فصل اول

مقدمه

هادی سلیمی
دانشکده مهندسی کامپیوتر
دانشگاه علم و صنعت ایران
hsalimi@iust.ac.ir

اهداف فصل

در این فصل به موضوعات زیر پرداخته می شود:

- مفهوم نرم افزار و مهندسی نرم افزار
- فرآیند توسعه نرم افزار و متدولوژی های آن
- مدل های توسعه نرم افزار
- ابزار CASE
- اصول اخلاقی مهندسین نرم افزار

مقدمه (۱)

- نظریه مهندسی نرم افزار در سال ۱۹۶۸ در کنفرانسی تحت عنوان "**بحران نرم افزار**" مطرح شد.
- بحران نرم افزار (Software Crisis) ناشی از معرفی نسل سوم سخت افزار با قدرت فوق العاده اش بود. این قدرت منجر به این شد که تولید نرم افزارهای کاربردی که تا آن زمان غیر ممکن به نظر می رسیدند، امکان پذیر شوند.
- تجربیات اولیه در ساخت این سیستم های نرم افزاری نشان داد که **رهیافت غیر مهندسی** برای توسعه نرم افزار مناسب نمی باشد زیرا:
  - انجام پروژه های بزرگ سالها طول می کشید.
  - هزینه آنها بیش از هزینه پیش بینی شده بود.
  - قابل اعتماد نبودند.
  - نگهداری آنها دشوار و کارایی آنها اندک بود.

مهندسی نرم افزار ۲- سیستم های اجتماعی تکنیکی

مقدمه(۲)

- با توجه به این بحران، تکنیک های جدیدی نیاز بود تا پیچیدگی ناشی از سیستم های نرم افزاری بزرگ را کنترل کند.

از این تکنیک ها امروزه به عنوان **مهندسی نرم افزار** (Software Engineering) یاد می شود.

نرم افزار چیست؟

نرم افزار فقط برنامه های کامپیوتری نیست، بلکه تمام مستند سازی ها و داده های پیکربندی را شامل می شود که برای درست کار کردن این برنامه ها ضروری اند.

مهندسین نرم افزار با توسعه محصولات نرم افزاری سروکار دارند. دو نوع محصول نرم افزاری وجود دارد:

- محصولات کلی: سیستم های مستقلی که توسط یک سازمان تولید کننده ایجاد شده و به بازار عرضه می گردد. مانند واژه پردازها، بانک های اطلاعاتی
- محصولات سفارشی: سیستم هایی که توسط مشتری سفارش داده می شوند و توسط پیمانکاران ایجاد می شوند. مانند سیستم های کنترلی دستگاههای الکترونیکی، سیستم های کنترل ترافیک هوایی

مهندسی نرم افزار ۲- سیستم های اجتماعی تکنیکی

مهندسی نرم افزار چیست؟

مهندسی نرم افزار یک نظام مهندسی است که با تمام جنبه های نرم افزاری محصول، از مراحل اولیه تعیین مشخصات سیستم تا نگهداری سیستم سروکار دارد.

نکات اساسی طبق این تعریف، عبارتند از:

۱. **نظام مهندسی** : بکارگیری تئوریها، روشها و ابزارها در جای مناسب و در نظر گرفتن محدودیت های سازمانی و عملیاتی

۲. **تمام جنبه های محصول نرم افزاری**: مهندسی نرم افزار فقط با فرآیند های تکنیکی توسعه نرم افزار سروکار ندارد بلکه با فعالیت هایی مثل مدیریت پروژه نرم افزاری سرو کار دارد.

تفاوت مهندسی نرم افزار و علم کامپیوتر چیست؟

علم کامپیوتر با تئوری ها و روشهایی سروکار دارد که به کامپیوترها و سیستم های نرم افزاری مربوط می شود. در حالیکه مهندسی نرم افزار با مسائل عملی تولید نرم افزار سروکار دارد.

توجه: برخی از اطلاعات علم کامپیوتر برای مهندسی نرم افزار ضروری است.

مهندسی نرم افزار ۲- سیستم های اجتماعی تکنیکی

تفاوت مهندسی نرم افزار و مهندسی سیستم چیست؟

مهندسی سیستم با توسعه سیستم شامل توسعه نرم افزار، سخت افزار و طراحی فرآیند سروکار دارد اما مهندسی نرم افزار بخشی از مهندسی سیستم است که با توسعه زیر ساختار نرم افزاری، کنترل بانک های اطلاعاتی و برنامه های کاربردی سروکار دارد.

مهندسی سیستم نسبت به مهندسی نرم افزار قدیمی تر است. مثل مونتاژ سیستم های صنعتی پیچیده نظیر ترن ها

فرآیند توسعه نرم افزار چیست؟

فرآیند توسعه نرم افزار (Software Process) مجموعه ای از فعالیت ها و نتایج مربوط به آنهاست که یک محصول نرم افزاری را تولید می نماید.

چهار فعالیت اساسی در تمام فرآیندهای نرم افزار عبارتند از:

- تعیین مشخصات نرم افزار: وظایف نرم افزار و محدودیت های عملیات آن را تعریف می نماید.
- توسعه نرم افزار: تولید نرم افزاری با مشخصات تعیین شده
- اعتبار سنجی: تضمین اینکه آیا نرم افزارخواسته های مشتری را برآورده می نماید.
- تکامل نرم افزار: نرم افزار باید تکامل یابد تا نیازهای جدید کاربران را برآورده نماید.

توجه: فرآیند های مختلف نرم افزار، این فعالیت ها را به روشهای متفاوتی سازماندهی می نمایند، همچنین زمانبندی این فعالیت ها متفاوت خواهد بود.

مهندسی نرم افزار ۲- سیستم های اجتماعی تکنیکی

مدل فرآیند توسعه نرم افزار چیست؟

مدل فرآیند توسعه نرم افزار(Software Process Model)، توصیف ساده ای از فرآیند توسعه نرم افزار است که از چشم انداز خاصی بیان می شود.

نمونه هایی از چشم انداز (Perspective) که مدل فرآیند توسعه نرم افزار بر اساس آن بیان می شود، عبارتند از:

۱. چشم انداز جریان کار(Workflow perspective): دنباله ای از فعالیت های موجود در فرآیند را به همراه ورودی ها، خروجی ها و وابستگی آنها نشان می دهد.

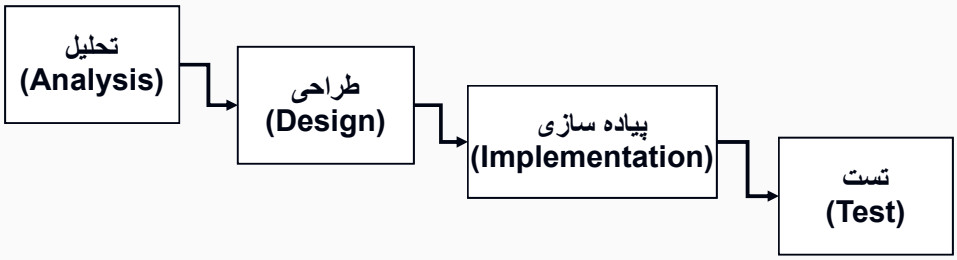
۲. چشم انداز جریان داده(Data-flow perspective): فرآیند را بصورت مجموعه ای از فعالیت ها نشان می دهد که هر کدام تغییراتی را در داده ها ایجاد می کند.

۳. چشم انداز نقش/فعالیت(Role/action perspective): نقشهای افراد موجود در فرآیند و فعالیت های تحت مسوولیت آنها را نشان می دهد.

مدل توسعه نرم افزار(۱)

مدل های مختلفی برای توسعه نرم افزار وجود دارد:

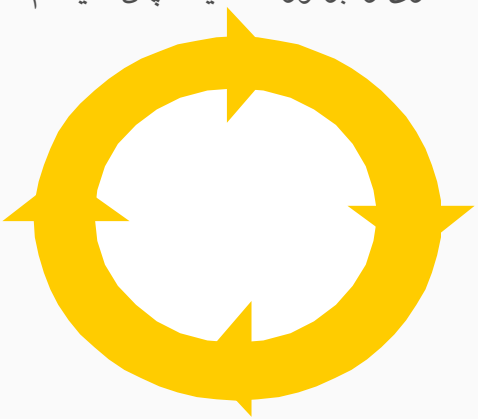
۱. رهیافت آبشاری(Waterfall): در این رهیافت فعالیت های مطرح شده بصورت تعیین و تحلیل مشخصات خواسته ها، طراحی نرم افزار، پیاده سازی و تست نمایش داده می شود. پس از تعریف هر مرحله، توسعه مرحله بعدی آغاز می گردد.



مهندسی نرم افزار ۲- سیستم های اجتماعی تکنیکی

مدل توسعه نرم افزار(۱)

۲. توسعه تکاملی(Iterative development): با استفاده از مشخصات انتزاعی، یک سیستم اولیه به سرعت ایجاد می شود. این سیستم با ورودی کاربر ترمیم می گردد تا سیستمی تولید گردد که نیازهای مشتری را برآورده نماید سپس سیستم تحویل داده می شود.



مدل توسعه نرم افزار(۲)

۳. تبدیل رسمی(Formal Transformation): در این رهیافت، مشخصات رسمی و ریاضی از سیستم تولید می گردد، آنگاه با استفاده از روشهای ریاضی این مشخصات به یک برنامه تبدیل می گردد. می توان تضمین نمود برنامه ای که توسعه یافته با مشخصات سازگار است.

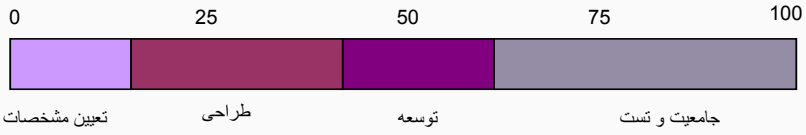
۴. روش مبتنی بر مولفه(Component-based method): این رهیافت فرض می کند بخش هایی از سیستم فعلا موجود است و از طریق مونتاژ مولفه های این بخش ها سیستم را توسعه می دهد.

مهندسی نرم افزار ۲- سیستم های اجتماعی تکنیکی

هزینه های مهندسی نرم افزار کدامند؟(۱)

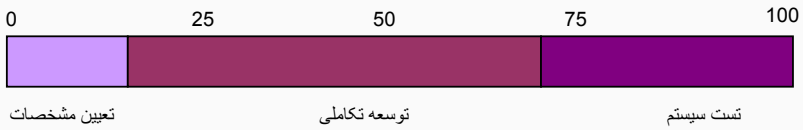
توزیع دقیق هزینه ها در فرآیند نرم افزار، به فرآیند مورد استفاده و نوع نرم افزاری بستگی دارد که در حال تولید است.

اگر کل هزینه توسعه یک سیستم نرم افزاری پیچیده را ۱۰۰ واحد در نظر بگیریم، توزیع این واحدهای هزینه بصورت شکل زیر خواهد بود:



هزینه های مهندسی نرم افزار کدامند؟(۲)

اگر نرم افزار به روش تکاملی توسعه یابد، مرز دقیقی بین تعیین مشخصات، طراحی و توسعه وجود ندارد. بنابراین توزیع این واحدهای هزینه بصورت شکل زیر خواهد بود:



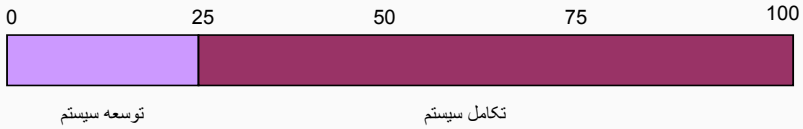
در این شکل هزینه های تعیین مشخصات کاهش یافته اند، زیرا قبل از توسعه، فقط مشخصات کلی سیستم ایجاد شده است.

مهندسی نرم افزار ۲- سیستم های اجتماعی تکنیکی

هزینه های مهندسی نرم افزار کدامند؟(۳)

علاوه بر هزینه های توسعه، هزینه های تغییر نرم افزار پس از بکارگیری وجود دارد. برای بسیاری از سیستم های نرم افزاری با طول عمر زیاد، این هزینه ها ممکن است با ۳ تا ۴ برابر افزایش یابد.

شکل زیر، توزیع این واحدهای هزینه را نشان می دهد.



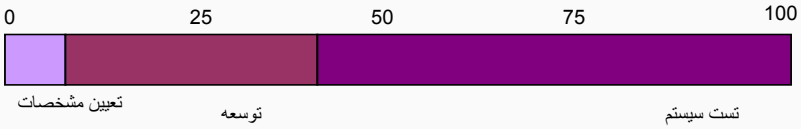
این شیوه توزیع هزینه، در آن دسته نرم افزارهای سفارشی انجام می شود که مشخصات آنها توسط مشتری تعیین می شود و پیمانکار آن را توسعه می دهد.

هزینه های مهندسی نرم افزار کدامند؟(۴)

محصولات نرم افزاری کامپیوترهای شخصی (PC)معمولا مشخصات کلی دارند و به روش توسعه تکاملی، توسعه می یابند. هزینه تعیین مشخصات تقریبا پایین است.

اما چون معمولا در پیکربندیهای مختلفی به کار گرفته می شود، باید بطور کامل تست شوند.

شکل زیر شرح هزینه این نوع محصولات را نشان می دهد.



متدولوژی های مهندسی نرم افزار کدامند؟

متدولوژی مهندسی نرم افزار یک رهیافت ساخت یافته برای توسعه نرم افزار است که نرم افزاری کیفی با هزینه مناسب را طراحی می نماید.

برخی انواع این متدولوژی ها عبارتند از
ساخت یافته (Structural)
عملکرد گرا (Function oriented)
شیء گرا (Object oriented)

این رهیافت های مختلف، امروزه در قالب یک رهیافت یکنواخت (Unified approach)جامعیت پیدا کرده اند که از **زبان مدل سازی یکنواخت** (UML) پیروی می کند.

توجه: هر یک از روش های مهندسی نرم افزار در زمینه خاصی بکار می آید. مثلا روش های شیء گرا برای سیستم های تعاملی(Interactive) مفیدند در حالیکه برای سیستم های بی درنگ(Real time) مناسب نیستند.

مولفه های روش های مهندسی نرم افزار

- اساس تمام روش ها، توسعه مدلهایی از سیستم است که ممکن است بصورت گرافیکی نمایش داده شوند.
- هر یک از این روش ها تعدادی مولفه را بصورت زیر در اختیار دارند:

مولفه	توصیف	مثال
توصیف های مدل سیستم	توصیفهای مدل سیستم و نشانه گذاری ها	مدل های شیء، مدل های داده، مدل های ماشین حالت
قواعد	محدودیت هایی که به سیستم اعمال می شوند	هر نهاد در یک مدل باید نام منحصر بفردی داشته باشد
پیشنهادات	تجربیاتی برای طراحی خوب	هیچ شیءای نباید بیش از هفت شیء فرعی داشته باشد
راهنمای فرآیند	توصیف هایی مربوط به فعالیت ها	قبل از تعریف عملیات مربوط به یک شیء، صفات آن تعریف گردد

مهندسی نرم افزار ۲- سیستم های اجتماعی تکنیکی

ابزار CASE چیست؟

مفهوم CASE (Computer-Aided Software Engineering)
مهندسی نرم افزار به کمک کامپیوتر می باشد.

CASE دامنه وسیعی از انواع مختلف برنامه ها را در بر می گیرد که برای پشتیبانی از فعالیت های نرم افزاری مثل تحلیل خواسته ها، مدل سازی سیستم و تست بکار گرفته می شود.

انواع ابزار کیس:

۱. ابزار کیس سطح بالا (Upper CASE): فرآیند نرم افزار را پشتیبانی می نمایند و فقط در تحلیل و طراحی بکار می روند.

۲. ابزار کیس سطح پایین (Lower CASE): برای پشتیبانی از پیاده سازی و تست طراحی شدند مثل عیب یابها، سیستم های تحلیل برنامه و ویراستارهای برنامه

صفات نرم افزار خوب کدامند؟

صفات نرم افزار، کیفیت آن نرم افزار را مشخص می کند. این صفات، رفتار نرم افزار را در حین اجرا، ساختار، کد برنامه و مستندات مربوطه مشخص می نماید. (نمونه ای از این صفات عبارتند از زمان پاسخ نرم افزار به تقاضای کاربر و قابلیت درک کد برنامه)

بسیاری از صفات به نوع کاربرد بستگی دارد، سیستم بانکداری باید امن باشد، سیستم Switching تلفن باید قابل اعتماد باشد.

این صفات بصورت زیر تعمیم داده شده است:

صفت محصول	توصیف
قابلیت نگهداری(Maintainability)	نرم افزار باید طوری نوشته شود که نیازهای جدید کاربر را برآورده کند.
قابلیت اتکا(Dependability)	نرم افزار قابل اتکا نباید هنگام خرابی سیستم، منجر به خرابی فیزیکی و خسارات مالی شود.
قابلیت استفاده (Usability)	نرم افزار باید برای کاربردی که تهیه شده است قابل استفاده باشد. یعنی واسط کاربر مناسب و مستندات کافی داشته باشد.

توجه: تکنیک هایی که در این درس بحث می شود، بر روی قابلیت اتکا و قابلیت نگهداری تاکید دارد.

مهندسی نرم افزار ۲- سیستم های اجتماعی تکنیکی

چالش های مهندسی نرم افزار کدامند؟

۱. **Legacy Challenge** : اغلب سیستم های نرم افزاری که امروزه در حال استفاده اند، چندین سال پیش ایجاد شده اند. این چالش، همان چالش نگهداری و بروز رسانی نرم افزارهاست.

۲. **Heterogeneity Challenge**: بسیاری از سیستم های امروزی بصورت سیستم های توزیعی در شبکه ها عمل می نمایند. این چالش، همان چالش توسعه تکنیک های ساخت نرم افزارهای قابل اتکا در برخورد با این ناهمگنی می باشد.

۳. **Delivery Challenge**: این چالش با تحویل سریع سیستم های بزرگ و پیچیده بدون کاهش کیفیت آنها سروکار دارد.

مسوولیت های تخصصی و اخلاقی

مهندسین نرم افزار باید مسوولیت های شغلی را که فراتر از مهارت های تکنیکی و کاری است، بپذیرند و نباید از مهارت ها و توانایی های خود سوء استفاده نمایند.

مواردی را که مهندسین نرم افزار باید به آن توجه نمایند عبارتند از:

- **محرمانگی(Confidentiality)**
- **صلاحیت(Competence)**
- **حقوق معنوی(Intellectual property rights)**
- **سوء استفاده کامپیوتر(Computer misuse)**

موسسات و سازمان های تخصصی زیادی، دستورالعمل های اخلاقی یا رفتاری را منتشر کرده اند.

در ادامه برخی از این دستورالعمل ها که توسط ACM و IEEE تهیه شده، آمده است:

مهندسی نرم افزار ۲- سیستم های اجتماعی تکنیکی

اصول اخلاقی ACM/IEEE

PUBLIC	مهندسین نرم افزار به نفع عموم کار می کنند.
CLIENT AND EMPLOYER	طوری عمل کنند که به نفع کارکنان و مشتریان باشد.
PRODUCT	محصولات و اصلاحات آنها بالاترین استاندارد را دارد.
JUDGMENT	جامعیت و استقلال را در قضاوت تخصصی خود حفظ کنند.
MANAGEMENT	مدیران و رهبران توسعه و نگهداری نرم افزار انجام می دهند.
PROFESSION	جامعیت و شهرت را مطابق با منافع عموم گسترش می دهند.
COLLEAGUES	حامی همکاران خود هستند و با آنها با عدالت برخورد می کنند.
SELF	سعی در آموزش بیشتر در حرفه خود دارند.

نکات کلیدی

- مهندسی نرم افزار یک نظام مهندسی است که با تمام جنبه های محصول نرم افزاری سروکار دارد.
- محصولات نرم افزاری شامل برنامه ها و مستندات اند.
- فعالیت های اساسی فرآیند توسعه نرم افزار شامل تعیین مشخصات سیستم، توسعه، اعتبار سنجی و تکامل می باشد.
- متدولوژی مهندسی نرم افزار، روش سازمان یافته برای تولید نرم افزار است.
- ابزار های Case، سیستم های نرم افزاری اند که برای پشتیبانی از فرایند توسعه نرم افزار بکار می رود.
- مهندسین نرم افزار فقط با اصول تکنیکی سرو کار ندارند بلکه دارای مسوولیت های اخلاقی نیز می باشند.

مهندسی نرم افزار ۲- سیستم های اجتماعی تکنیکی

پرسش های درس

پرسش ۱) تفاوت بین مدل فرآیند نرم افزار و فرآیند نرم افزار چیست؟

پرسش ۲) چرا هزینه های تست نرم افزار برای محصولات نرم افزاری عمومی، بسیار زیاد است؟