



اصول و مفاهیم تحلیل

درس مهندسی نرم افزار ۲

مفاهیم کلیدی

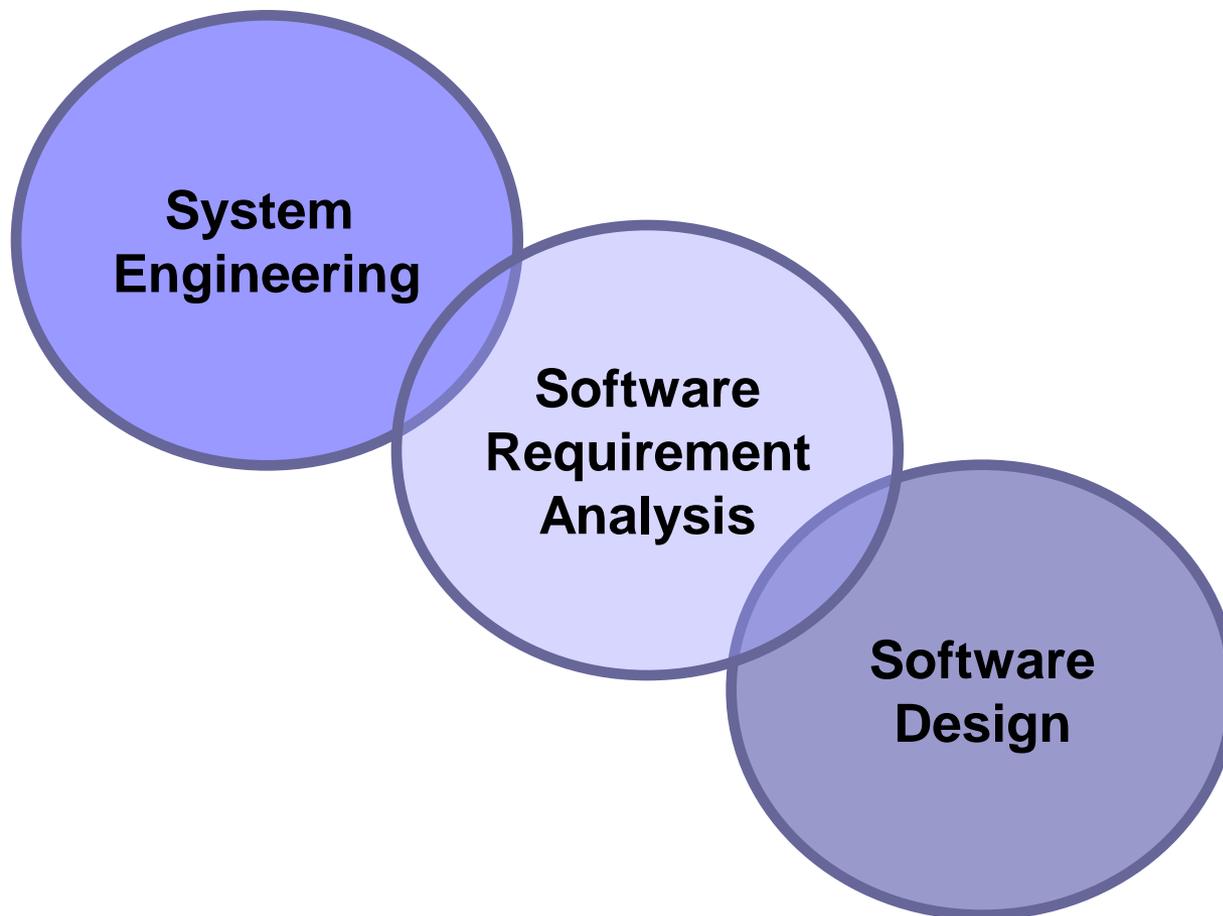
- مشکل درک خواسته‌های مشتریان
- تحلیل نیازمندی‌های نرم‌افزار
- فعالیت‌های تحلیل نرم‌افزار
- استقرار تابع کیفیت (*Quality Function Deployment*)
- اصول مشترک در روش‌های تحلیل
- نمونه‌سازی
- بیان خصوصیات

مشکل درک خواسته‌ها

- مشتریان ایده‌های مبهمی از چیزی که نیازهایشان است، دارند
- توسعه دهنده با این امید که می‌تواند جزئیات بیشتری از «ایده مبهم» بدست آورد، شروع می‌کند
- با افزایش درک مشتری، ایده‌های او تغییر می‌کنند
 - این تغییرات می‌توانند بسیار گسترده و یا متضاد باشند
- توسعه دهنده سعی به در نظر گرفتن این تغییرات می‌کند
- خصوصیات اولیه و توسعه دچار تغییر می‌شود و از همین جا خطاها شروع می‌شوند

این روند در توسعه نرم افزار بسیار طبیعی است!

تحلیل نیازمندی‌های نرم‌افزار



تحلیل نیازمندی‌های نرم‌افزار (ادامه)

■ تحلیل نیازمندی‌ها

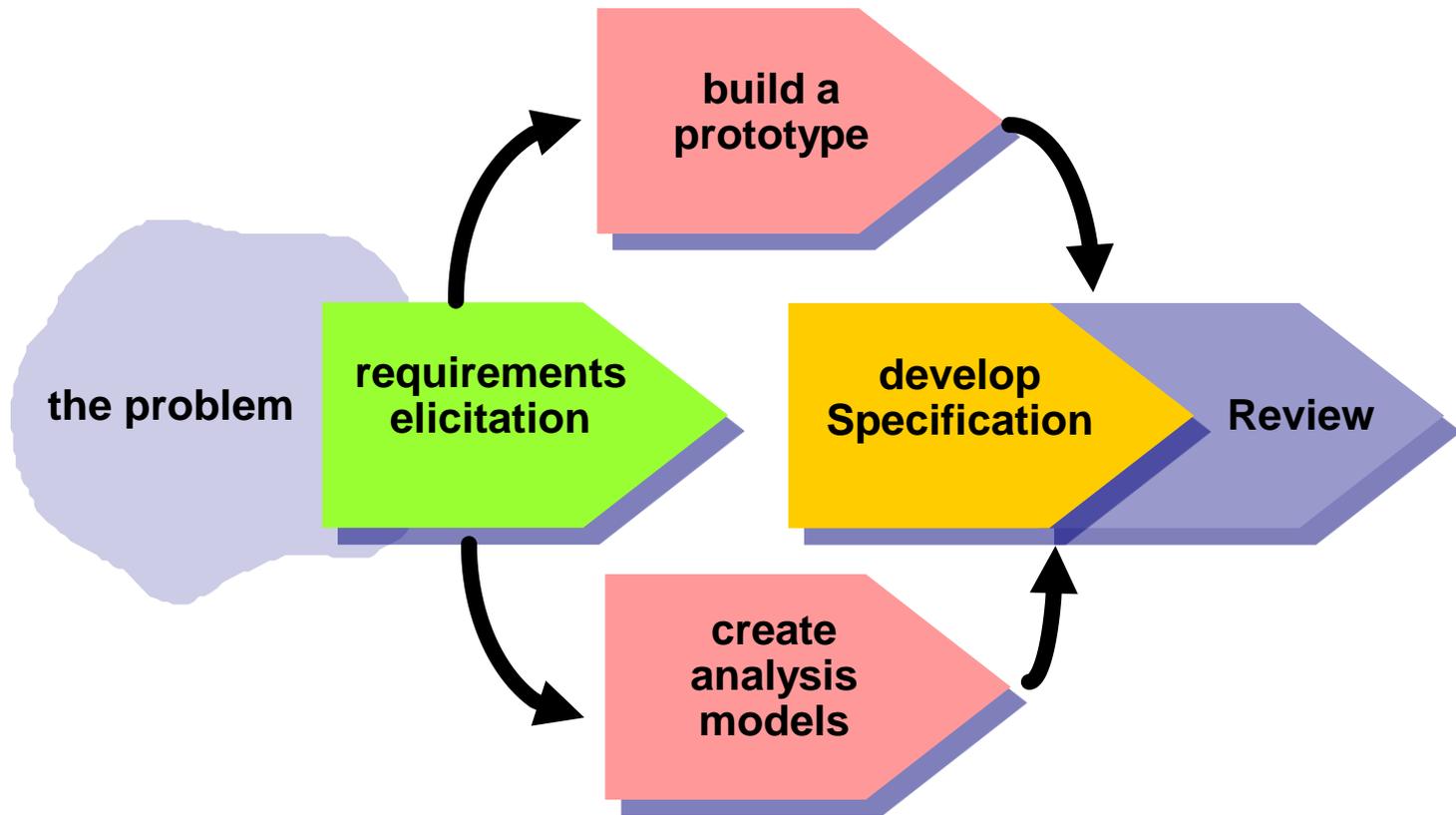
- خصوصیات عملیاتی نرم‌افزار مشخص شود
 - از نظر داده، وظیفه و رفتار
- ارتباط بین نرم‌افزار و سایر عناصر سیستم مشخص شود
- مدل‌های مورد نیاز نرم‌افزار، انتخاب شوند و مدلسازی داده‌ها،
وظایف و رفتار انجام شود
- مدل‌های طراحی با استفاده از نیازمندی‌ها شکل می‌گیرند
- بازبینی‌های فنی رسمی نرم‌افزار تولید شوند

فعالیت‌های تحلیل نرم افزار



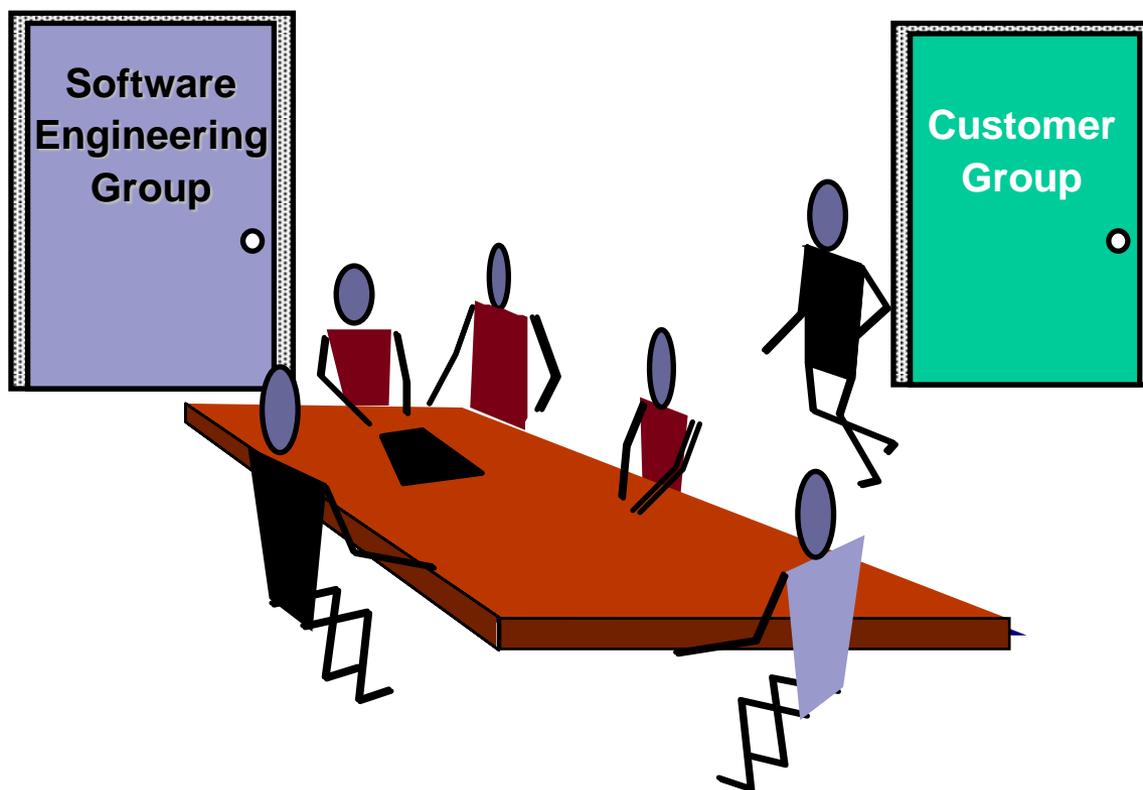
- شناخت مسئله (*Problem Recognition*)
- ارزیابی و سنتز (*Synthesis*)
- مدل سازی
- تعیین خصوصیات
- بازبینی

فعالیت‌های تحلیل نرم‌افزار (ادامه)



شناخت مسئله

- استفاده از روش‌هایی چون پرسش‌نامه، مصاحبه و ...



(Facilitated Application Specification Techniques) FAST

edu.mandegar@yahoo.com

دکتر عزیزاله محبی گرگری

استقرار تابع کیفیت (Quality Function Deployment)

- يك تكنيك مدیریت کیفیت است که نیازهای مشتری را به نیازمندی های فنی نرم افزار ترجمه می کند
- QFD بر حداکثرسازی رضایت مشتری از فرآیند مهندسی نرم افزار تمرکز دارد
- به منظور دستیابی به این هدف، QFD بر درک چیزهای با ارزش برای مشتری تاکید دارد
- چیزهای با ارزش برای مشتری در سراسر فرآیند مهندسی گسترش یافته و بکار گرفته می شوند

استقرار تابع کیفیت (ادامه)

■ سه نوع نیازمندی ها

● نیازمندی های نرمال (*Normal Requirements*)

- اهداف و مقاصد یک محصول یا سیستم که در جلسه با مشتری بیان می شوند

● نیازمندی های مورد انتظار (*Expected Requirements*)

- این نیازمندی ها برای محصول یا سیستم ضمنی هستند و ممکن است آن چنان اساسی باشند که مشتری به صراحت آنها را بیان نکند
- فقدان آنها می تواند باعث نارضایتی فراوانی می شود

● نیازمندی های مهیج (*Exciting Requirements*)

- این جنبه ها فراتر از انتظارات مشتری هستند و در صورت وجود، رضایت زیادی را به دنبال دارند

استقرار تابع کیفیت (ادامه)

Function Deployment ■

- تعیین «ارزش» هر وظیفه مورد نیاز برای سیستم

Information Deployment ■

- تعیین موضوعات داده‌ای و رویدادها

Task Deployment ■

- شناسایی رفتار سیستم

Value Analysis ■

- تعیین اولویت‌های نسبی نیازمندی‌ها



مورد کاربری (Use-Case)

■ ترتیبی از گام‌های مرتبط به منظور تکمیل يك وظیفه حرفه

- هر مورد کاربری مجموعه‌ای از سناریوهاست که نحوه استفاده از سیستم را نمایش می‌دهند

■ هر سناریو به موارد زیر را پاسخ می‌دهد

- وظایف اصلی کنشگر
- اطلاعاتی که کنشگر دریافت، تولید و یا تغییر می‌دهد
- اطلاعات مورد نیاز کنشگر

نمونه مورد کاربری سطح بالا

نام مورد کاربری	سفارش جدید
کنشگرها	عضو
شرح	این مورد کاربری فرآیند درخواست سفارش يك عضو براي محصول خاص را بیان می کند. در صورت تکمیل سفارش، برای عضو پیامی مبنی بر اینکه سفارش پذیرفته شده است، ارسال می شود.

اصول مشترک در روش‌های تحلیل

- محدوده اطلاعات مساله باید درک و نمایش داده شود
- وظایفی که نرم‌افزار باید اجرا کند باید تعریف شود
- رفتار نرم‌افزار (در نتیجه وقایع خارجی) باید نمایش داده شود
- مدل‌هایی که حاوی اطلاعات، وظایف و رفتار هستند باید به گونه‌ای افراز شوند تا جزئیات را نمایش دهند
 - با استفاده از روش لایه‌ای یا سلسله مراتبی
- فرآیند تحلیل باید از اطلاعات پایه‌ای به سمت جزئیات پیاده‌سازی حرکت کند

اصول Davis برای مهندسی نیازمندی‌ها

■ درک مسئله قبل از شروع به ساخت مدل تحلیل

■ توسعه نمونه برای درک بهتر مشتری

■ ثبت منشاء و دلیل هر نیازمندی

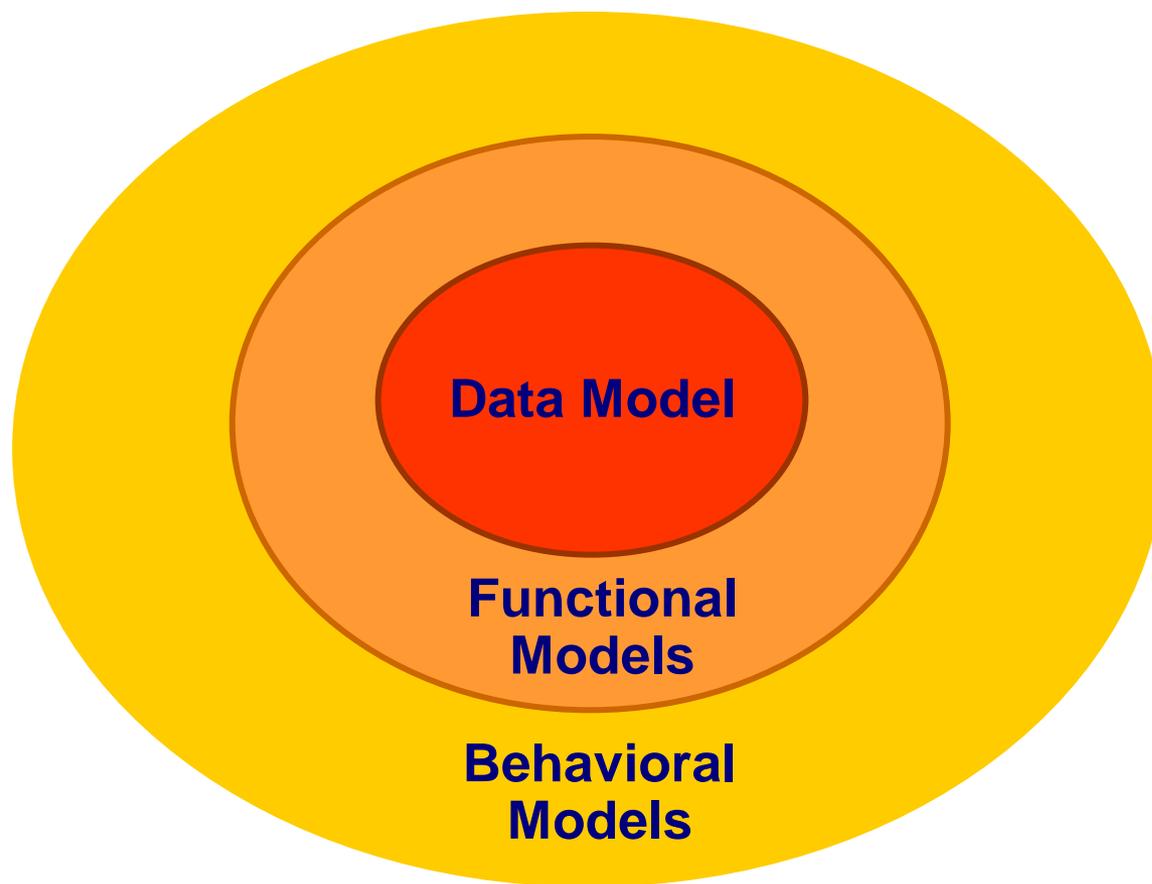
■ استفاده از چندین دیدگاه برای نیازمندی‌ها

■ اولویت‌بندی نیازمندی‌ها

■ تلاش برای حذف ابهامات



مدل‌های تحلیل



مدل‌های تحلیل (ادامه)

■ مدل‌سازی داده

- تعریف موضوعات داده‌ای
- تشریح صفات داده
- برقراری ارتباط بین داده‌ها

■ مدل‌سازی وظایف

- تعیین وظایفی که موضوعات داده‌ای را انتقال می‌دهند
- مشخص نمودن جریان داده‌ها در سیستم
- نمایش تولیدکنندگان و مصرف‌کنندگان داده

مدل‌های تحلیل (ادامه)

■ مدل‌سازی رفتار

- مشخص نمودن حالات متفاوت سیستم
- تعیین رویدادهایی که سبب تغییر وضعیت سیستم می‌شوند



نمونه‌سازی نرم افزار (*Prototyping*)

■ انواع نمونه

• نمونه دور انداختنی (*Throwaway prototyping*)

- نمونه تنها برای نمایش برخی قابلیت‌ها مورد استفاده قرار می‌گیرد
- بعد از نمایش و بدست آوردن نتایج، کنار گذاشته می‌شود

• نمونه تکاملی (*Evolutionary prototyping*)

- نمونه به عنوان بخشی از مرحله شناخت مورد استفاده قرار می‌گیرد
- در مراحل بعدی پروژه (طراحی و ساخت) توسعه داده می‌شود



نمونه‌سازی نرم‌افزار (ادامه)

■ روش‌ها و ابزارهای توسعه نمونه

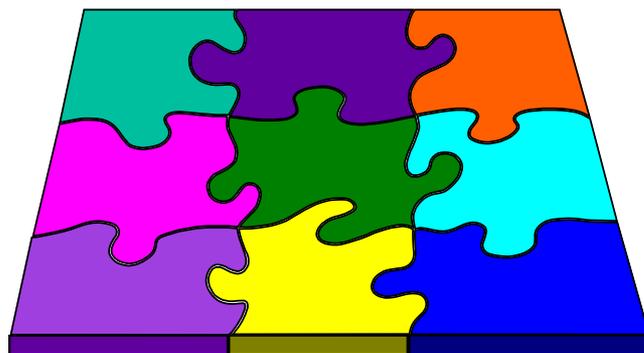
• تکنیک‌های نسل چهارم

□ پایگاه داده‌ها، زبان‌ها و ابزارهای سطح بالا

• استفاده از مولفه‌های نرم‌افزاری قابل استفاده مجدد

□ کتابخانه‌های از مولفه‌های قابل استفاده مجدد

• زبان‌ها و ابزارهای بیان رسمی خصوصیات و محیط‌های نمونه‌سازی



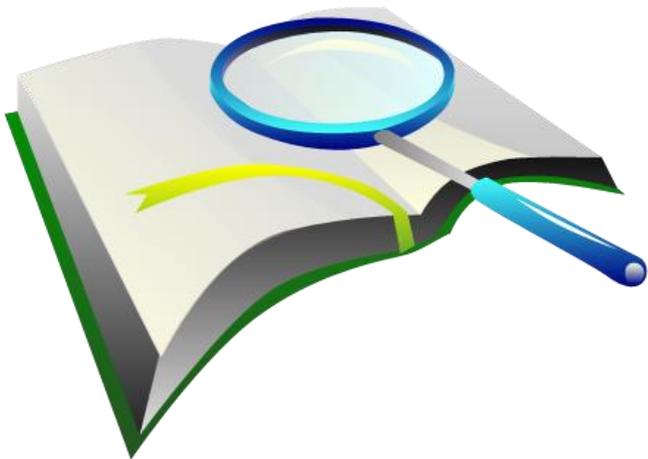
تعیین خصوصیات

■ مفاهیم و اصول تعیین خصوصیات (*Balzer and Goodman*)

- جداسازی وظیفه‌مندی از پیاده‌سازی
- توسعه مدل مناسبی از رفتار سیستم که حاوی داده و وظیفه باشد
- مشخص نمودن فضایی که نرم‌افزار در آن اجرا خواهد شد
 - با تعیین ارتباط سیستم با سایر سیستم‌ها
- تعیین محیطی که نرم‌افزار در آن اجرا می‌شود
- ایجاد مدل شناخت سیستم (*cognitive model*)
 - فراتر از مدل طراحی یا پیاده‌سازی
- تشخیص میزان ناکامل بودن خصوصیات
- مشخص نمودن متن و ساختار خصوصیات که توانایی تغییر داشته باشد

بازبینی خصوصیات

- ابتدا در سطح ماکروسکوپی (*macroscopic*) انجام می‌پذیرد
 - تعیین اینکه خصوصیات، کامل، سازگار و صحیح هستند
- بازبینی جزئیات تمام حوزه‌های اطلاعاتی، وظیفه‌ای و رفتاری خصوصیات را مورد توجه قرار می‌دهد



پایان