

مقدمه : تحلیل و طراحی سیستمهای نرم افزاری

تحلیل و طراحی سیستمها

دکتر قاسمی

دانشگاه آزاد تهران شمال

تعریف سیستم

❖ ما در جهانی زندگی میکنیم که پر از سیستم هاست. سیستم هایی که هدف های مختلف و مشخصی دارند. بذارین یکم برگردیم و اصلا ببینیم سیستم یعنی چی؟

❖ سیستم System

- مجموعه ای Set از اجزاء element و رابطه های relation بین آنها.
- (G.Klir , System Theory)

❖ White box : جعبه ی سفید یعنی شکستن یک سیستم به اجزاء و رابطه ها (به این میگویند تحلیل سیستم-
system analysis یا system decomposition

❖ Black box : جعبه ی سیاه یعنی سیستمی که نمی توان اجزای آن را مشاهده نمود. برای شناخت چنین سیستم
هایی باید از طریق رفتارها شناسایی شوند.

اهمیت تحلیل و طراحی سیستم ها

- ❖ گسترش استفاده از سیستم در علوم مختلف سبب شد تا لزوم شناخت و تجزیه و تحلیل سیستم‌های موجود، مطرح گردد و با توجه به ویژگی‌های نگرش سیستمی، فرایندگرایی و کلی‌نگری به عنوان رویکردی اثربخش ارائه شود.
- ❖ از طرفی، ناکارآمدی سیستم‌های موجود، جریان فکری ایجادشده را به سوی طراحی سیستم‌های جدید و مطلوب سوق داد که نتیجه آن، یک تجزیه و تحلیل اثربخش در سیستم‌های موجود است، همچنین با ورود به عصر اطلاعات، طراحی سیستم‌هایی سریع و اثربخش در انتقال اطلاعات، ضرورت یادگیری فنون تجزیه و تحلیل و طراحی سیستم را افزایش داده است.

❖ مهندسی نرم افزار Software engineering یعنی استفاده از اصول مهندسی بجا و مناسب برای تولید و ارائه محصول نرم افزاری با کیفیت که قابل اطمینان و با صرفه بوده و بر روی ماشین های واقعی به طور کارآمدی عمل کند.

❖ مهندسی نرم افزار یک روش سیستماتیک، منظم و دقیق برای ساخت و ارائه محصولی نرم افزاری با کیفیت است.

❖ مهندسی نرم افزار اغلب شامل فرایند خطی تحلیل، طراحی، پیاده سازی و آزمون است؛ که با به کارگیری روش های فنی و علمی از علوم مهندسی موجب تولید نرم افزاری با کیفیت مطلوب در طول یک فرایند انتخابی مناسب پروژه می شود.

نیاز به مهندسی نرم افزار

- ❖ نرم افزار عموماً از محصولات و موقعیت‌هایی شناخته می‌شود که قابلیت اطمینان زیادی از آن انتظار می‌رود،
- ❖ حتی در شرایط طاقت فرسا، مانند نظارت و کنترل نیروگاه‌های انرژی هسته‌ای، یا هدایت یک هواپیمای مسافربری در هوا،
- ❖ چنین برنامه‌هایی شامل هزاران خط کد هستند، که از نظر پیچیدگی با پیچیده‌ترین ماشین‌های نوین قابل مقایسه هستند.
- به عنوان مثال، یک هواپیمای مسافربری چند میلیون قطعه فیزیکی دارد (و یک شاتل فضایی حدود ده میلیون بخش دارد)، در حالی که نرم افزار هدایت چنین هواپیمایی می‌تواند تا ۴ میلیون خط کد داشته باشد.
- ❖ با توجه به گسترش روزافزون دنیای رایانه امروزه بیش از هر زمان دیگری نیاز به مهندسی نرم افزار احساس می‌شود.
- ❖ مهم‌ترین شاخص در مهندسی نرم افزار : تحلیل و طراحی و پیاده‌سازی و استقرار نرم افزار با کیفیت متناسب با «نیازهای مشتری» است.

❖ مهندسی سیستم ها یک رویکرد میان رشته ای است که امکان تحقق سیستم های موفق را فراهم می سازد. مهندسی سیستم ها بر تعیین نیازهای مشتری و کارکردهای لازم در ابتدای چرخه توسعه و مستند کردن الزامات تمرکز می کند و با سنتز طراحی و اعتباردهی سیستم ادامه پیدا می کند، در حالی که همواره تمامی مساله را در نظر دارد:

❖ مهندسی سیستم ها تمامی رشته های علمی و گروه های تخصصی را برای یک کار تیمی یکپارچه می سازد تا فرآیند توسعه ساختار یافته ای را از Concept ها تا ساخت و عملیات شکل دهد. مهندسی سیستم ها هم نیازهای کسب و کار و هم نیازهای فنی تمامی مشتریان را در نظر می گیرد تا بتواند یک محصول باکیفیت فراهم کند و نیازهای کاربر را محقق سازد.

توسعه نرم افزار (۱ از ۳)

❖ برنامه کامپیوتری (برنامه)

- یک برنامه نرم افزاری کامپیوتری که بر روی یک دستگاه محاسباتی اجرا می شود تا مجموعه خاصی از عملکردها را انجام دهد.
- دامنه متوسط

❖ سیستم اطلاعاتی

- مجموعه ای از اجزای مرتبط به هم که اطلاعات مورد نیاز برای تکمیل وظایف تجاری را جمع آوری، پردازش، ذخیره و به عنوان خروجی ارائه می کند.
- دامنه گسترده تر از "برنامه"
- شامل پایگاه داده و فرآیندهای دستی مرتبط است

توسعه نرم افزار (۲ از ۳)

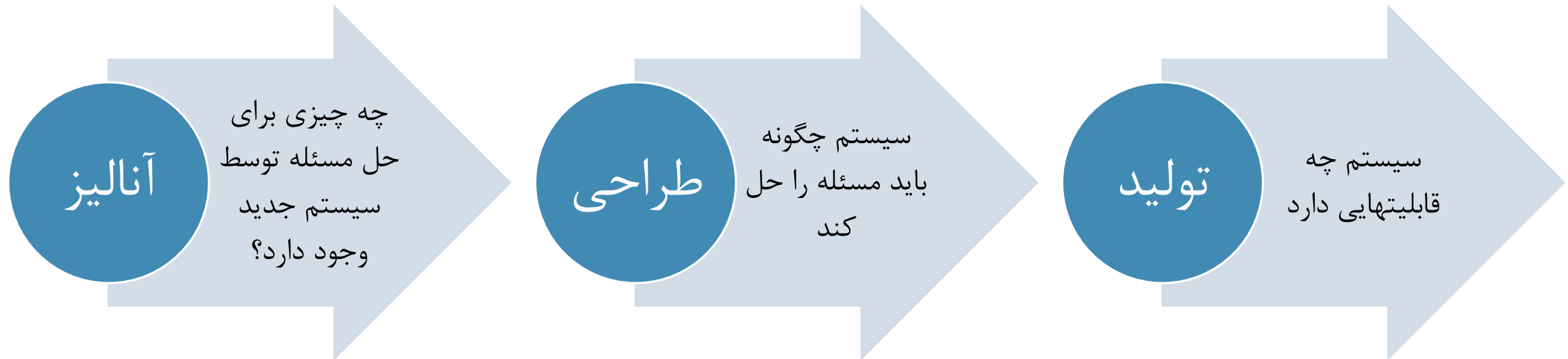
❖ تجزیه و تحلیل سیستم ها

- آن دسته از فعالیت هایی که فرد را قادر می سازد تا بفهمد و مشخص کند که یک سیستم اطلاعاتی باید چه کاری انجام دهد

❖ طراحی سیستم ها

- آن دسته از فعالیت هایی که فرد را قادر می سازد تا سیستمی را که نیاز را برطرف می کند تعریف و توصیف کند

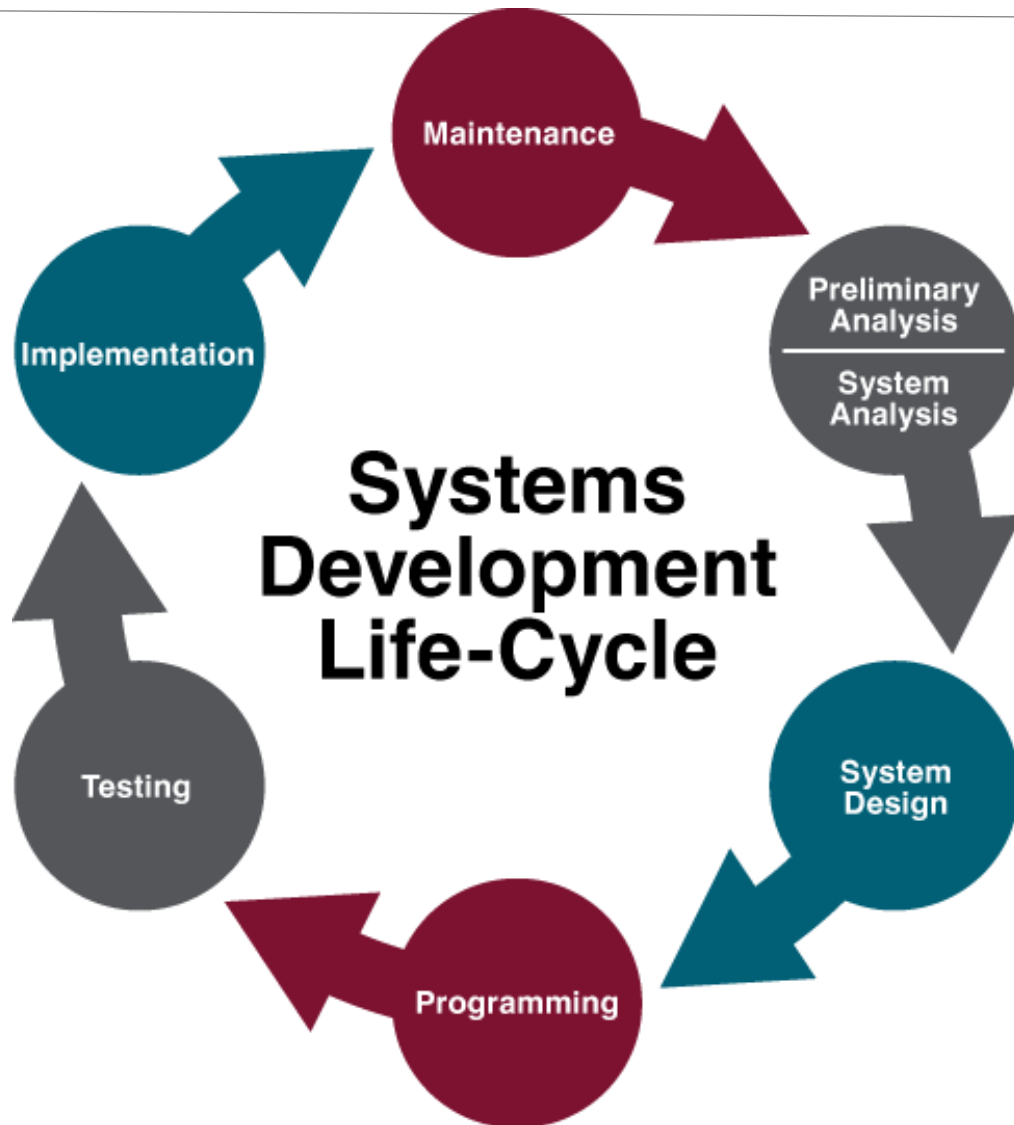
توسعه نرم افزار (۲ از ۳)



توسعه نرم افزار (۳ از ۳)

- ❖ نیاز را درک کنید (نیاز کسب و کار)
- ❖ چشم انداز را تعیین کنید
- ❖ راه حل را تعریف کنید
- ❖ چشم انداز و راه حل را به اشتراک بگذارید
- ❖ راه حل را بسازید Solution Making
- ❖ تأیید کنید که راه حل نیاز را برآورده می کند
- ❖ سیستم Solution را راه اندازی کنید

System Development Life Cycle (SDLC) (1 of 3)



❖ فرآیندی که شامل تمام فعالیت های مورد نیاز برای ساخت، راه اندازی و نگهداری یک سیستم اطلاعاتی است. شش فرآیند اصلی عبارتند از:

- مشکل یا نیاز را شناسایی کنید و تاییدیه بگیرید
- برنامه ریزی و نظارت بر پروژه انجام دهید
- جزئیات مشکل یا نیاز را کشف و درک کنید
- اجزای سیستم را طراحی کنید که مشکل را حل کند
- ساخت، تست و ادغام اجزای سیستم را تکمیل کنید
- تست های سیستم را کامل کنید و سپس راه حل را مستقر کنید

System Development Life Cycle (SDLC) (2 of 3)

❖ پروژه

- یک تعهد برنامه ریزی شده که شروع و پایانی دارد و نتیجه مشخصی را ایجاد می کند
- برای توسعه یک سیستم اطلاعاتی استفاده می شود
- نیاز به دانش تجزیه و تحلیل سیستم ها و ابزارها و تکنیک های طراحی سیستم ها دارد



System Development Life Cycle (SDLC) (2 of 3)



PMBOK 

- محدوده - Scope
- زمان - Time
- هزینه - Cost
- کیفیت - Quality
- یکپارچگی - Integration
- منابع انسانی - Human Resource
- ارتباطات - Communications
- ریسک - Risk
- تدارکات - Procurement



System Development Life Cycle (SDLC) (3 of 3)

❖ فرآیند توسعه سیستم

- رویکرد واقعی مورد استفاده برای توسعه یک سیستم اطلاعاتی خاص (معروف به روش شناسی)
- فرآیند یکپارچه Unified process (UP)
- برنامه نویسی شدید یا سریع Extreme programming (XP)
- اسکرام Scrum
- اکثر فرآیندها/روش شناسی ها اکنون از توسعه چابک و تکراری استفاده می کنند Agile

Iterative Development

❖ توسعه چابک Agile development

- فرآیند توسعه سیستم اطلاعاتی که بر انعطاف پذیری برای پیش بینی الزامات جدید در طول توسعه تأکید دارد
- سریع روی پا؛ پاسخگو به تغییر

❖ توسعه تکراری Iterative development

- رویکردی به توسعه سیستم که در آن سیستم تکه به تکه از طریق تکرارهای متعدد "رشد" می شود.
- بخش کوچکی از سیستم را کامل کنید (پروژه کوچک)، سپس فرآیندها را برای اصلاح و اضافه کردن موارد دیگر تکرار کنید، سپس برای اصلاح و اضافه کردن بیشتر تکرار کنید تا زمانی که انجام شود

Iterative and Agile Systems Development Lifecycle (SDLC)

