

بسمه تعالی

فصل اول

بحران نرم افزار و بررسی پیچیدگی نرم افزار

اهداف جلسه



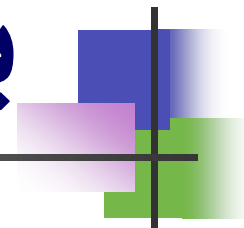
- درک مشکلات تولید نرم افزار و راه مقابله با آن
- درک اهمیت مهندسی نرم افزار
- درک ضرورت روی آوردن به یک متدولوژی مدون
- درک ماهیت نرم افزار و تفاوت آن با بقیه محصولات مهندسی
- درک پیچیدگی نرم افزار و عوامل پدید آورنده آن
- آشنائی با مبحث سیستم‌های پیچیده و ویژگیهای آنها

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



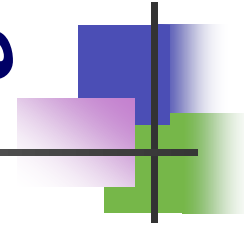
“پیشرفت شگرف سخت افزار و ضعف روش های توسعه
نرم افزار و ناتوانی این روش ها در کنترل پیچیدگی
نرم افزار بحران نرم افزار را بوجود آورد”

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

تعریف مهندسی نرم افزار

«بکارگیری یک روش سیستماتیک، منظم و قابل اندازه گیری

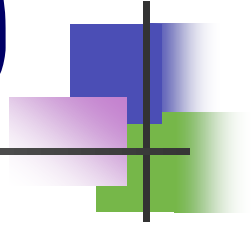
برای تولید و توسعه، عملیاتی کردن و نگهداری نرم افزار؛

بکارگیری اصول مهندسی در تولید نرم افزار»

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

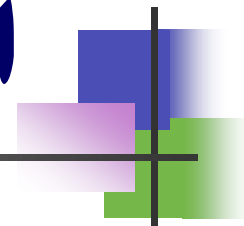
اصول مهندسی نرم افزار



تکنولوژی مهندسی نرم افزار



اهمیت متدولوژی



«یکی از علل اساسی بحران نرم افزار عدم وجود روش های مناسبی برای تولید نرم افزار»

■ روش (*Method*)

- فرآیندی منظم که با استفاده از مجموعه ای از نمادگذاری های خوش تعریف، مجموعه ای از مدل ها را ایجاد می کند که هر کدام بخشی از سیستم نرم افزاری در دست تولید (یا توسعه) را توصیف می نمایند

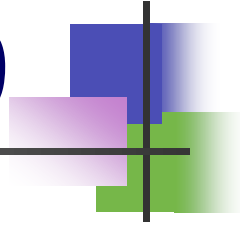
اهمیت متدولوژی (ادامه)

■ 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

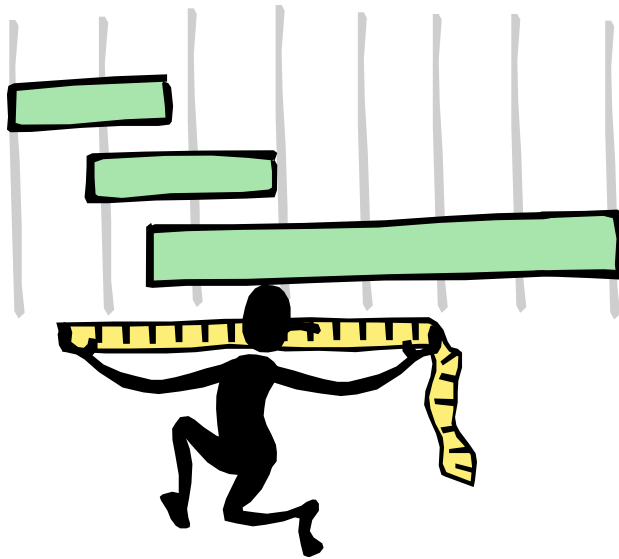
ویژگیهای یک متدولوژی مطلوب



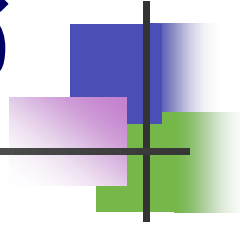
- ارائه تعاریفی از مفاهیم اولیه بکار رفته در متدولوژی
- ارائه مدلی برای فرآیند تولید
- داشتن مدل زیربنایی (مدل معماری)
- ارائه یک شیوه علامت‌گذاری استاندارد
- معرفی تکنیک‌هایی برای پیاده‌سازی متدولوژی که توانایی کنترل پیچیدگی سیستم‌های کنونی را دارا باشند

ویژگیهای یک متدولوژی ... (ادامه)

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



تفاوت نرم افزار و سخت افزار



فرآیند تولید نرم افزار یک فرآیند مهندسی است نه
یک فرآیند تولید صنعتی



نرم افزار

بوسیله استفاده کنندگان نهایی

فرآیند مهندسی که برای هر کاربرد جدید منحصر به فرد است

منطقی

سخت افزار

بوسیله متخصصین

مکانیکی

فیزیکی

تعیین مشخصات

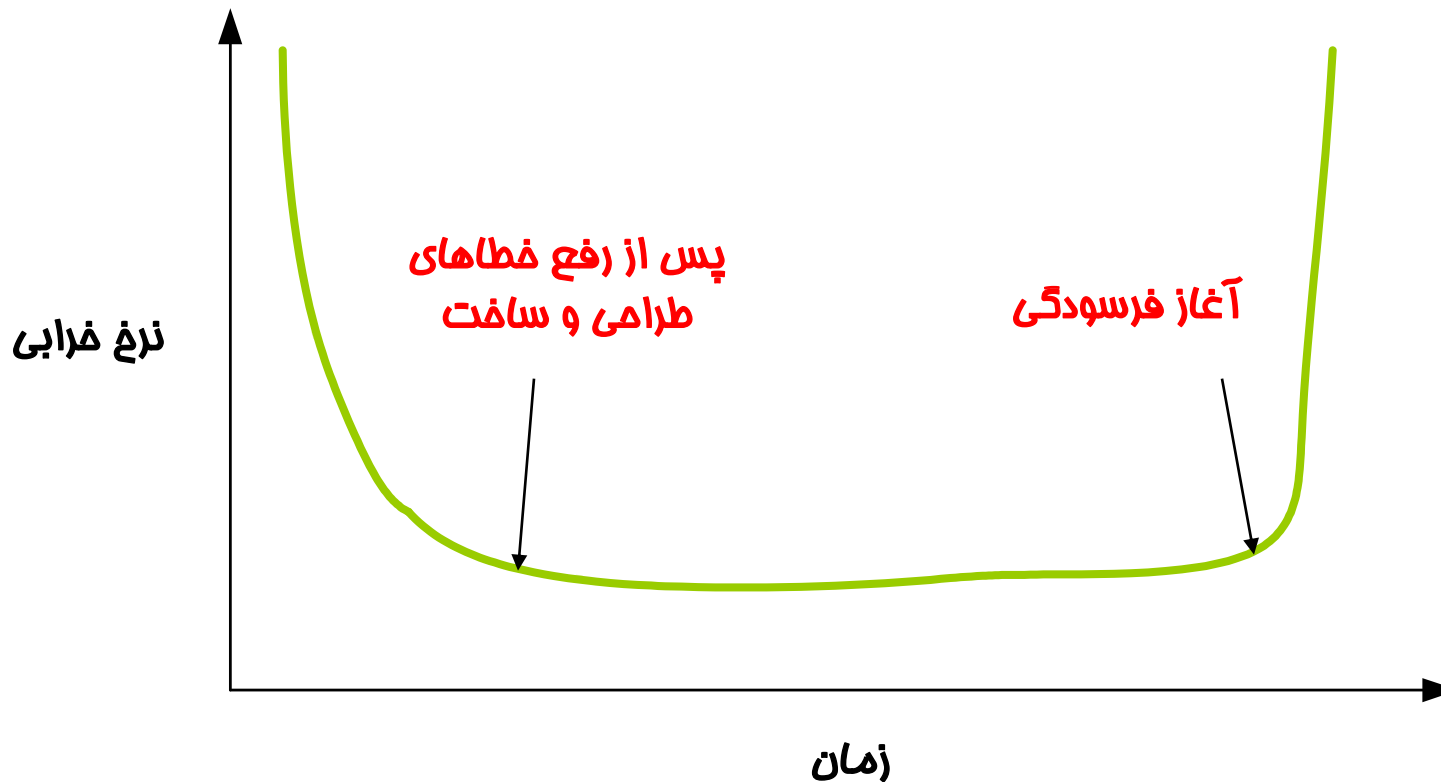
ماهیت فرآیند تولید

ماهیت محصول

تفاوت نرم افزار و سخت افزار (ادامه)

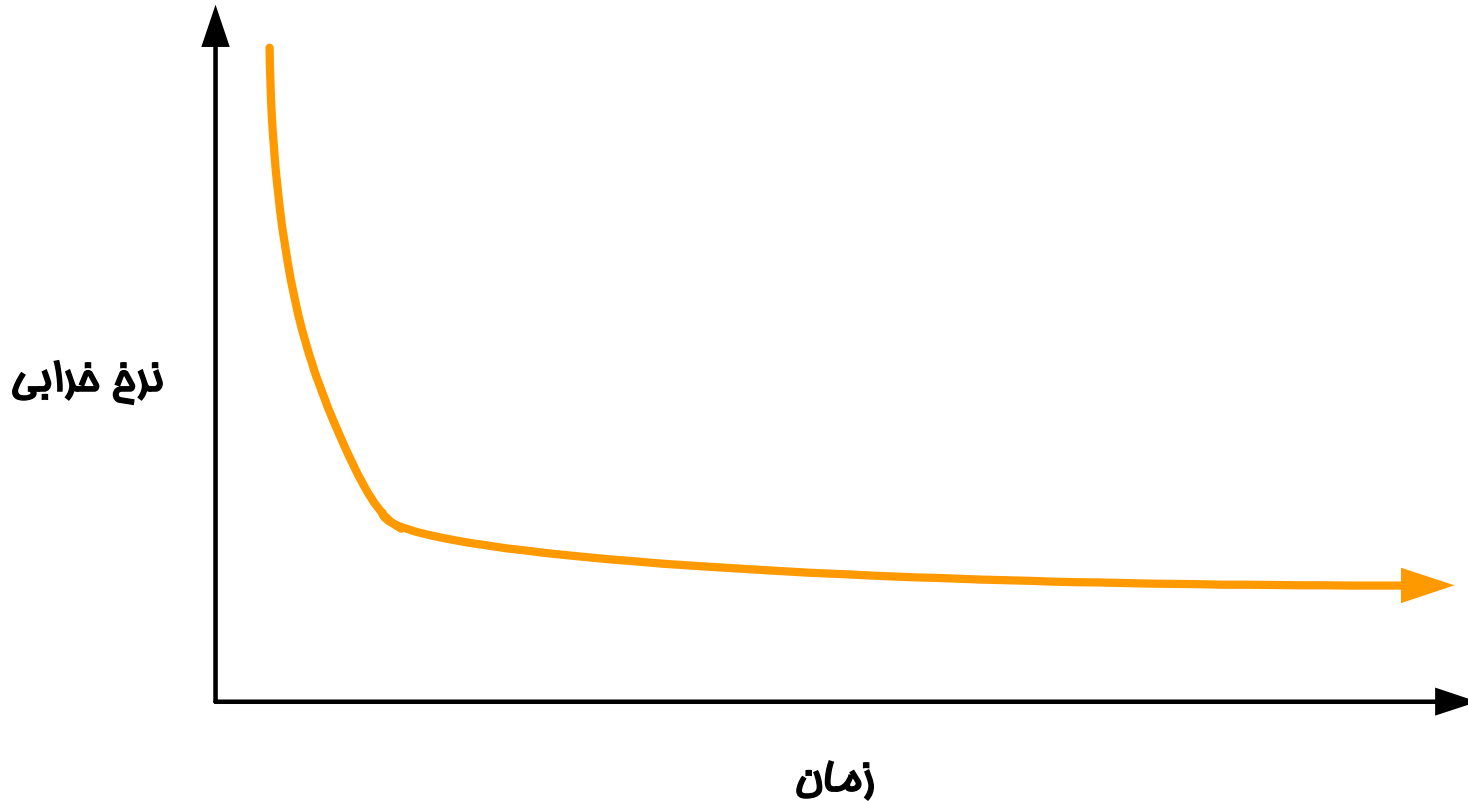


نرم افزار با گذشت زمان دچار فرسودگی نشده بلکه فاسد می گردد



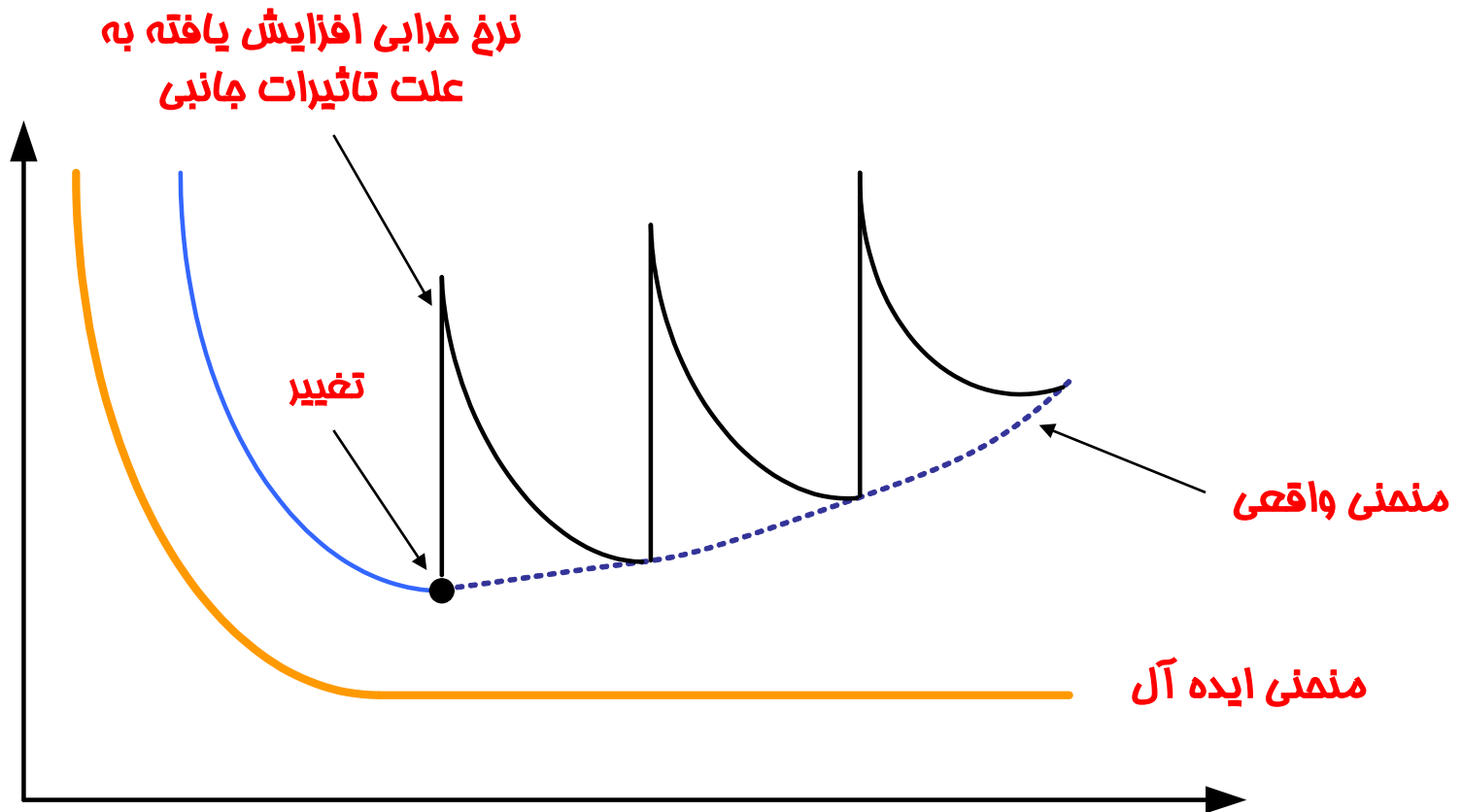
منحنی نرخ خرابی سخت افزار نسبت به زمان

تفاوت نرم افزار و سخت افزار (ادامه)



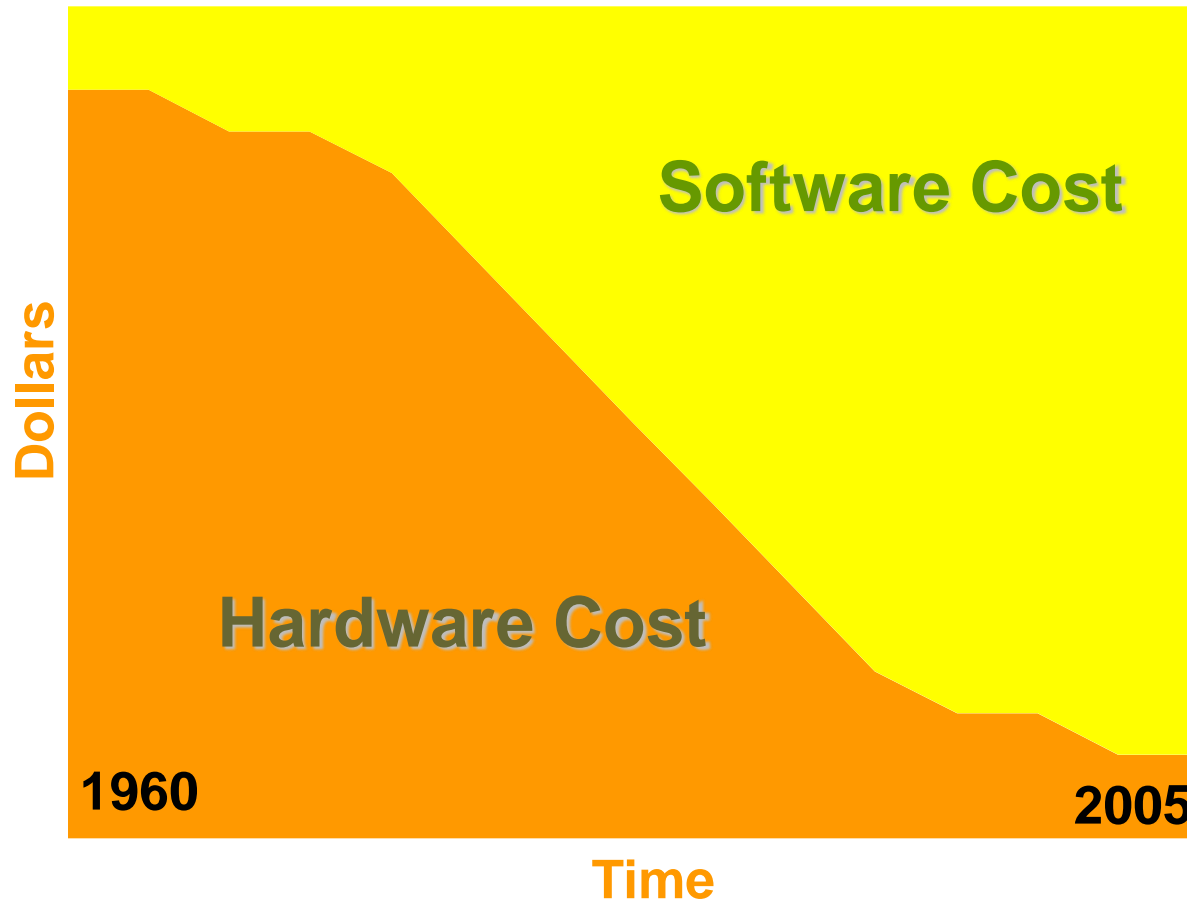
منحنی نرخ خرابی ایده آل نرم افزار نسبت به زمان

تفاوت نرم افزار و سخت افزار (ادامه)

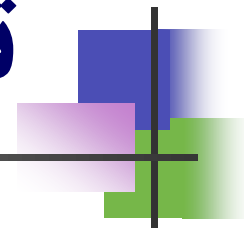


منحنی نرخ خرابی واقعی نرم افزار نسبت به زمان

تفاوت نرم افزار و سخت افزار (ادامه)

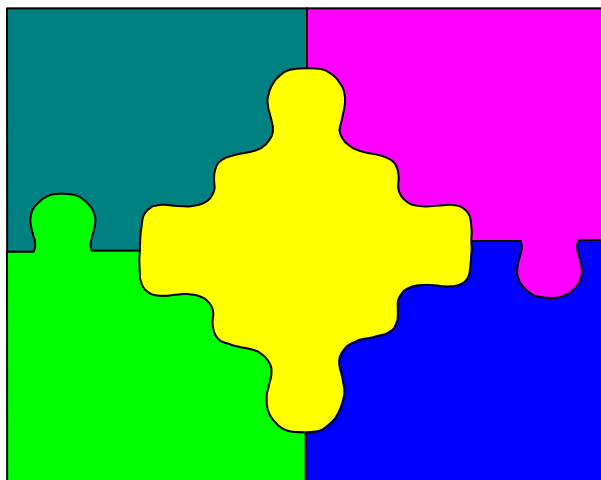


تفاوت نرم افزار و سخت افزار (ادامه)



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

۳



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،

بنابراین ”نمی توان این پیچیدگی را از بین برد بلکه باید آن را کنترل نمود

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

عوامل پدید آورنده پیچیدگی نرم افزار

■ پیچیدگی حوزه مساله

- نیازمندی‌های گوناگون و متضاد
- ارتباط بین کاربر و مهندس نرم افزار
- تغییر نیازها

■ پیچیدگی فرآیند تولید

- انعطاف پذیری نرم افزار و استاندارد نبودن آن
- مشکل توصیف رفتار سیستم‌های پیچیده

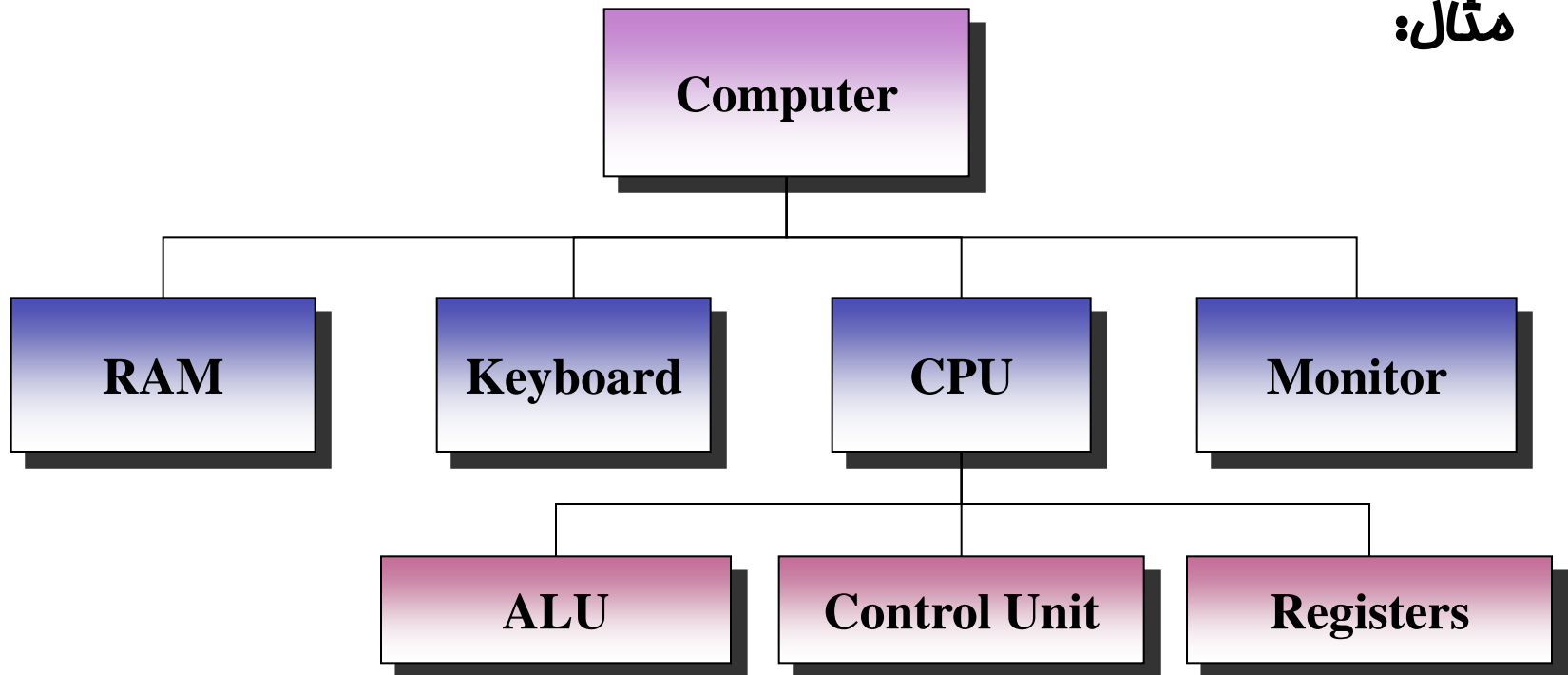
ویژگیهای سیستمهای پیچیده

”در اغلب سیستمهای پیچیده، پیچیدگی به صورت

سلسله مراتب (Hierarchy) ظاهر می شود“



مثال:



ویژگیهای سیستم‌های پیچیده (ادامه)

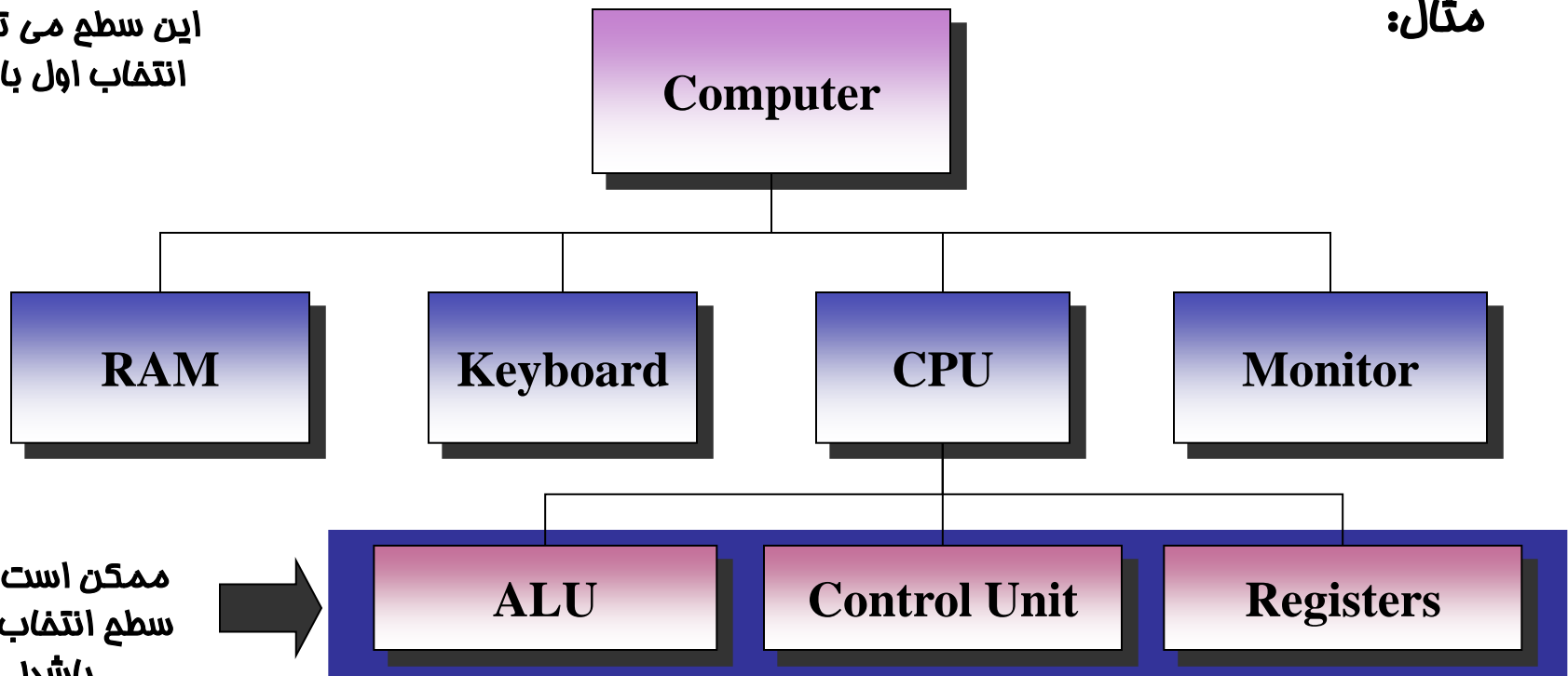
”انتخاب مؤلفه‌های اولیه سیستم، امری است نسبتاً دلخواه

۲

بوده و تا حدود زیادی بستگی به دید طراح سیستم دارد“

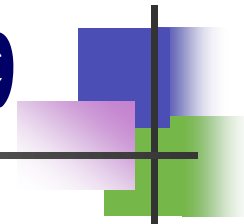
مثال:

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

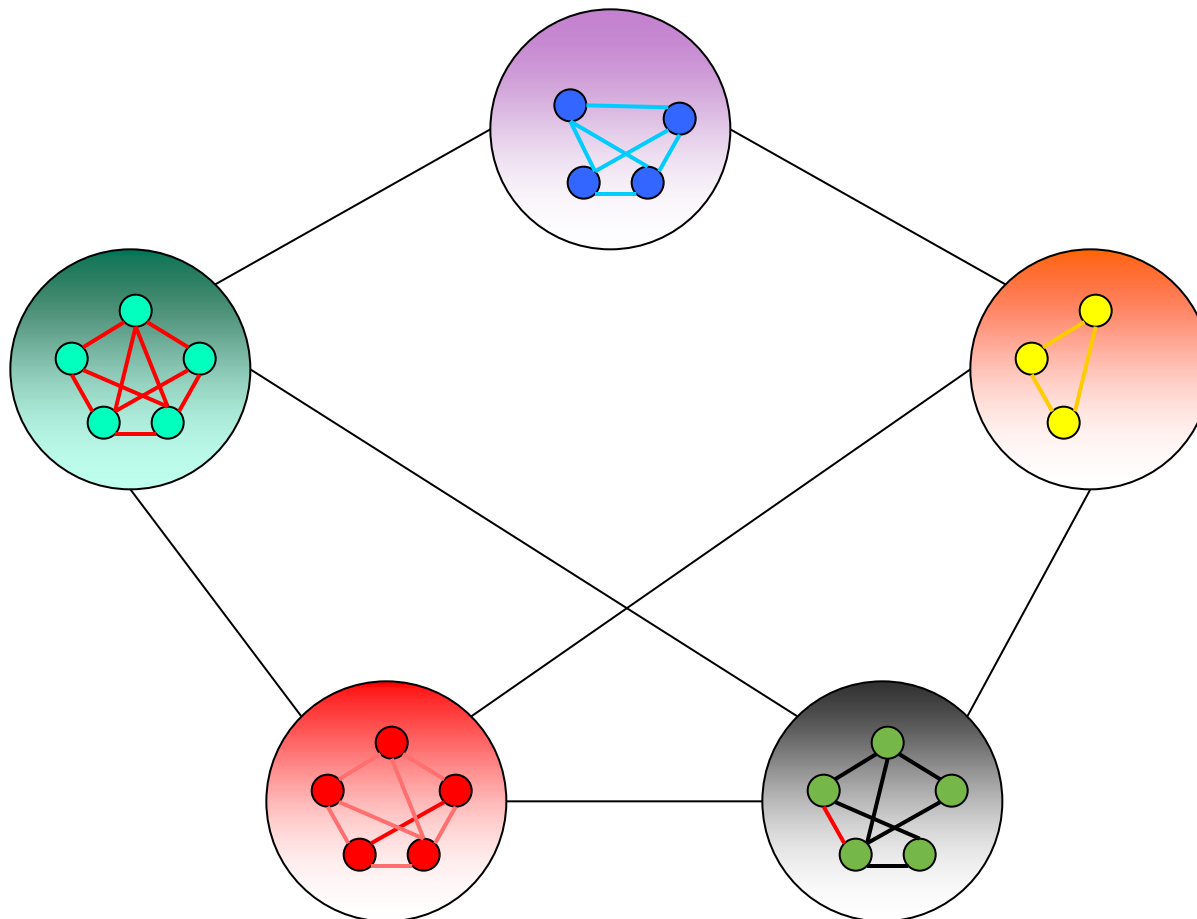
(Intra-component Linkage)

قویتر از ارتباط بین خود زیر سیستم‌ها

(Inter-components Linkage) است“

ویژگیهای سیستم‌های پیچیده (ادامه)

مثال:

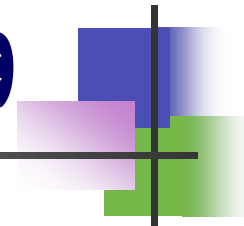


ویژگیهای سیستم‌های پیچیده (ادامه)

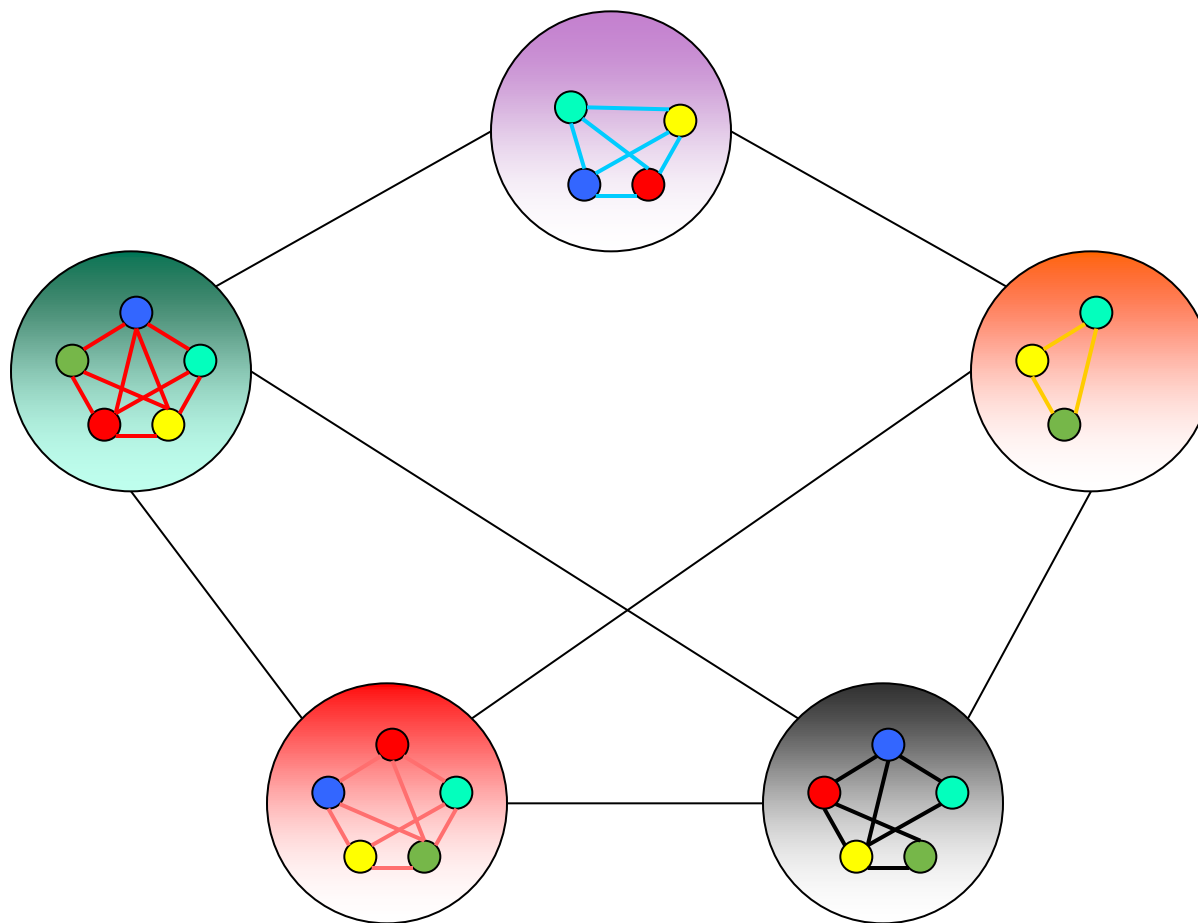
”سیستم‌های سلسله مراتبی معمولاً از تعداد کمی از زیرسیستم‌های مشخص و متفاوت تشکیل می‌شوند که این زیرسیستم‌ها به صورت‌های گوناگون و ترتیب‌های مختلف ظاهر می‌شوند“



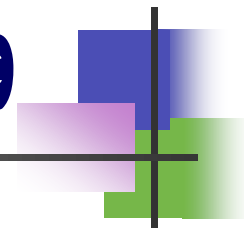
ویژگیهای سیستم‌های پیچیده (ادامه)



مثال:



ویژگیهای سیستم‌های پیچیده (ادامه)



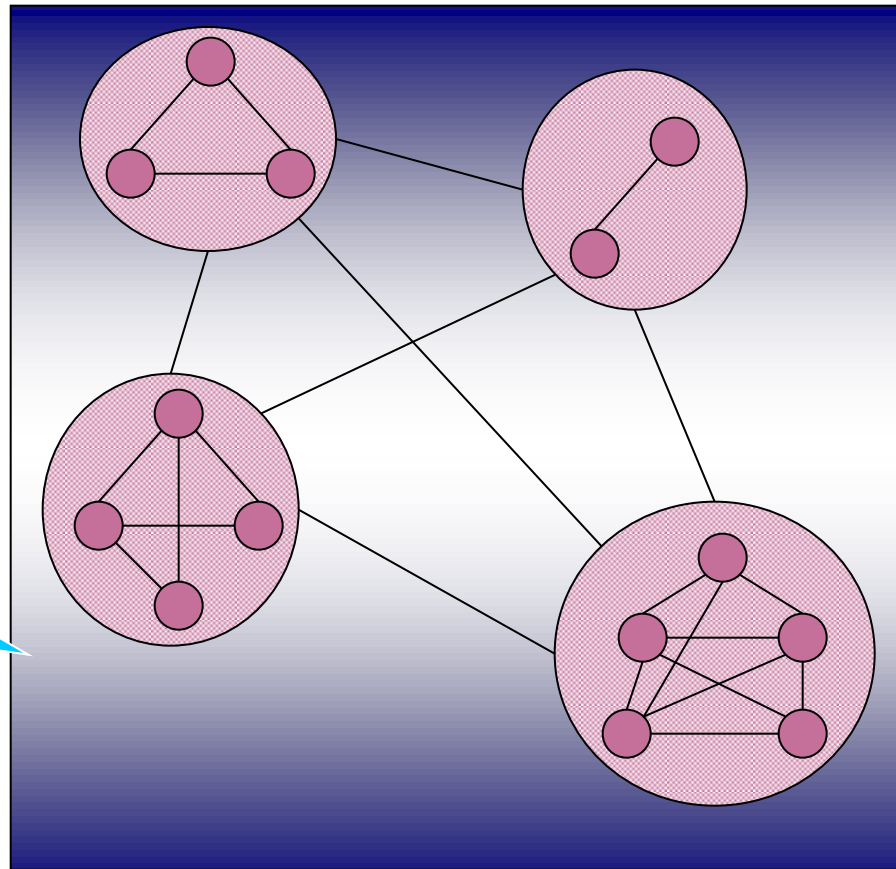
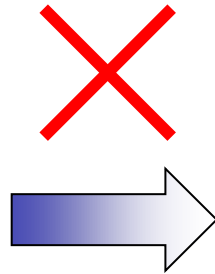
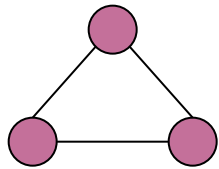
«معمولاً سیستم‌های پیچیده که به صورت **محکم و استوار** عمل می‌کنند حاصل **تکامل سیستم‌های ساده‌ای** هستند که به درستی عمل می‌کردند»



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

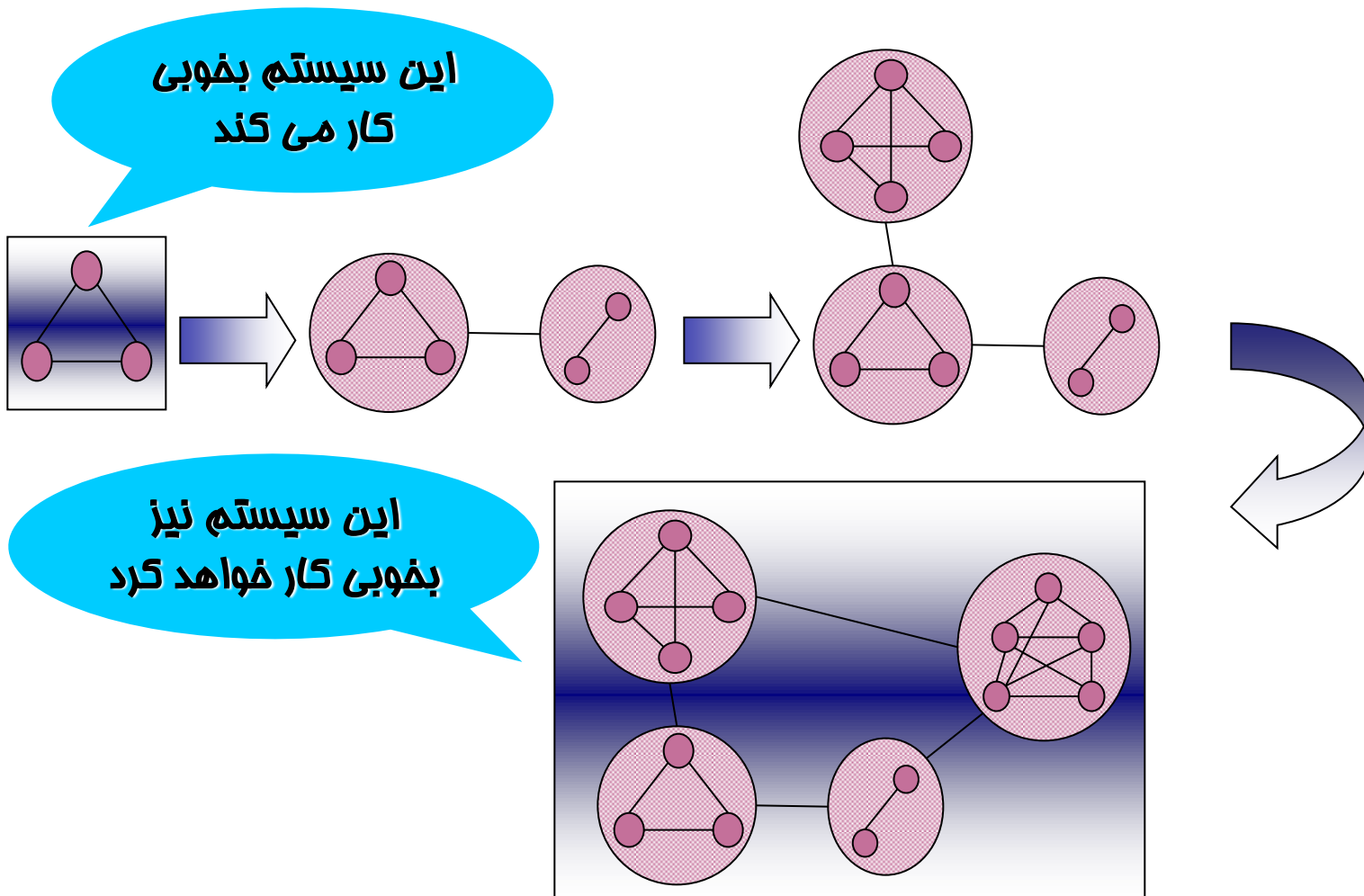
ویژگیهای سیستم‌های پیچیده (ادامه)

مثال:



این سیستم کار نخواهد کرد

ویژگیهای سیستم‌های پیچیده (ادامه)



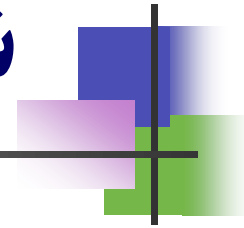
شکل اصلی سیستم‌های پیچیده

”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

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

شکل اصلی سیستم‌های پیچیده (ادامه)

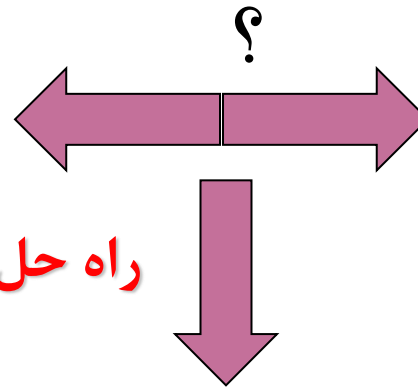


■ پیچیدگی سازمان نیافته: وضعیت مسئله قبل از تحلیل و طراحی (اجزاء بسیار زیاد)

■ پیچیدگی سازمان یافته: وضعیت مسئله پس از تحلیل و طراحی

پیچیدگی سیستم‌های نرم‌افزاری
روز به روز در حال افزایش است

قدرت پردازش همزمان
مغز انسان محدود است



**بوسیله متدولوژی
خوب**

تجزیه
نحوه اعمال



پرسش و پاسخ