

## فصل ۱- نرم افزار و مهندسی نرم افزار

### ۱. ماهیت نرم افزار

#### ۱.۱. ماهیت و تعریف نرم افزار

امروزه نرم افزار نقشی دو گانه دارد:

- نرم افزار نوعی محصول است.
- در عین حال وسیله نقلیه ای برای تحویل یک محصول است.

نرم افزار عبارت است از:

- دستورالعمل ها که هنگام اجرا، ویژگی، عملکرد و کارایی مطلوب را فراهم می سازند.
- ساختمان های دادهایی که برنامه ها را قادر به پردازش مناسب داده ها کنند.
- اطلاعات توصیفی در هر دو قالب کپی سخت و مجازی که راه اندازی و استفاده از برنامه ها را شرح دهند.

#### ۱.۲. خصوصیات نرم افزار

نرم افزار بیشتر یک عنصر منطقی است تا یک عنصر سیستمی فیزیکی. بنابراین نرم افزار با سخت افزار تفاوت چشمگیری دارد:

- نرم افزار، مهندسی و بسط داده می شود و چیزی نیست که به معنای کلاسیک کلمه ساخته شود.
- نرم افزار فرسوده نمی شود.
- گرچه صنعت در حال حرکت به سوی مونتاژ قطعات است، اکثر نرم افزارها همچنان به صورت سفارشی ساخته می شوند.

#### ۱.۳. نرم افزار، مهندسی و بسط داده می شود

تفاوت های بسط سخت افزار و نرم افزار:

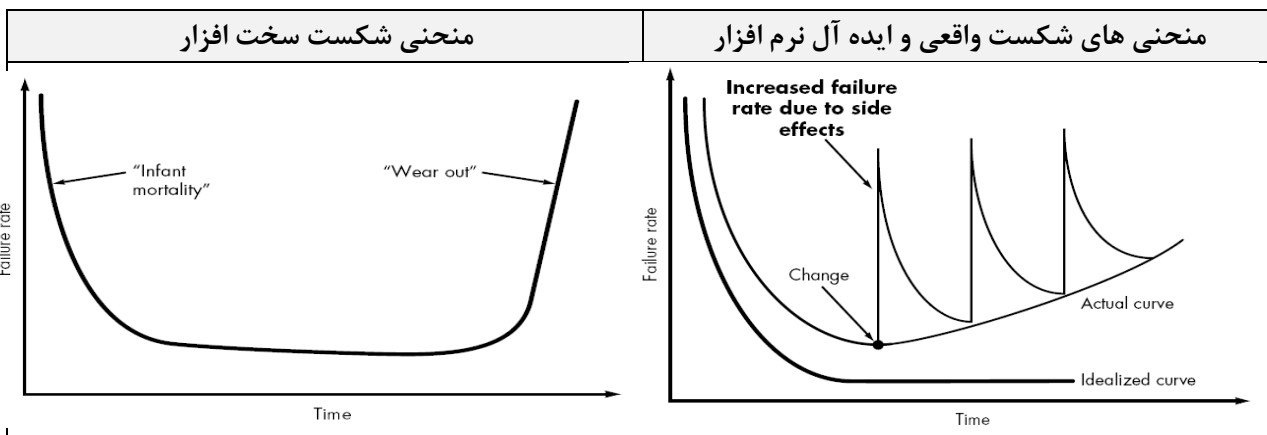
- فاز ساخت برای سخت افزار باعث بروز مشکلات کیفیتی می شود که برای نرم افزار وجود ندارند یا به راحتی قابل رفع است.
- هر دو عمل وابسته به انسان هستند ولی رابطه ی میان انسان و کاری که انجام می شود، کاملا متفاوت است.
- هر دو عمل مستلزم ساخت یک محصول هستند ولی روش ها متفاوت است.
- نرم افزارها ساخته نمی شوند، بلکه مهندسی می شوند.

#### ۱.۴. نرم افزار فرسوده نمی شود

منحنی شکست سخت افزار

- منحنی وانی نمودار آهنگ شکست را به صورت تابعی از زمان برای سخت افزار نشان می دهد.
- منحنی وانی، نشان می دهد که سخت افزار در ابتدای عمر آهنگ شکست شدیدی دارد.
- این عیوب، تصحیح و آهنگ شکست در یک دوره زمانی به مقداری ثابت نزول می کند.

- با گذشت زمان سخت افزار شروع به فرسایش کرده و دوباره آهنگ شکست شدت می گیرد.
- منحنی های شکست واقعی و ایده آل نرم افزار
- نرم افزار نسبت به ناملاپمات محیطی که باعث فرسایش آن می شود نفوذ پذیر نیست.
- بنابراین در تئوری، منحنی شکست برای نرم افزار باید شکل منحنی ایده آل را به خود بگیرد.
- منحنی ایده آل نسبت به منحنی واقعی مدل های شکست نرم افزار، بسیار ساده تر است.
- ولی، معنای آن واضح است، نرم افزار هیچ گاه دچار فرسایش نمیشود بلکه زوال می یابد!
- نرم افزار در دوران حیات خود دستخوش تغییر می شود.
- با اعمال این تغییرات احتمال دارد که برخی عیوب جدید وارد شوند و باعث خیز منحنی آهنگ شکست می شود.
- و پیش از آن که منحنی بتواند به آهنگ شکست منظم اولیه خود برسد، تغییر دیگری درخواست می شود که باعث خیز دوباره منحنی می شود.



### ۱.۵. نرم افزارها به صورت سفارشی ساخته می شوند

در جهان سخت افزار، استفاده مجدد از قطعات بخشی طبیعی از فرایند مهندسی است. در مهندسی نرم افزار این امر به تازگی مورد توجه قرار گرفته است.

## ۲. دامنه های کاربرد نرم افزار

### ۲.۱. انواع نرم افزارها

نرم افزارهای	توضیح
سیستمی	مجموعه ای از برنامه هاست که برای سرویس دهی به برنامه های دیگر نوشته شده اند.
کاربردی	برنامه های مستقلی که یک نیاز تجاری مشخص را بر طرف می کند.
مهندسی - علمی	نرم افزارهای علمی توسط الگوریتم هایی مشخص می شوند که اعداد و ارقام را پردازش می کنند. کاربردهای نوین در حیطه مهندسی و علمی از الگوریتم های عددی مرسوم فراتر رفته اند.
تعبیه شده	در حافظه فقط خواندنی جای دارند و برای کنترل محصولات و سیستم های مربوط به بازارهای صنعتی و مصرفی به کار می رود.
خط تولید	برای فراهم آوردن یک قابلیت خاص جهت استفاده توسط بسیاری از مشتریان مختلف طراحی می شوند.
کاربردی تحت وب	این گروه از نرم افزارهای شبکه ای شامل مجموعه ی گسترده ای از برنامه های کاربردی می باشد.

هوش مصنوعی	برای حل مسائل پیچیده ای که به روش های عددی قابل حل نیستند، از الگوریتم های غیر عددی استفاده می کنند. سیستم های خبره، تشخیص الگوها، شبکه های عصبی مصنوعی، اثبات قضایا و بازی مثال هایی از کاربرد این گروه هستند.
------------	---

## ۲.۲. مسائل جدید کاربردهای نرم افزار

مسائل جدید	توضیح
کار با کامپیوتر در جهانی باز	چالشی که مهندسان فرا روی خود خواهند داشت، توسعه ی سیستم ها و برنامه های کاربردی است که برقراری ارتباط میان کامپیوترهای شخصی، دستگاه های همراه و سیستم های اداری را از طریق شبکه های گسترده میسر می سازند.
کد منبع باز	تمایل رو به رشدی است که منجر به توزیع کدهای منبع سیستم ها و برنامه های کاربردی شده است به طوری که افراد بسیاری بتوانند در توسعه آن سهیم شوند.

## ۲.۳. نرم افزارهای قدیمی

- سیستم های نرم افزاری قدیمی چند دهه قبل ساخته شده اند و پیوسته اصلاح شده اند تا تغییرات به عمل آمده درخواست های تجاری و سکویهای محاسباتی را پاسخگو باشند.
- ازدیاد این گونه سیستم ها باعث دردسر برای سازمان های بزرگی می شود که نگهداری از آن ها را پر هزینه و تکامل بخشیدن به آن ها را خطرناک می دانند.

## ۲.۴. دلایل تکامل سیستم های قدیمی

- نرم افزار باید برای برآورده ساختن نیازهای محیط های جدید کامپیوتری یا فناوری های جدید اصلاح گردد.
- نرم افزار باید بهبود یابد تا خواسته های تجاری جدید را پیاده سازی کند.
- نرم افزار باید گسترش داده شود تا با سایر سیستم ها یا بانک اطلاعاتی جدیدتر، قابلیت همکاری را داشته باشد.
- نرم افزار باید دوباره معماری شود تا در یک محیط شبکه نیز قادر به ادامه ی حیات باشد.

## ۳. ماهیت برنامه های کاربردی تحت وب

### ۳.۱. صفات برنامه های کاربردی تحت وب

امروزه برنامه های تحت وب به ابزارهای کامپیوتری پیچیده ای تکامل یافته اند که نه تنها عملکردی مستقل را در اختیار کاربر نهایی قرار می دهند بلکه با بانک های اطلاعاتی و برنامه های کاربردی تجاری یکی شده اند. در اکثریت وسیع برنامه های تحت وب این صفت ها مشاهده می شود.

صفات	توضیح
میزان تمرکز شبکه	برنامه های تحت وب روی یک شبکه قرار دارند و باید نیازهای جامعه ای متنوع از کلاینت ها را برآورده سازند.
همروندی	ممکن است یک باره تعداد بسیاری از کاربران به برنامه های تحت وب دسترسی داشته باشند.
بار غیر قابل پیش بینی	تعداد کاربران برنامه های تحت وب از روزی به روز دیگر ده یا صد برابر شوند.

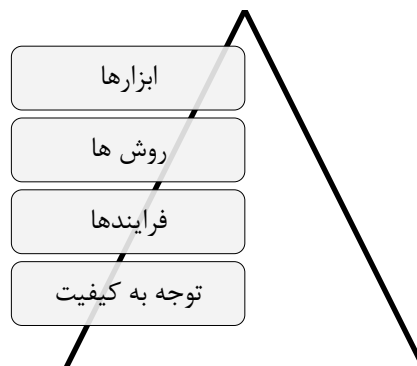
کارایی	اگر کاربر یک برنامه تحت وب باید یک مدت طولانی منتظر بماند، ممکن است تصمیم بگیرد به جای دیگری برود.
قابلیت دسترسی	گرچه انتظار ۱۰۰ در صد قابلیت دسترسی، غیر منطقی است، کاربران برنامه های تحت وب پر طرفدار غالباً تقاضای دسترسی ۲۴ ساعته در ۷ روز هفته و ۱۲ ماه سال را دارند.
داده محوری	عملکرد اصلی بسیاری از برنامه های تحت وب استفاده از ابر رسانه ها برای ارائه متون به کاربران نهایی است.
حساس به محتویات	کیفیت و ماهیت زیبا شناختی محتویات از جمله مهمترین عوامل تعیین کننده کیفیت در برنامه های تحت وب است.
تکامل پیوسته	برخلاف نرم افزارهای کاربردی سنتی، برنامه های تحت وب پیوسته در حال تکامل هستند.
بی واسطگی	نیاز اجباری برای رساندن سریع نرم افزار به بازار است.
امنیت	برای محافظت از محتویات حساس معیارهای امنیتی قوی ای باید پیاده سازی شود.

## ۴. مهندسی نرم افزار

### ۴.۱. نکات کلیدی در ساخت نرم افزار

- پیش از آن که برای مسئله راهکاری بیابید آن را درک کنید.
- یکی از فعالیت های محوری در مهندسی نرم افزار طراحی است.
- کیفیت و قابلیت نگهداری هر دو نتیجه طراحی خوب هستند.

### ۴.۲. لایه های مهندسی نرم افزار



## ۵. فرایند نرم افزار

### ۵.۱. فرایند نرم افزار

فرایند مجموعه ای از فعالیت ها، کنش ها و وظایف است که هنگام ایجاد یک محصول کاری اجرا می شوند.

- یک فعالیت، کوششی است در جهت رسیدن به هدفی گسترده
- یک کنش شامل مجموعه ای از وظایف است که یک محصول کاری عمده را تولید می کند.

در حیطه مهندسی نرم افزار فرایند یک روش انطباق پذیر است که تیم نرم افزار به کمک آن می توانند مجموعه ای مناسب از کنش ها و وظایف کاری را برگزینند.

## ۵.۲. فعالیت های چارچوبی

توضیح	فعالیت
هدف ارتباطات درک اهداف طرف های ذینفع برای پروژه و جمع آوری خواسته هایی است که می توانند ویژگی ها و قابلیت های نرم افزار را تعیین کنند.	ارتباطات
با توصیف وظایف فنی که قرار است اجرا شوند، خطرات احتمالی، منابعی که مورد نیاز خواهد بود، محصولات کاری که باید تولید شوند و زمان بندی کاری، مهندسی نرم افزار را مشخص می کند.	برنامه ریزی
مهندسی نرم افزار با ایجاد مدل هایی جهت درک بهتر خواسته ها و طراحی که به این خواسته ها برسد همین کار را می کند.	مدل سازی
این فعالیت تولید کدها و آزمون لازم برای آشکار کردن خطاهای موجود در کدها را با هم تلفیق می کند.	ساخت
نرم افزار به مشتری تحویل داده شده که محصول تحویل داده شده را ارزیابی و براساس ارزیابی بازخوردی ارائه کند.	استقرار

## ۵.۳. فعالیت های چتری

این فعالیت ها در سرتاسر فرایند نرم افزار رخ می دهد و کانون توجه آن ها اساساً مدیریت پروژه، پیگیری و کنترل است.



## ۵.۴. تفاوت های مدل های فرایند

مدل های فرایند در ابعاد زیر با یکدیگر تفاوت دارند:

- جریان کلی فعالیت ها، کنش ها و وظایف و بستگی آن ها به یکدیگر
- درجه تعریف کنش ها و وظایف در هر فعالیت چارچوبی
- درجه شناسایی محصولات کاری و نیاز به آن ها
- شیوه اعمال فعالیت های تضمین کیفیت
- درجه کلی جزئیات به کار رفته در توصیف فرایند
- درجه دخالت مشتری و طرف های ذینفع در پروژه
- سطح استقلال داده شده به تیم نرم افزار
- درجه توصیف نقش ها و سازماندهی تیم

## ۵.۵. مدل فرایند تجویزی

- مدل فرایند تجویزی بر جزئیات تعریف، شناسایی و کاربرد فعالیت ها و وظایف تاکید دارد.
- هدف آن ها بهبود بخشیدن به کیفیت سیستم، بالا بردن قابلیت مدیریت پروژه، قابل پیش بینی کردن تاریخ تحویل و هزینه ها و راهنمایی تیم مهندسان نرم افزار را برای کارهای لازم است.

## ۵.۶. مدل فرایند چابک

- مدل فرایند چابک بر سرعت تاکید دارد و مجموعه ای از اصول را دنبال می کند که به یک روش غیر رسمی تر برای فرایند نرم افزار منجر می شود.
- این مدل بر قابلیت مانور و انطباق پذیر تاکید دارد و برای انواع بسیاری از پروژه ها به ویژه مهندسی برنامه کاربردی تحت وب مناسب است.

## ۶. مهندسی نرم افزار در عمل

### ۶.۱. اصول کلی در مهندسی نرم افزار

- اصول کلی که دیوید هوکر در مهندسی نرم افزار به عنوان یک کلیت است نام برد عبارتند از:

تفکر	برنامه ریزی پیشاپیش برای استفاده مجدد	آینده نگری	آنچه که شما تولید می کنید دیگران مصرف کنند.	حفظ چشم انداز	ساده نگه داشتن	دلیل وجود سیستم
------	--	------------	--	------------------	-------------------	--------------------

## ۷. پندارهای باطل نرم افزاری

### ۷.۱. پندارهای باطل مدیریتی

- ما از قبل کتابی داریم که آکنده از استانداردها و روال های لازم برای ساختن نرم افزارهاست. آیا این کتاب آن چه را که افراد من باید بدانند در اختیارشان قرار خواهد داد؟
- اگر از برنامه عقب بیفتیم، می توانیم بر تعداد برنامه نویسان بیفزاییم و عقب ماندگی را جبران کنیم (یورش مغولی)
- اگر تصمیم به برون سپاری یک پروژه نرم افزاری به شرکت دیگری بگیریم می توانم خودم را آسوده سازم و بگذارم تا آن شرکت آن را بسازد.

### ۷.۲. پندارهای باطل مشتریان

- بیانی کلی از اهداف، برای شروع به نوشتن برنامه ها کفایت می کند، جزئیات را بعدا می توانیم پر کنیم.
- نیازهای پروژه پیوسته در حال تغییر است، ولی این تغییرات را می توان در نرم افزار جای داد، زیرا نرم افزار انعطاف پذیر است.

### ۷.۳. پندارهای باطل سازندگان

- هنگامی که برنامه را نوشتیم و برنامه کار کرد، دیگر کار تمام است.
- تا هنگامی که برنامه را اجرا نکرده ام، راهی برای ارزیابی کیفیت آن ندارم.
- تنها چیز قابل تحویل برای یک پروژه موفق، برنامه ای است که کار کند.
- مهندسی نرم افزار، ما را وادار می سازد که مستندات حجیم و بیهوده تهیه کنیم و از سرعت ما می کاهد.