

بسمه تعالی

درس مهندسی نرم افزار پیشرفته

بحران تره افزار و بررسی مسئله پیچیدگی در تره افزار

نیمسال دوم : ۸۷ - ۱۳۸۶

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

- **بهران نرم افزار**
- **مهندسی نرم افزار و ضرورت آن**
- **متدولوژی و اهمیت آن**
- **تفاوت نرم افزار و سخت افزار**
- **پیمیدگی ذاتی نرم افزار**
- **عوامل پدید آورنده پیمیدگی نرم افزار**
- **ساختار سیستمهای پیمیده**
- **ویژگی سیستمهای پیمیده**
- **پیمیدگی سازمان یافته و سازمان نیافته**

بحران نرم افزار



”پیشرفت شگرف سخت افزار و ضعف روشهای تولید نرم افزار و ناتوانی این روشها در کنترل پیچیدگی نرم افزار بحران نرم افزار را بوجود آورد“

علايه اين بحران عبارتند از:

- عدم بهره گیری کامل از قدرت سخت افزار
- ناتوانی روشهای تولید نرم افزار در پاسخگوئی به افزایش تقاضا

بحران نرم افزار (ادامه)



- هزینه های هنگفت تولید نرم افزار
- 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 روشهای برنامه ریزی و کنترل
- استفاده از تجربیات دیگر رشته های مهندسی (مثال الگوها Patterns)
- تهیه و جایگزینی استانداردهای مطمئن

مهندسی نرم افزار و ضرورت آن (ادامه)



تعریف:

”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

مهندسی نرم افزار و ضرورت آن (ادامه)



- کاهش هزینه
- تحویل به موقع
- استفاده از مولفه های استاندارد
- استفاده مجدد

مهندسی نرم افزار و ضرورت آن (ادامه)



تکنولوژی مهندسی نرم افزار

فرایند تولید

متدولوژی

ابزارهای خودکار
سازی
(CASE Tools)

اهمیت متدولوژی

”یکی از علل اساسی بحران نرم افزار عدم وجود روشهای مناسبی برای تولید نرم افزار“

روش: فرایندی منظم که با استفاده از مجموعه ای از نماد گذارهای خوش تعریف، مجموعه ای از مدلها را ایجاد می کند که هر کدام بخشی از سیستم نرم افزاری در دست تولید (یا توسعه) را توصیف می نماید.

متدولوژی: مجموعه ای از روشها که در تمام چرخه حیات سیستم نرم افزاری اعمال شده و بر یک نوع نگرش کلی درباره جهان نرم افزار متکی می باشند.

اهمیت متدولوژی (ادامه)

”بنظر بسیاری از متخصصین بکارگیری یک متدولوژی بدون در تولید نرم افزار می تواند تا اندازه قابل توجهی مشکلات بیان شده را برطرف سازد“

ویژگیهای یک متدولوژی مطلوب:

- ارائه تعاریف از مفاهیم اولیه بکار رفته در متدولوژی
- ارائه مدلی برای فرایند تولید
- داشتن مدل زیر بنایی (مدل معماری)

اهمیت متدولوژی (ادامه)



- ارائه یک شیوه علامت گذاری استاندارد
- معرفی تکنیکهایی برای پیاده سازی متدولوژی که توانایی کنترل پیچیدگی سیستمهای کنونی را دارا باشند
- 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



① فرایند تولید نرم افزار یک فرایند مهندسی است نه یک فرایند تولید صنعتی

نرم افزار

بوسیله استفاده کننده گان نهائی

فرایند مهندسی که برای هر کاربرد جدید منحصر به فرد است

منطقی

سخت افزار

بوسیله متخصصین

مکانیکی

فیزیکی

تعیین مشخصات:

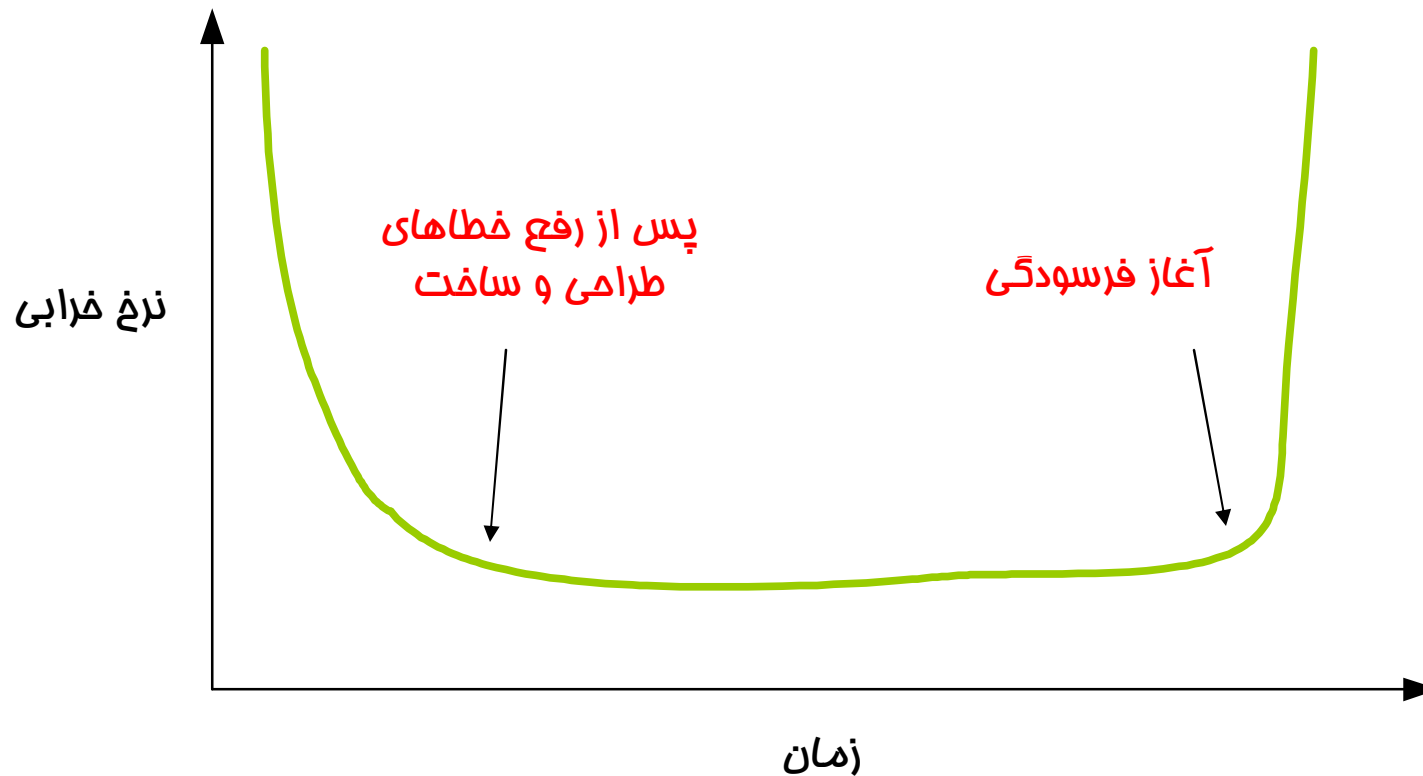
ماهیت فرایند تولید:

ماهیت محصول:

تفاوت نرم افزار و سخت افزار (ادامه)

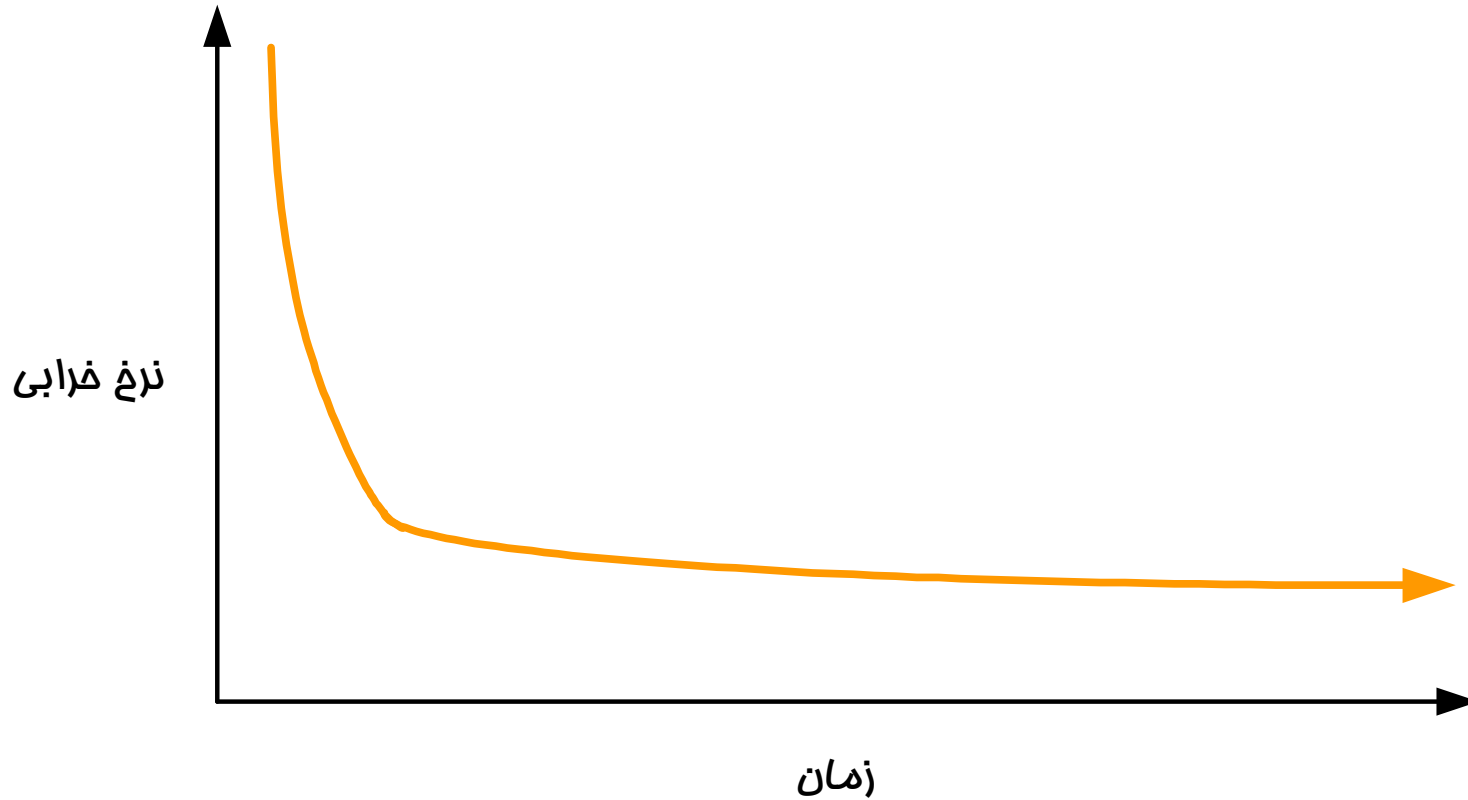


۲ نرم افزار با گذشت زمان دچار فرسودگی نشده بلکه فاسد می گردد



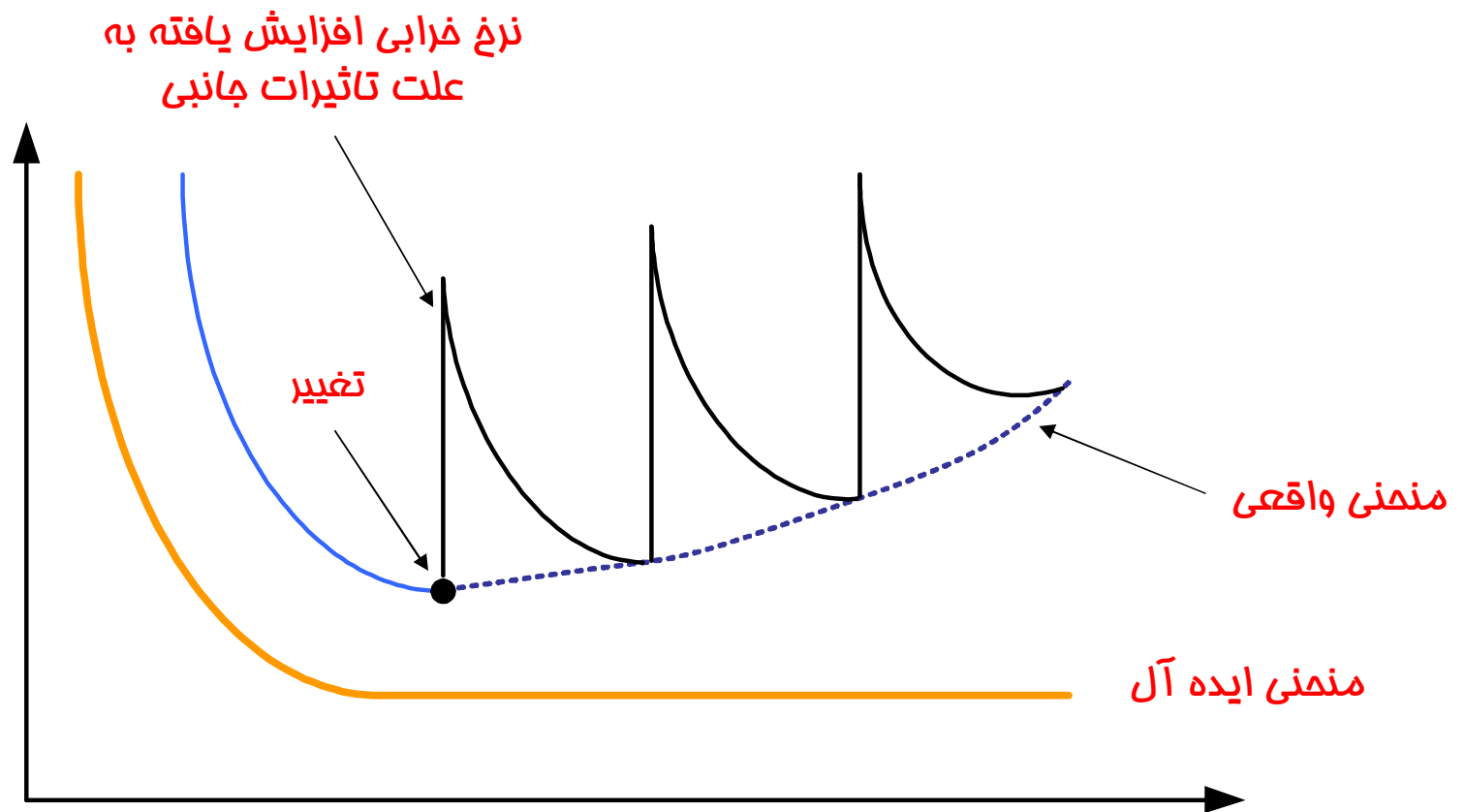
منحنی نرخ خرابی سخت افزار نسبت به زمان

تفاوت نرخ افزار و سخت افزار (ادامه)



منحنی نرخ خرابی ایده آل نرم افزار نسبت به زمان

تفاوت نرخ افزار و سخت افزار (ادامه)



منحنی نرخ خرابی واقعی نرم افزار نسبت به زمان

تفاوت نرم افزار و سخت افزار (ادامه)



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

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 ”نمی توان این پیچیدگی را از بین برد بلکه باید آنرا کنترل نمود“

عوامل پدید آورنده پیچیدگی نرم افزار

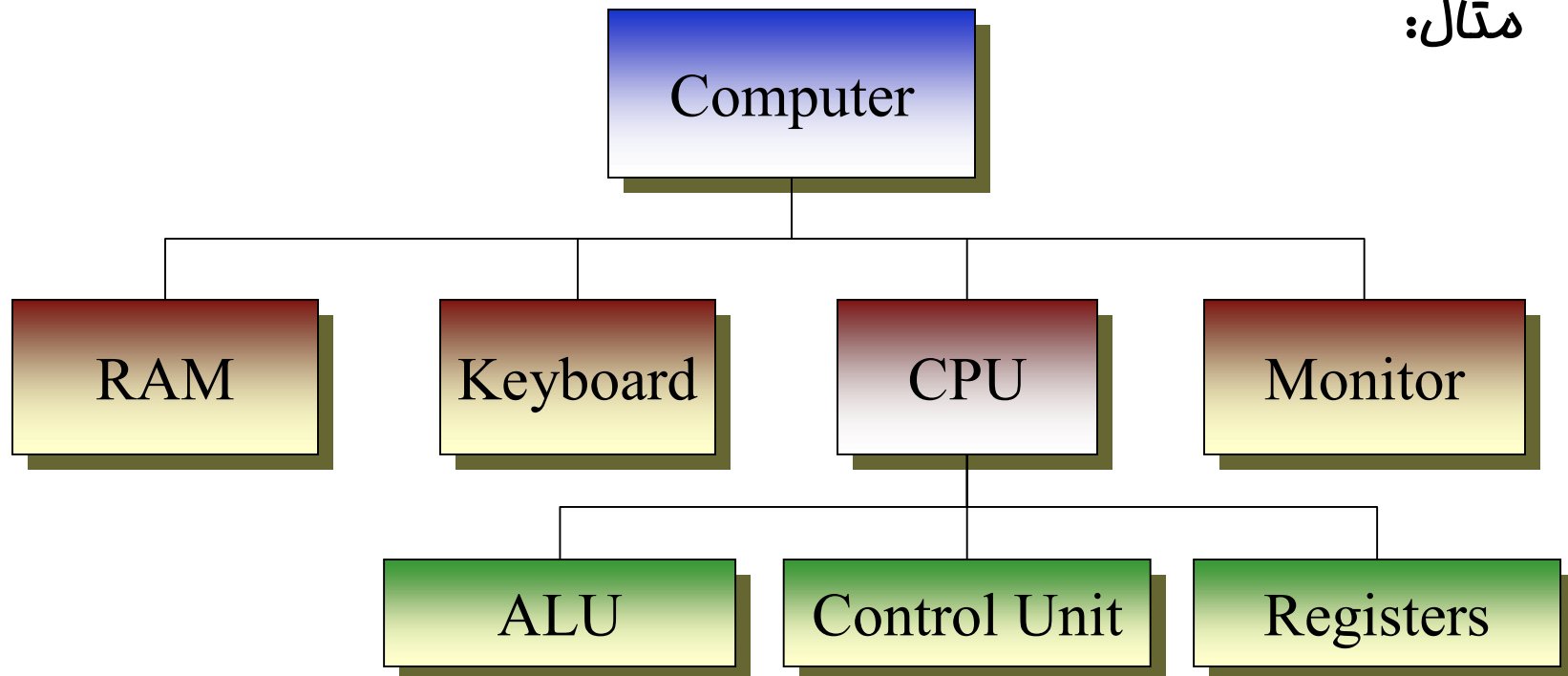


- پیچیدگی حوزه مساله
- نیازمندیهای گوناگون و متضاد
- ارتباط بین کاربر و مهندس نرم افزار
- تغییر نیازها
- پیچیدگی فرآیند تولید
- انعطاف پذیری نرم افزار و استاندارد نبودن آن
- مشکل توصیف رفتار سیستمهای گسسته

ویژگیهای سیستمهای پیچیده

① ”در اغلب سیستمهای پیچیده، پیچیدگی به صورت سلسله مراتب (*Hierarchy*) ظاهر می شود“

مثال:



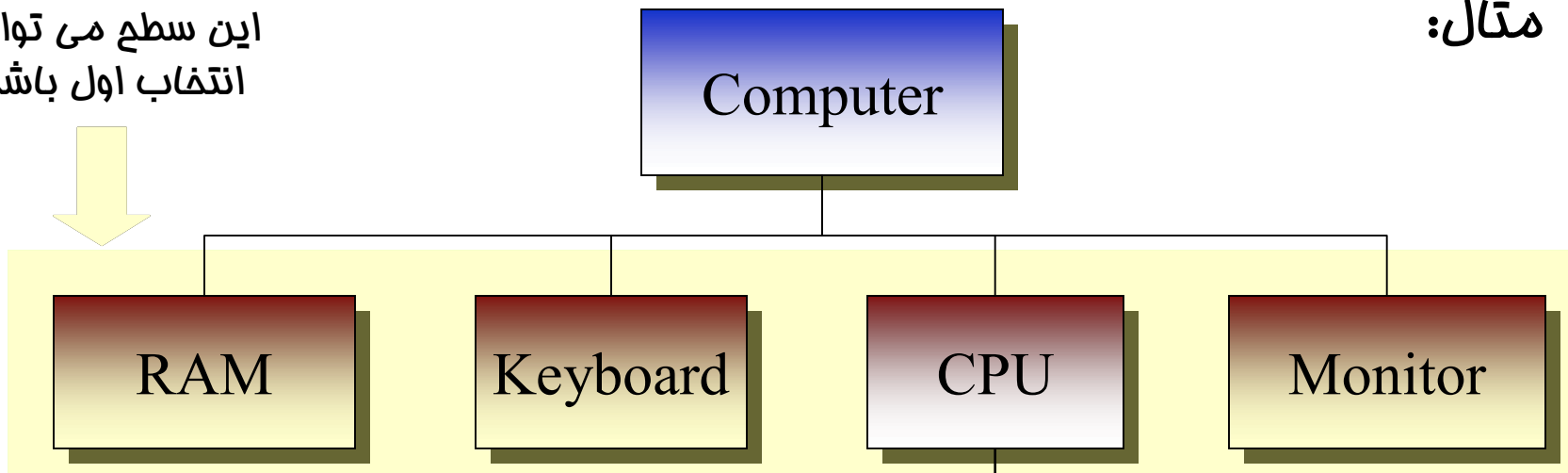
ویژگیهای سیستمهای پیچیده (ادامه)



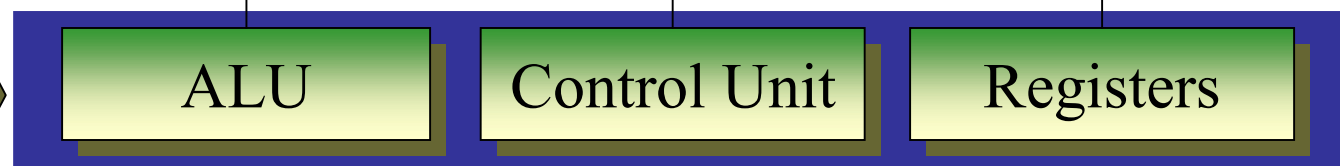
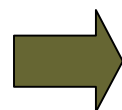
۲ ” انتخاب مؤلفه های اولیه سیستم، امری نسبتاً دلخواه بوده و تا حدود زیادی بستگی به دید طراح سیستم دارد“

مثال:

این سطح می تواند انتخاب اول باشد



ممکن است این سطح انتخاب اول باشد!



ویژگیهای سیستمهای پیچیده (ادامه)

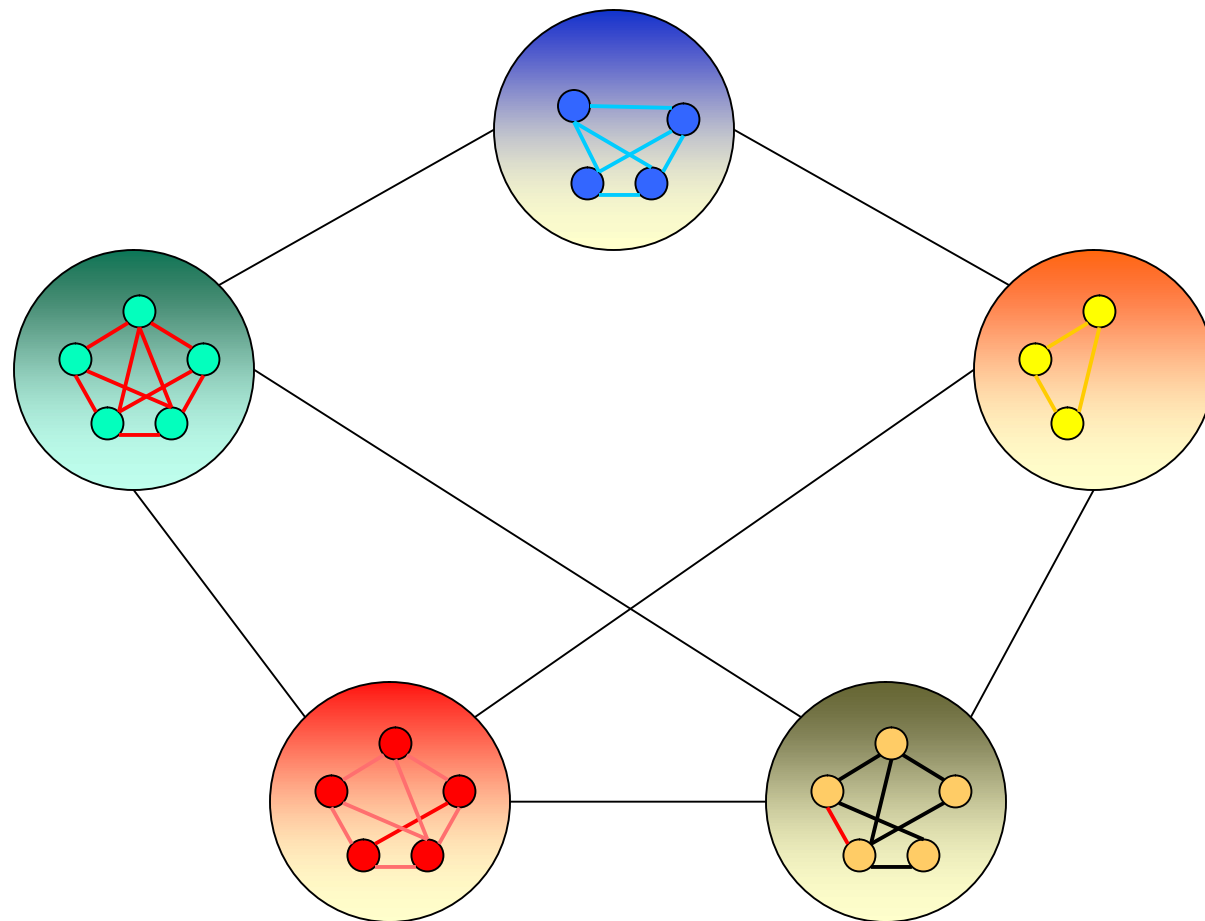


③ ” در سیستمی که از چند زیرسیستم تشکیل می گردد، ارتباط بین اجزای درونی هر زیر سیستم (*Intra-component Linkage*) قویتر از ارتباط بین خود زیر سیستمها (*Inter-component Linkage*) است“

ویژگیهای سیستمهای پیچیده (ادامه)



مثال:



ویژگیهای سیستمهای پیچیده (ادامه)



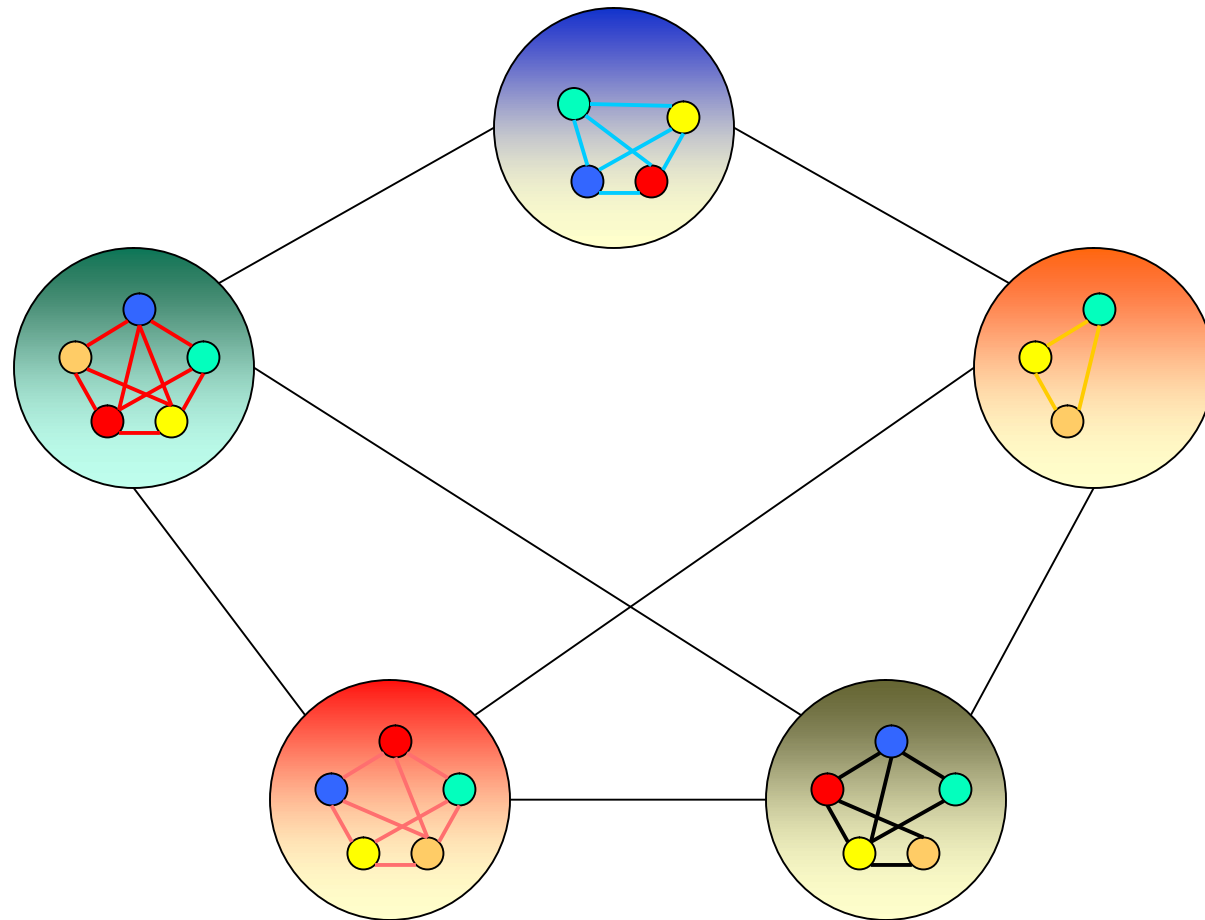
“سیستم های سلسله مراتبی معمولا از تعداد کمی از زیر سیستمهای مشخص و متفاوت تشکیل می شوند که این زیرسیستمها به صورتهای گوناگون و ترتیب های مختلف ظاهر می شوند“



ویژگیهای سیستمهای پیچیده (ادامه)



مثال:



ویژگیهای سیستمهای پیچیده (ادامه)

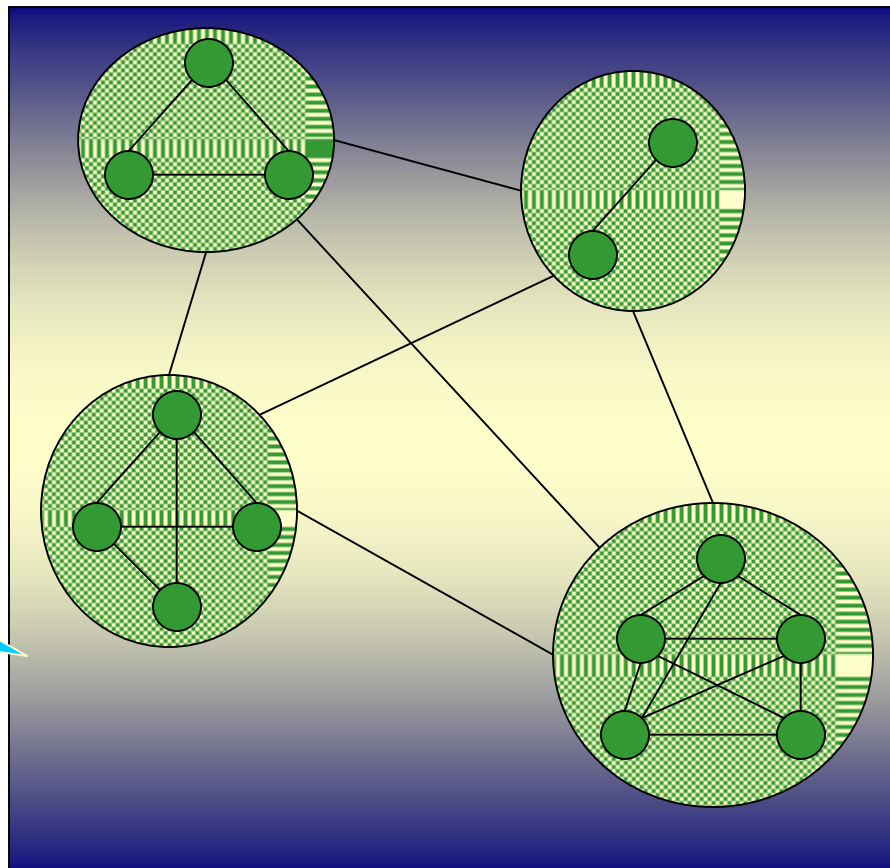
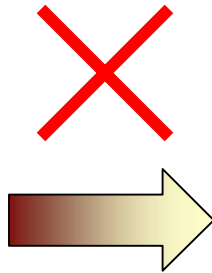
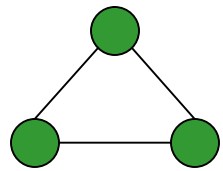


⑤ ” معمولاً سیستمهای پیچیده که به صورت محکم و استوار عمل می کنند حاصل تکامل سیستمهای ساده ای هستند که به درستی عمل می کردند. سیستمهای پیچیده که از ابتدا به صورت پیچیده طراحی می شوند، هرگز کار نخواهند کرد“

ویژگیهای سیستمهای پیچیده (ادامه)



مثال:

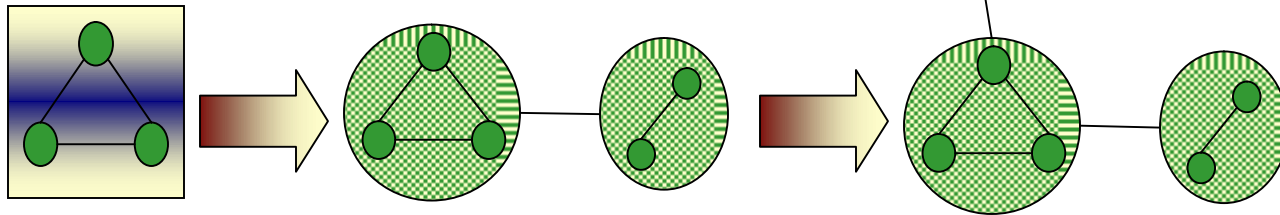


این سیستم کار نخواهد کرد

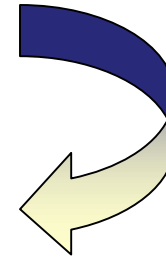
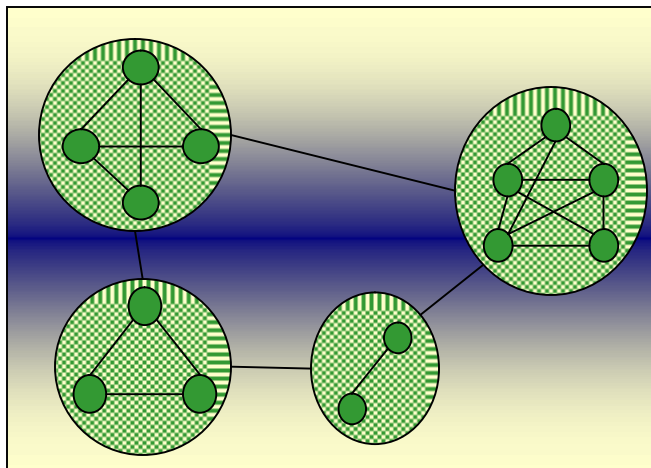
ویژگیهای سیستمهای پیچیده (ادامه)



این سیستم بقبوی
کار می کند



این سیستم نیز
بقبوی کار خواهد کرد



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

