>x = x + 1;</b>	رفتار

- بر اساس مدل FURPS+ تشخیص داده می شوند.
- به طور کلی نیازها دسته هستند
  ۱. نیازهای عملکردی (functional)
  ۲. نیازهای غیر عملکردی (non functional)

# ۱. نیازهای عملکردی (Functionality)

- نیازهایی که مجموعه عملکرد و سرویس‌هایی که یک نرم‌افزار ارائه می‌دهد را توصیف می‌نماید.
- صدور سند در یک سیستم حسابداری نمونه‌ای از این نیازهاست، چرا که صدور سند جزئی از علم حسابداری است.

## ۲. نیازهای غیر عملکردی

- مجموعه نیازهایی که جزء نیازهای عملکردی نیست اما نرم افزار به آنها از حیث دقت یا صحت عملکرد نیاز دارد.
- به عنوان مثال وجود مکانیزم‌های امنیتی مانند **username** و **password** که برای ورود به سیستم الزامی است.

## ۲. نیازهای غیر عملکردی ...

- انواع نیازهای غیرعملکردی

۲.۱. نیازهای قابلیت استفاده (Usability): قابلیت‌های استفاده از نرم‌افزار از جمله نیازهای یک نرم‌افزار هستند. مانند وجود راهنما و ابزار آموزشی جهت استفاده از آن نرم‌افزار.

۲.۲. نیازهای اعتباری (Reliability): مجموعه نیازهایی که میزان خطاها، تعداد میانگین زمان وقوع رخداد و قابلیت تحمل خطای نرم‌افزار را نشان می‌دهند. مانند خطاهایی که در برنامه برای آنها پیش بینی نکرده‌ایم، باید قابلیت تحمل داشته باشند.

۲.۳. نیازهای کارایی (Performance): کارایی نرم‌افزار از حیث پارامترهای متعددی چون سرعت اجرای برنامه، زمان تولید پاسخ خروجی و ... را نشان می‌دهد. مانند یک بازی کامپیوتری که روی کامپیوتری خاص اجرا نشوند.

## ۲. نیازهای غیر عملکردی ...

- انواع نیازهای غیرعملکردی ...

۲.۴. نیازهای پشتیبانی (Supportability): قابلیت اعمال تغییرات در نرم افزار، قابلیت نصب و راه اندازی، مستندات و ارائه خدمات نرم افزاری را نشان می دهد.

۲.۵. نیازهای جانبی: استانداردهای طراحی و پیاده سازی نرم افزار، استفاده از زبان های برنامه نویسی خاص را توصیف می کند.

## نیازسنجی ...

- به عنوان نمونه یک نرم افزار ماشین حساب، محاسبات ۴ عمل اصلی را انجام می دهد. این ماشین حساب، محاسبات را تا ۹ رقم اعشار و برای اعداد تا ۵۰ رقم انجام می دهد. نیازهای نرم افزار برا اساس مدل FURPS+ چیست؟

## نیازسنجی ...

- برای تجزیه یک سیستم به زیر سیستم‌های آن باید عملکردهای سیستم را به طور کامل لیست نمود.
- تقسیم بندی نیازها باعث تقسیم بندی سیستم به زیر سیستم‌های کوچکتر خواهد شد.

# مدیریت نیازها

- تعریف: فرآیند سیستماتیک یافتن، سازماندهی، مستندسازی و پیگیری تغییرات یک سیستم را مدیریت نیازها گویند.
- فردی که مدیریت نیازهای یک پروژه را انجام می دهد چه نامیده می شود؟

# فرایند مدیریت نیازها

۱. فهم دامنه‌ی (Domain) مسئله ⇔ فهم نیازهای عملکردی.
۲. جمع آوری نیازهای مربوط به کاربر.
۳. دسته بندی نیازها.
۴. اولویت دهی به نیازها.
۵. چک کردن نیازها.

# مشکلات موجود در مدیریت نیازها

- نیازها همیشه کاملا مشخص نیستند.
- همیشه نمی توان نیازها را در قالب کلمات بیان کرد.
- بعضی از نیازها محدود نیستند و نیاز به سطوح بیان جزئیات مختلفی دارند.
- بعضی از نیازها اگر کنترل نشوند و حد و مرزی برای محدود کردن آنها در نظر گرفته نشوند، غیرقابل مدیریت خواهند بود.
- تغییراتی که در نیازهای کاربر رخ می دهند.
- نیازهایی که در محدوده زمانی خاص معتبر بوده و به زمان وابسته اند.