

بسمه تعالی

درس مهندسی نرم افزار پیشرفته

فصل اول

بحران نرم افزار و بررسی پیچیدگی نرم افزار

دکتر فریدون شمس

اهداف جلسه



- درک مشکلات تولید نرم افزار و راه مقابله با آن
- درک اهمیت مهندسی نرم افزار
- درک ضرورت روی آوردن به یک متدولوژی مدون
- درک ماهیت نرم افزار و تفاوت آن با بقیه محصولات مهندسی
- درک پیچیدگی نرم افزار و عوامل پدید آورنده آن
- آشنائی با مبحث سیستم های پیچیده و ویژگیهای آنها

فهرست مطالب



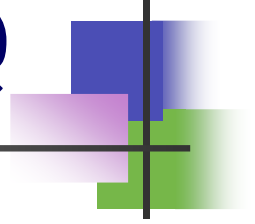
- بحران نرم افزار
- مهندسی نرم افزار و ضرورت آن
- متدولوژی و اهمیت آن
- تفاوت نرم افزار و سخت افزار
- پیچیدگی ذاتی نرم افزار
- عوامل پدیدآورنده پیچیدگی نرم افزار
- ساختار و ویژگی سیستم های پیچیده
- پیچیدگی سازمان یافته و سازمان نیافته

بحران نرم افزار



“پیشرفت شگرف سخت افزار و ضعف روش های توسعه
نرم افزار و ناتوانی این روش ها در کنترل پیچیدگی
نرم افزار بحران نرم افزار را بوجود آورد”

بحران نرم افزار (ادامه)



■ علایم این بحران عبارتند از:

- عدم بهره‌گیری کامل از قدرت سخت‌افزار
- ناتوانی روش‌های تولید نرم‌افزار در پاسخگویی به افزایش تقاضا
- هزینه‌های هنگفت تولید نرم‌افزار
- عدم تحویل به موقع
- عدم تامین نیازمندی‌های کاربر
- کیفیت پایین و نامطمئن
- سختی نگهداری بعلت کیفیت پایین طراحی

مهندسی نرم افزار و ضرورت آن



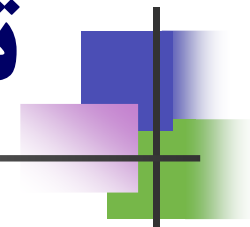
“برای مقابله با این بحران **مهندسی نرم افزار** مطرح شد“

ویژگیهای مهندسی نرم افزار

- تجدید نظر در روش های برنامه ریزی و کنترل
- استفاده از تجربیات دیگر رشته های مهندسی (مثال الگوها)
- تهیه و جایگزینی استانداردهای مطمئن

design patterns

تعریف مهندسی نرم افزار



«بکارگیری یک روش سیستماتیک، منظم و قابل اندازه گیری
برای تولید و توسعه، عملیاتی کردن و نگهداری نرم افزار؛
بکارگیری اصول مهندسی در تولید نرم افزار»

codingextend (debug, new feature)

اهداف مهندسی نرم افزار



- افزایش کیفیت، قابلیت اطمینان، قابلیت نگهداری
- رضایت کاربر و سهامداران
- کاهش هزینه
- تحویل به موقع
- استفاده از مولفه‌های استاندارد
- استفاده مجدد

module public methods of component = interface

اصول مهندسی نرم افزار



اهمیت متدولوژی

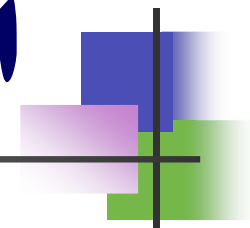


«یکی از علل اساسی بحران نرمافزار عدم وجود روش‌های
مناسبی برای تولید نرمافزار»

■ روش (*Method*)

■ فرآیندی منظم که با استفاده از مجموعه‌ای از نمادگذاری‌های
خوش‌تعریف، مجموعه‌ای از مدل‌ها را ایجاد می‌کند که هر کدام بخشی
از سیستم نرم‌افزاری در دست تولید (یا توسعه) را توصیف می‌نمایند

اهمیت متدولوژی (ادامه)



■ متدولوژی

■ مجموعه‌ای از روش‌ها که در تمام چرخه حیات سیستم نرم‌افزاری اعمال شده و بر یک نوع نگرش کلی درباره جهان نرم‌افزار متکی هستند

بنظر بسیاری از متخصصین بکارگیری یک **متدولوژی مدون** در توسعه نرم‌افزار می‌تواند تا اندازه قابل توجهی مشکلات بیان شده را برطرف سازد

ویژگیهای یک متدولوژی مطلوب

ex: define class, method, etc in OOP

■ ارائه تعاریفی از مفاهیم اولیه بکار رفته در متدولوژی

■ ارائه مدلی برای فرآیند تولید

process modellike R

ex: client serve

■ داشتن مدل زیربنایی (مدل معماری)

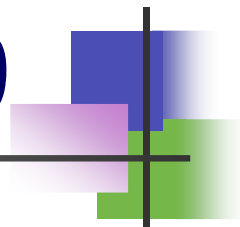
UML

■ ارائه یک شیوه علامت‌گذاری استاندارد

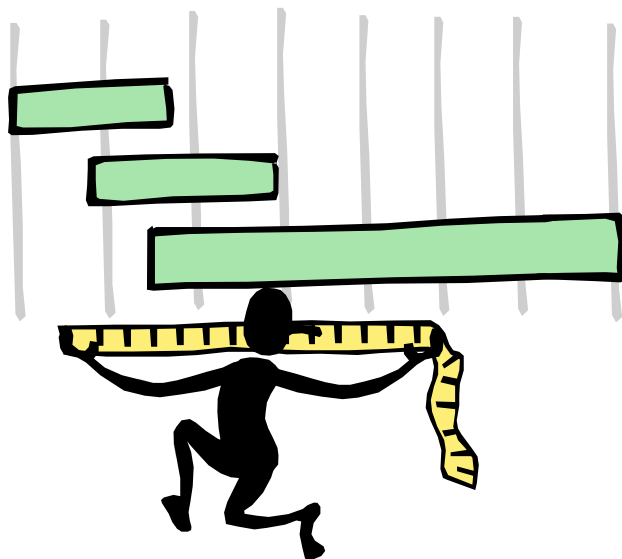
■ معرفی تکنیک‌هایی برای پیاده‌سازی متدولوژی که توانایی

کنترل پیچیدگی سیستم‌های کنونی را دارا باشند

ویژگیهای یک متدولوژی ... (ادامه)



- ارائه معیارهای برای ارزیابی نتایج ^{artifacts} حاصل از بکارگیری متدولوژی
- وجود ابزار اتوماتیک برای کمک به تولید و اجرای مدل‌های مبتنی بر متدولوژی



cohesion vs coupling for modules (components) number of lines

تفاوت نرم افزار و سخت افزار



فرآیند تولید نرم افزار یک فرآیند مهندسی است نه



یک فرآیند تولید صنعتی

نرم افزار

سخت افزار

بوسیله استفاده کنندگان نهایی

بوسیله متخصصین

تعیین مشخصات

فرآیند مهندسی که برای هر کاربرد جدید منحصر به فرد است

ماهیت فرآیند تولید مکانیکی

منطقی

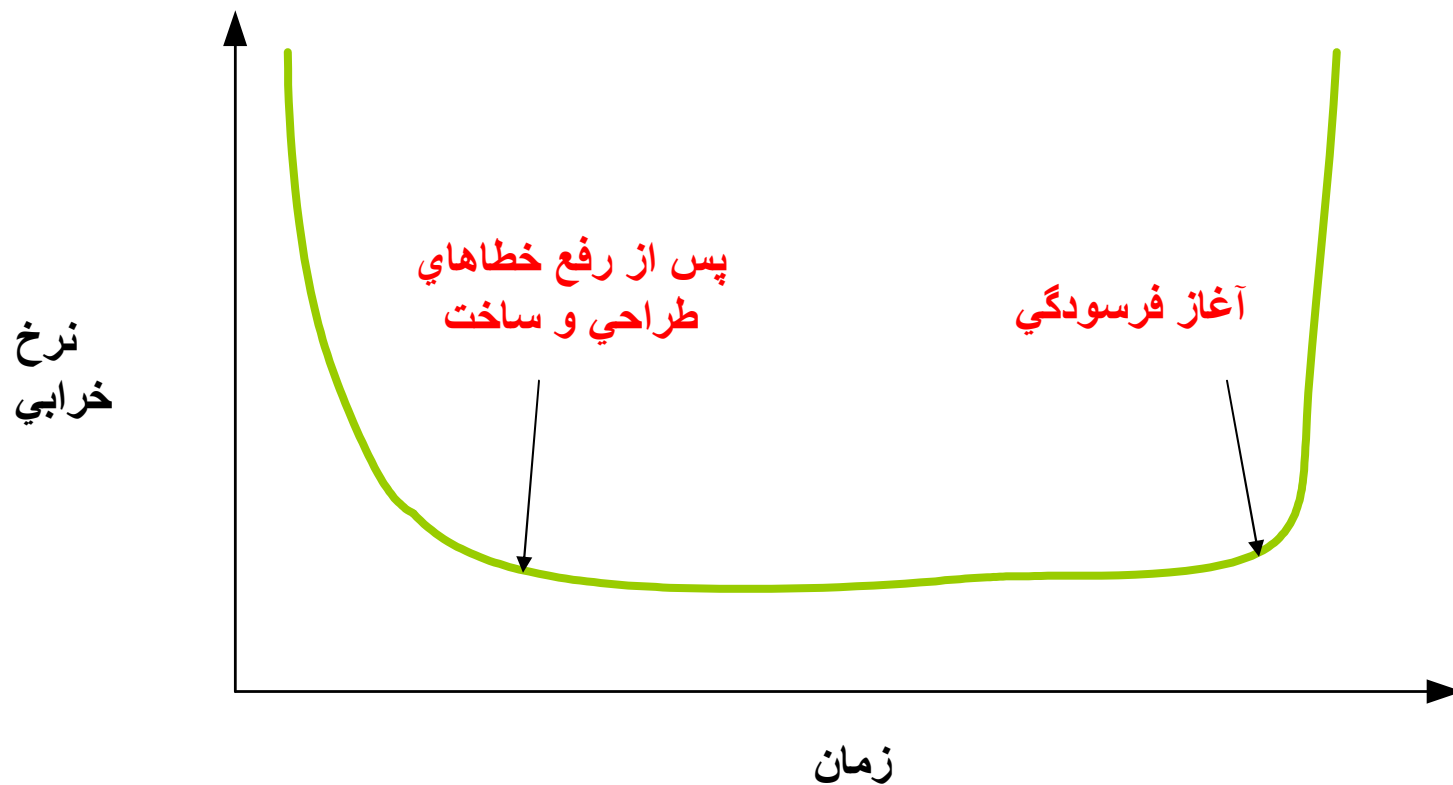
فیزیکی

ماهیت محصول

تفاوت نرم افزار و سخت افزار (ادامه)

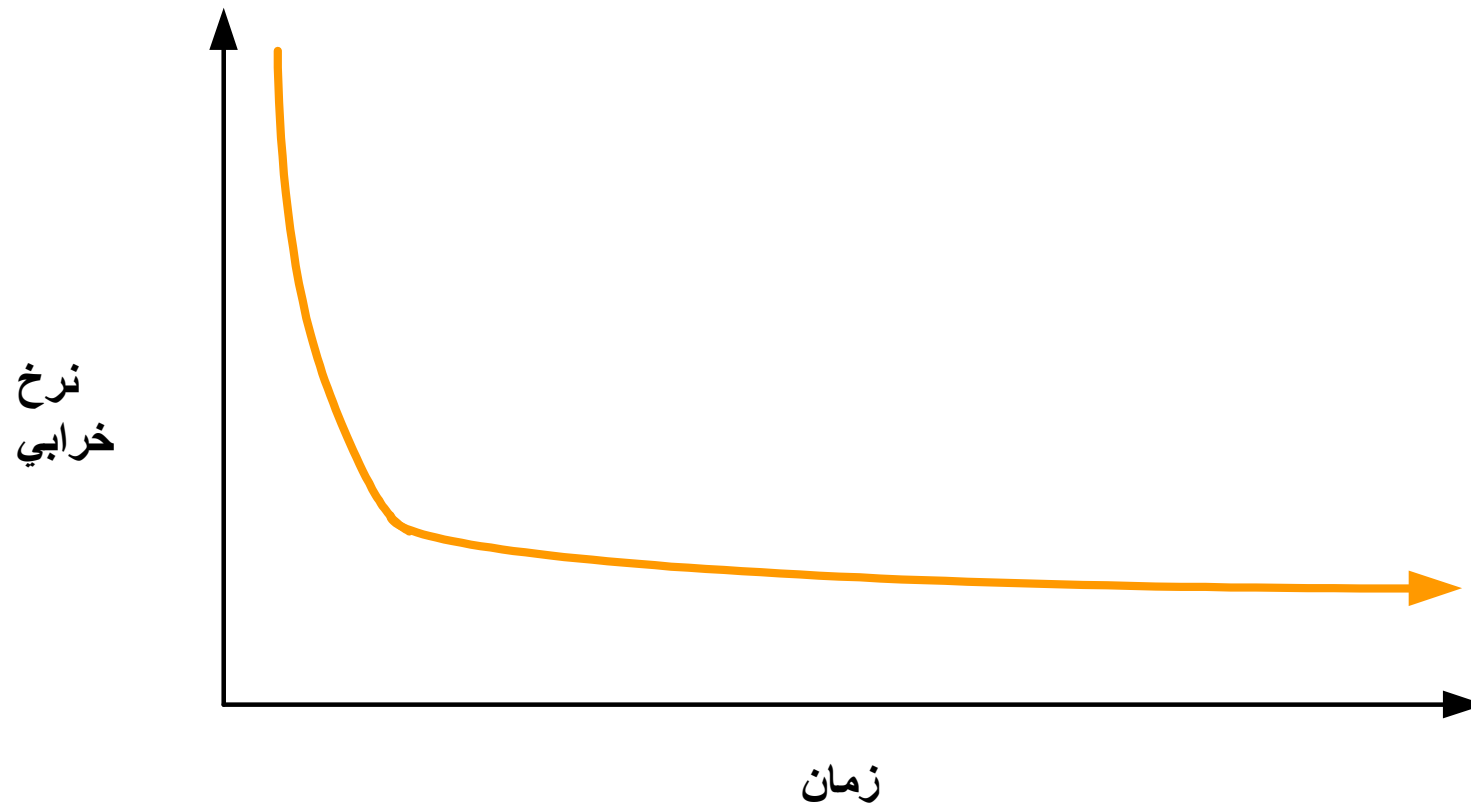


نرم افزار با گذشت زمان دچار فرسودگی نشده بلکه فاسد می گردد ۲



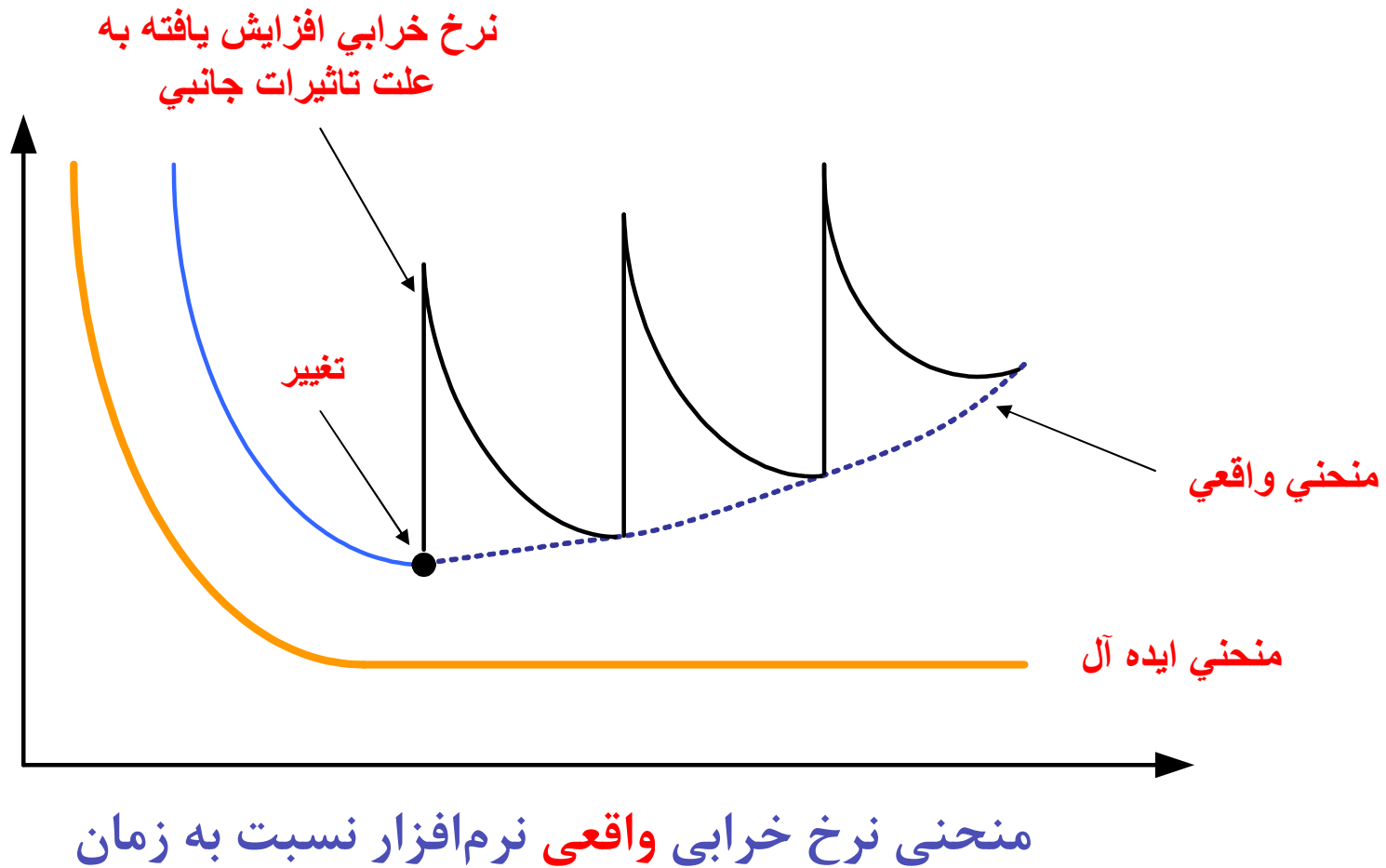
منحنی نرخ خرابی سخت افزار نسبت به زمان

تفاوت نرم افزار و سخت افزار (ادامه)

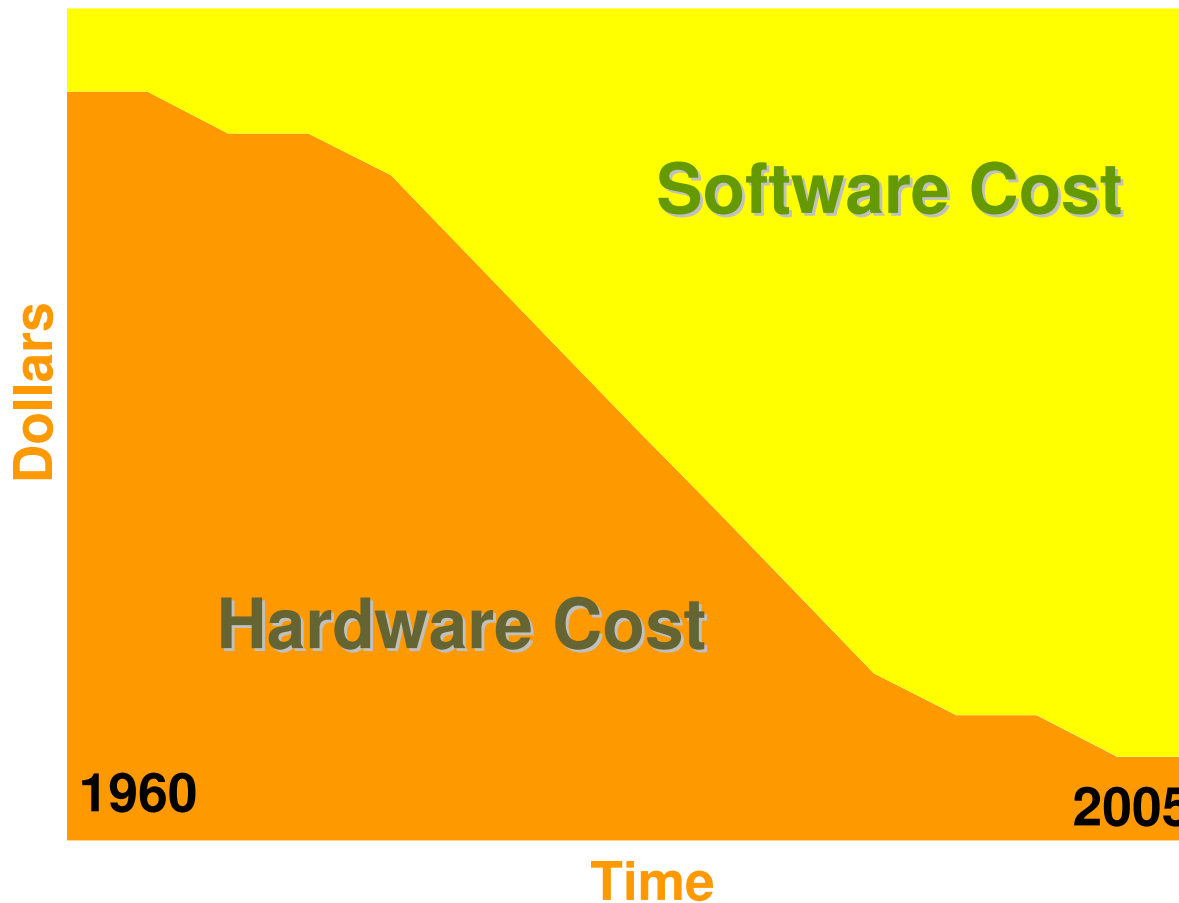


منحنی نرخ خرابی ایده آل نرم افزار نسبت به زمان

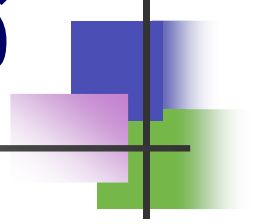
تفاوت نرم افزار و سخت افزار (ادامه)



تفاوت نرم افزار و سخت افزار (ادامه)



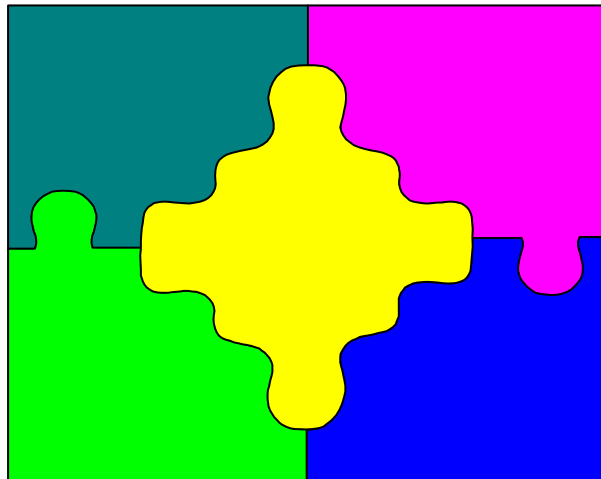
تفاوت نرم افزار و سخت افزار (ادامه)



بیشتر نرم افزارها به شیوه **سنتی** ساخته می شوند و از ترکیب **مولفه های استاندارد** تولید نمی گردند

۳

usability.100% to reach t'can .Weusable component



پیچیدگی ذاتی نرم افزار



”بدلیل تفاوت ذاتی بین نرم افزار و سخت افزار پیچیدگی خاصی در ابعاد مختلف از جمله در تعریف نرم افزار، طراحی، پیاده سازی، تست و نگهداری آن وجود دارد“

■ ویژگیهای این پیچیدگی عبارتند از:

- با پیچیدگی سیستمهای طبیعی و محصولات فیزیکی ساخت دست بشر متفاوت است
- یک خاصیت ذاتی سیستمهای نرم افزاری بزرگ



بنابراین "نمی‌توان این پیچیدگی را از بین برد بلکه باید
آن را کنترل نمود

عوامل پدید آورنده پیچیدگی نرم افزار



■ پیچیدگی حوزه مساله

■ نیازمندی‌های گوناگون و متضاد

They can't talk clearly. Technical langu

■ ارتباط بین کاربر و مهندس نرم افزار

■ تغییر نیازها

Team work is complex. Many teams working on

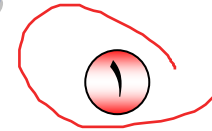
■ پیچیدگی فرآیند تولید

■ انعطاف پذیری نرم افزار و استاندارد نبودن آن

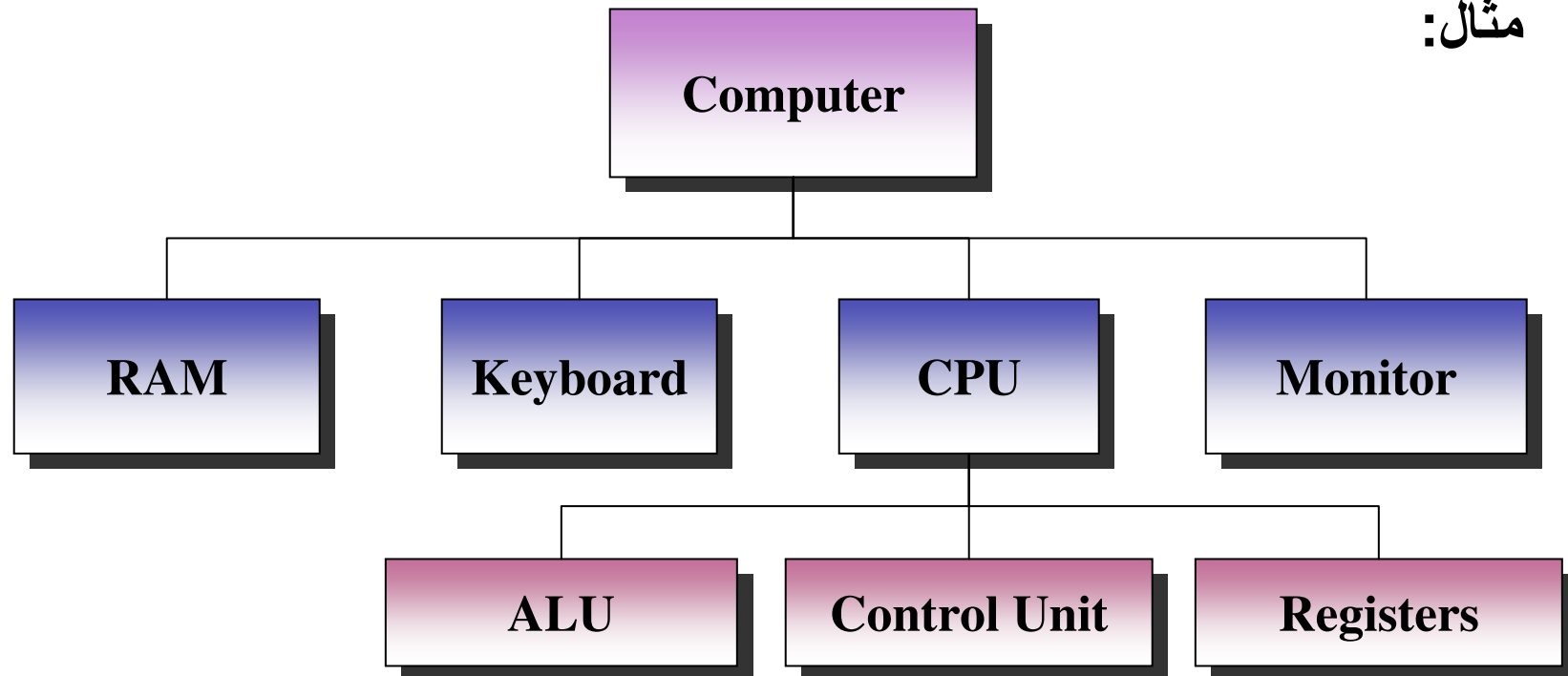
■ مشکل توصیف رفتار سیستم‌های گسسته

ویژگیهای سیستمهای پیچیده

”در اغلب سیستمهای پیچیده، پیچیدگی به صورت
سلسله مراتب (Hierarchy) ظاهر می شود“



مثال:



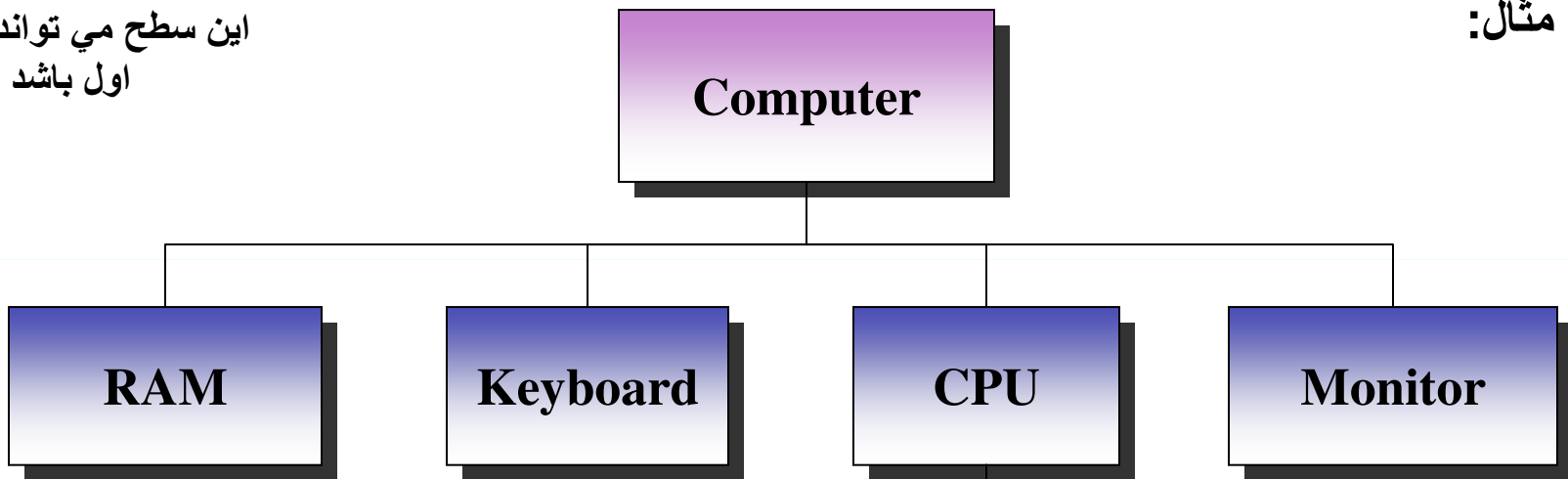
ویژگیهای سیستمهای پیچیده (ادامه)

”انتخاب مؤلفه‌های اولیه سیستم، امری است نسبتاً دلخواه بوده و تا حدود زیادی بستگی به دید طراح سیستم دارد“

۲

مثال:

این سطح می‌تواند انتخاب اول باشد



ممکن است این سطح انتخاب اول باشد!



ویژگیهای سیستم‌های پیچیده (ادامه)



”در سیستمی که از چند زیرسیستم تشکیل می‌گردد،



ارتباط بین اجزای درونی هر زیر سیستم

(Intra-component Linkage)

high cohesion

قویتر از ارتباط بین خود زیر سیستم‌ها

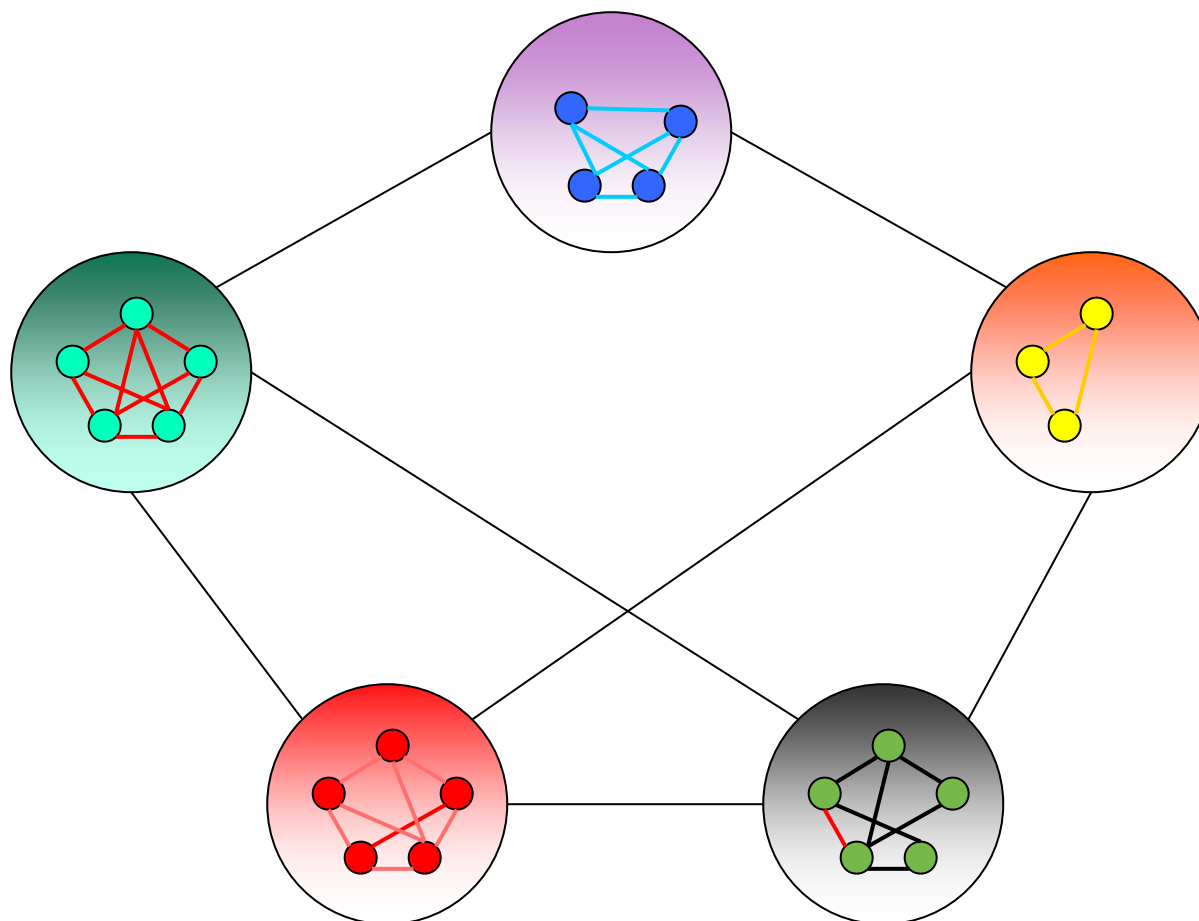
(Inter-components Linkage) است“

loosely coupled

ویژگیهای سیستم‌های پیچیده (ادامه)



مثال:



ویژگیهای سیستمهای پیچیده (ادامه)



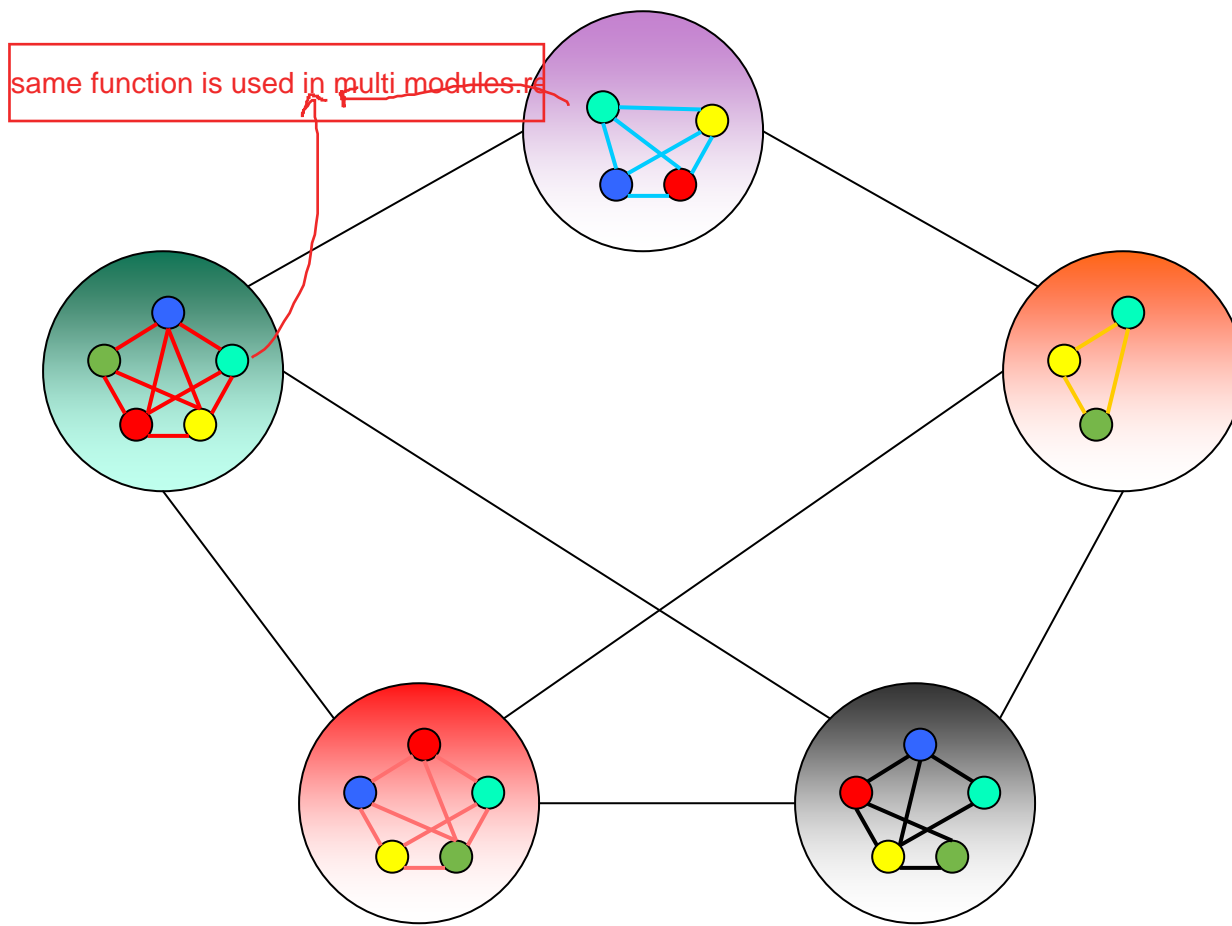
”سیستمهای سلسله مراتبی معمولاً از تعداد کمی از زیرسیستمهای مشخص و متفاوت تشکیل می‌شوند که این زیرسیستمها به صورت‌های گوناگون و ترتیب‌های مختلف ظاهر می‌شوند“

۴

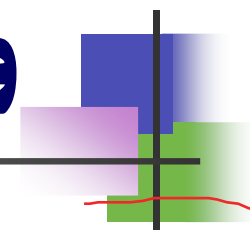
ویژگیهای سیستمهای پیچیده (ادامه)



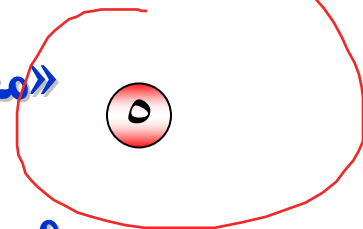
مثال:



ویژگیهای سیستمهای پیچیده (ادامه)



«معمولاً سیستمهای پیچیده که به صورت محکم و استوار عمل می کنند حاصل تکامل سیستمهای ساده ای هستند که به درستی عمل می کردند»

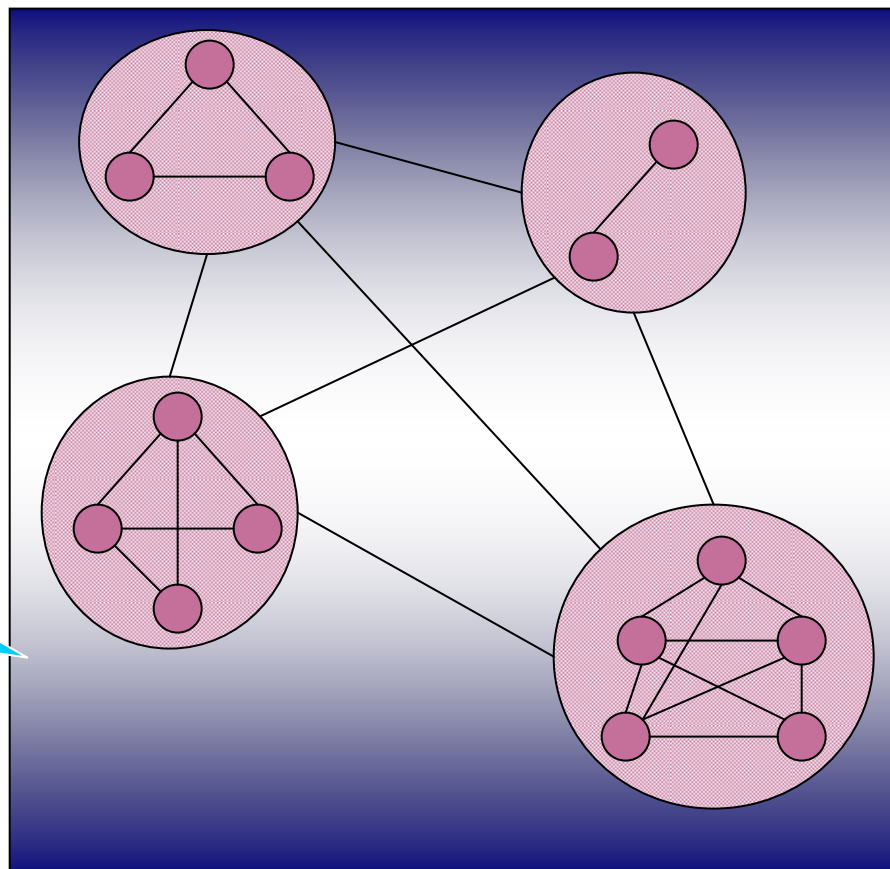
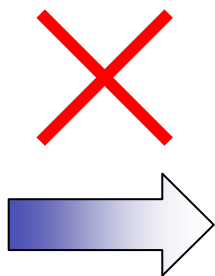
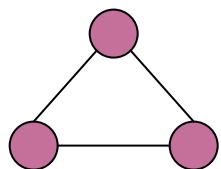


سیستمهای پیچیده که از ابتدا به صورت پیچیده طراحی می شوند، هرگز کار نخواهند کرد

ویژگیهای سیستم‌های پیچیده (ادامه)

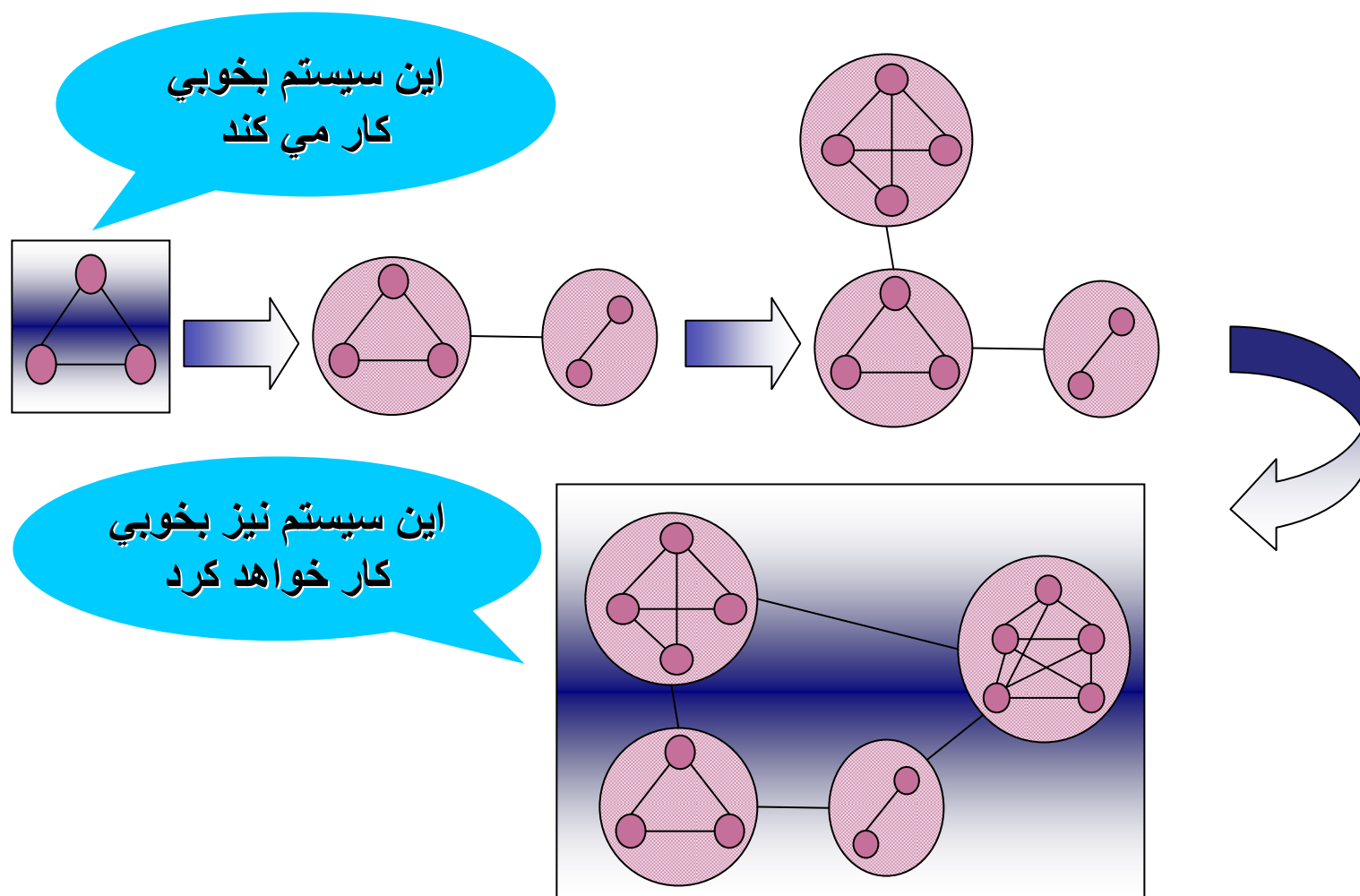


مثال:

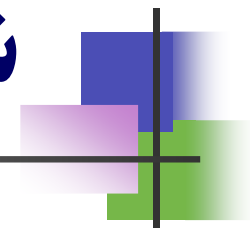


این سیستم کار
نخواهد کرد

ویژگیهای سیستم‌های پیچیده (ادامه)



شکل اصلی سیستم‌های پیچیده



”مجموعه ویژگی‌های کلی که در همه سیستم‌های پیچیده، صرفنظر از اندازه پیچیدگی، به چشم می‌خورند“

- ساختار کلاس: سلسه مراتب IS-A
- ساختار شیء: سلسه مراتب PART-OF

inheritance vs composition

شکل اصلی سیستم‌های پیچیده = خواص پنجگانه +
سلسله مراتب IS-A + سلسه مراتب PART-OF

شکل اصلی سیستم‌های پیچیده (ادامه)

■ پیچیدگی سازمان‌نیافته: وضعیت مسئله قبل از تحلیل و طراحی (اجزاء بسیار زیاد)

■ پیچیدگی سازمان‌یافته: وضعیت مسئله پس از تحلیل و طراحی

