

# مبانی مهندسی نرم افزار

# فصل دوم

فرآیند

## فرآیند (Process)

- فرآیند چیست؟ وقتی که محصول یا سیستمی را می‌سازیم این امر اهمیت دارد که از یکسری مراحل قابل پیش بینی استفاده کنیم. نقشه مسیری که به ما کمک می‌کند تا نتایجی را در زمان معین و با کیفیت یا لا ایجاد کنیم فرآیند نرم افزار نام دارد.
- چه کسی این کار را انجام می‌دهد؟ مهندسين نرم افزار و مدیران آن‌ها فرآیند را تحت نیاز خود انتخاب کرده و آن را دنبال می‌کنند.
- چرا فرآیند مهم است؟ فرایند به کار ما سامان می‌بخشد و فعالیت‌های موجود را کنترل نموده و نتایج ثابتی را برای آن‌ها پیش بینی می‌کند. در غیراینصورت، کار ما دچار هرج و مرج و بی‌نظمی خواهد شد.
- مراحل انجام کار چیست؟ فرآیندی که از آن استفاده می‌کنیم وابسته به نرم‌افزاری است که آن را می‌خواهیم ایجاد کنیم، پس انتخاب فرآیند براساس نوع نرم‌افزار صورت می‌گیرد.
- محصول کار چیست؟ از دیدگاه یک مهندس نرم‌افزار، برنامه‌ها، اسناد و داده‌هایی است که طی مراحل مهندسی در یک فرآیند به دست می‌آیند.

## مهندسی نرم افزار

- کاربرد سیستماتیک منظم و رویکرد قابل سنجش برای توسعه ، عملکرد و نگه داری نرم افزار.
- مطالعه ی رویکردهای کاربردی فوق الذکر
- مهندسی نرم افزار یک تکنولوژی لایه ای است.

## تکنولوژی لایه ای



■ مهندسی نرم افزار باید تعهدی برای فراهم آوردن کیفیت داشته باشد.

■ تمرکز کیفیت عنصری پایه ای برای پشتیبانی از مهندسی نرم افزار است.

■ مهندسی نرم افزار روش ها و ابزارهای فرایند است.

■ لایه فرایند

■ لایه های تکنولوژی را در کنار هم نگه می دارد.

■ یک چارچوب تعریف می کند.

■ اساس کنترل و مدیریت را تشکیل می دهد.

■ برقرار می کند زمینه ای را برای :

■ روش های تکنیکی که به کار گرفته می شوند.

■ محصولات کاری که تولید می شوند.

■ نقاط عطفی (milestone) که باید ایجاد شوند.

■ اطمینان از کیفیت

■ مدیریت تغییرات

■ روش ها (methods) چگونگی انجام کارها را مشخص می کنند.

■ این کارها شامل ارتباط . تحلیل نیازمندی ها . مدل سازی طراحی . ساخت برنامه . تست و پشتیبانی از آن است.

■ ابزارها از فرایند ها و روش ها بصورت اتوماتیک و یا نیمه اتوماتیک پشتیبانی میکنند.

## مراحل مهندسی نرم افزار

□ فعالیت های مرتبط با مهندسی نرم افزار را صرف نظر از اندازه و پیچیدگی پروژه و همچنین زمینه کاربردی آن می توان به سه مرحله یا فاز (Phase) کلی زیر تقسیم نمود.

□ مرحله تعریف (Definition)

□ مرحله توسعه (Development)

□ مرحله نگهداری (Maintenance)

## مرحله تعریف

- در این مرحله، تمرکز بر تعیین **“What”** می باشد.
- تولیدکننده نرم افزار سعی دارد داده های مورد پردازش، وظایف، عملکردها و رفتار سیستمی مورد انتظار، واسط ها، قیود و شرایط طراحی و همچنین معایرهای تشخیص اعتبار سیستم را تعیین نماید.
- در این مرحله، سه فعالیت عمده زیر انجام می شود:
  - تجزیه و تحلیل خواسته ها و نیازمندی ها (مهم)
  - مهندسی اطلاعات یا سیستم
  - برنامه ریزی پروژه نرم افزاری

## مرحله توسعه

- در این مرحله، تمرکز بر تعیین چگونگی **“How”** می باشد.
- مهندس نرم افزار سعی دارد چگونگی ساختمان داده ها، چگونگی پیاده سازی وظایف، چگونگی واسط ها و نحوه تبدیل طراحی به یک زبان برنامه سازی و همچنین چگونگی آزمون نرم افزار را کاملا مشخص نماید. در این مرحله، سه فعالیت عمده زیر انجام می شود:
  - طراحی نرم افزار
  - پیاده سازی نرم افزار و تولید کد
  - آزمون نرم افزار



## مرحله نگهداری

□ تمرکز این مرحله بر " تغییر " است. در طول این مرحله ممکن است با 4 نوع

تغییر زیر مواجه شویم:

۱. نگهداری تصحیحی:

۲. نگهداری تطبیقی:

۳. نگهداری تکمیلی:

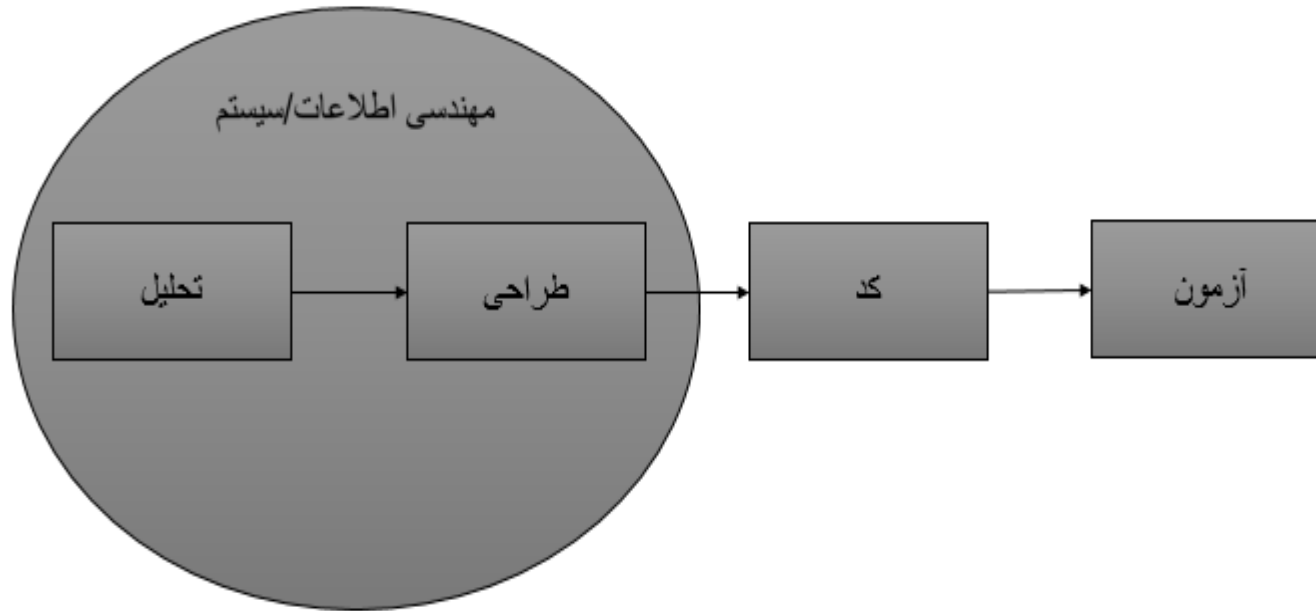
۴. نگهداری پیشگیری کننده:

## مدل های فرایند نرم افزار

- برای حل مسائل واقعی در یک صنعت یا یک شرکت، یک مهندس نرم افزار یا سیستم مهندسی باید یک استراتژی توسعه یا متدولوژی انتخاب کند، به نحوی که شامل لایه های متدها، فرآیندها و ابزارهای شرح داده شده باشد.
- این فرایند گاهی اوقات تحت عنوان مدل های تولید نرم افزار نامیده می شود.
- هر مدل فرایند دارای دیدگاه های گوناگونی است. مثلا:
  - جریان کاری (Work flow): ترتیب فعالیت ها.
  - جریان داده (Data flow): داده ها چگونه و از چه جاهایی جریان می یابند.
  - نقش / عمل (Role/Action): چه کسی چه کاری را انجام می دهد.

# مدل ترتیبی خطی

- مشهور به مدل «چرخه حیات کلاسیک توسعه سیستم»
- ساده ترین و کارآمدترین روش برای تولید نرم افزارهای نسبتاً محدود است که در آن تحلیل نیازمندیها و تعریف صورت مساله به سادگی انجام می شود.



# مشکلات مدل ترتیبی خطی

1. پروژه های واقعی به ندرت جریان ترتیبی پیشنهاد شده توسط این مدل را دنبال می کنند. اگرچه مدل خطی می تواند پذیرای تکرار باشد، این عمل را به طور غیرمستقیم انجام می دهد. در نتیجه، با پیش رفتن تیم پروژه، ممکن است تغییرات باعث ایجاد سردرگمی شوند.
2. غالباً برای مشتری دشوار است که همه نیازهای خود را به وضوح بیان کند. مدل ترتیبی خطی، به بیان واضح نیاز دارد و به خوبی از پس موارد غیرقطعی که در آغاز اکثر پروژه ها وجود دارند، برنمیآید.
3. مشتری باید حوصله داشته باشد. یک نسخه کاری از برنامه ها تا آخرین روهای پروژه در دسترس او قرار نخواهد گرفت. یک اشتباه عمده که تا زمان بازبینی برنامه کاری از دید پنهان بماند، می تواند بسیار دردسرافرین باشد.

## مدل توسعه کاربردی سریع (RAD)

□ این مدل، یک مدل فرآیندی توسعه نرم افزار ترتیب خطی است که همان گونه که از نامش پیداست، قصد دارد فرآیند توسعه را تسریع کند.

□ دلایل نیاز به روشی جهت تهیه سریع نرم افزار:

۱. فناوری ها و کاربرد آن ها در مهندسی نرم افزار به سرعت پیشرفت دارد.
۲. تهیه یک سیستم بزرگ از تعریف سیستم تا پیاده سازی آن به زمان زیادی وقت نیاز دارد.
۳. محصول نرم افزاری تهیه شده ممکن است جوابگوی نیازهای روز نباشد.

## مدل توسعه کاربردی سریع (RAD)

- ✓ در این مدل، کارهای نرم افزاری به بخش های مختلفی تقسیم شده و افراد تیم پروژه روی بخش های تقسیم شده متمرکز شده و کار شروع می شود.
- ✓ در پایان کار نتیجه حاصله از هر بخش با یکدیگر تلفیق می شود تا محصول نهایی ساخته شود.
- ✓ آیا سیستم قابلیت تفکیک پذیری دارد؟؟؟؟؟؟
- ✓ آیا می توان همه کارها را در یک زمان به پایان رساند تا امکان ترکیب و تحویل دهی آن میسر باشد؟؟؟؟؟؟؟؟
- ✓ مدیریت چنین پروژه هایی ساده نیست زیرا با دو مشکل افراد و زمان محدود روبرو هستیم.

## مدل توسعه کاربردی سریع (RAD)

✓ این مدل از دسته مدل های تکراری است و نیاز به ایجاد یک یا چند مدل نمونه دارد.

✓ این مدل معمولا برای پروژه های نسبتا بزرگ کاربرد دارد و افراد تیم باید نهایت همکاری و دقت را داشته باشند تا بتوانند در موعد مقرر تولید نهایی را به مشتری تحویل دهند، چون اگر یک قسمت از این پروژه انجام نشده باشد تحویل پروژه میسر نیست.

# مدل RAD

