

## آموزش مهندسی نرم افزار و UML

### جلسه اول:

اولین قدم در تولید و توسعه نرم افزار داشتن یک نگرش سیستمی به بسته یا محصول نرم افزاری می‌باشد. اما چرا ما باید نرم افزار را به عنوان یک سیستم در نظر بگیریم ؟  
جواب این سؤال را باید از تعریف تئوری سیستم و خصوصیات که یک سیستم دارا می‌باشد استخراج کنیم.

### تئوری سیستم‌ها

دانشی برای سهولت کار با سیستم‌ها و بررسی دقیق این مفهوم است ؛ در واقع تئوری سیستم‌ها روشی برای شناخت محیط اطراف یا روشی برای شناخت دنیای واقع می‌باشد .

از تعریف فوق می‌توان نتیجه گرفت :  
برنامه نویسان برای ساخت برنامه هایی که با نیاز کاربران همسو باشد ، نیاز به شناخت محیطی دارند که کاربران در آن فعالیت می‌کنند پس برای شناخت محیط باید با دید سیستمی به مسئله نگاه کرد.

### خصوصیات مهم سیستم :

1. محیط - Environment: هر سیستم در یک محیط قرار دارد.
2. مرز - Boundary : سیستم‌های موجود در یک محیط توسط مرزها از یکدیگر جدا می‌شوند.
3. ورودی و خروجی - I/O : هر سیستم ورودی هایی را از محیط می‌گیرد و خروجی هایی را به محیط پس می‌دهد.
4. واسط - Interface : امکان محاوره سیستم‌ها در یک محیط را فراهم می‌کند.
5. زیر سیستم - Sub System : هر سیستم می‌تواند حاوی چندین زیرسیستم باشد . زیر سیستم‌ها تمام خصوصیت‌های یک سیستم را دارا می‌باشند.
6. مکانیزم کنترلی - Controller : مهمترین بخش یک سیستم می‌باشد. مکانیزم کنترلی در واقع کنترل کننده تمامی فعالیت‌های انجام شده توسط یک سیستم است . ورودی‌ها از طریق مکانیزم کنترلی دریافت می‌شود و بر اساس آن خروجی هایی به محیط پس داده می‌شود.

### نتیجه گیری :

با توجه به خصوصیات که در مورد سیستم‌ها مطرح شد به راحتی می‌توانیم دلیل علاقه مندی برنامه -نویسان به نوع نگرش سیستمی را در یابیم ، و وجود محیط پیرامون یک سیستم و نحوه تبادل اطلاعات این سیستم با سایر سیستم‌ها در این محیط ، شکستن یک سیستم به چند زیر سیستم برای راحتی مسئله و پیاده سازی آسانتر آن و نیز وجود اینترفیس‌ها برای برقراری محاوره ای استاندارد بین زیر سیستم‌های یک سیستم و همچنین وجود ورودی هاو تصمیم گیری براساس ورودی هاو تولید یک خروجی همه و همه از نکات مورد توجه برنامه نویسان در تولید یک بسته نرم افزاری هستند که هماهنگی کاملی با مفاهیم تئوری سیستم‌ها دارند.

## نظرات خوانندگان

نویسنده: هاشم  
تاریخ: ۱۳۹۱/۰۴/۰۲ ۲:۰۶

سلام  
دوست عزیز با توجه به قدیمی بودن UML بهتره به سمت مباحثی نظیر اسکرام در Agile ، RUP رفت

نویسنده: علی قمشلویی  
تاریخ: ۱۳۹۱/۰۴/۰۲ ۱۰:۰۶

با سلام و تشکر  
UML و RUP دو مقوله جدا از یکدیگر می باشند در واقع RUP از UML استفاده میکند در ادامه مقالات تمامی این موارد را توضیح خواهم داد و شما هر چه بیشتر با مزایای استفاده از UML آشنا خواهید شد.

نویسنده: صالح  
تاریخ: ۱۳۹۱/۰۴/۰۲ ۲۰:۴۴

امید وارم این مطلب رو ادامه بدید.

نویسنده: رضا.ب  
تاریخ: ۱۳۹۱/۰۴/۲۱ ۱۰:۲۳

متدولوژی هایی که شما می گین (RUP یا متدهای چابک) اصلا ربطی به مفاهیم تحلیلی سیستم که در قالب UML ارائه میشه، نداره. بهتره سنجیده تر کامنت گذاشت. UML زبان مشترک مدل کردن سیستمه بین توسعه دهندگان و شاید دیگر ذی نفعان سیستم.

نویسنده: اژدری  
تاریخ: ۱۳۹۱/۰۶/۱۳ ۱۳:۲۳

دقیقا از نام uml مشخصه که فقط یک زبان مشترکه و ربطی به سایر مسائل نداره ، علت به وجود آمدنش هم ساده است ، برای اینکه هر کسی از ظن خودش یار قضیه نشه و خیال همه راحت باشه که میدونن دارن در مورد چی صحبت می کنن و داستان اون فیل در تاریکی هم نشود ، و در ضمن رساندن منظور با عکس خیلی بهتر از یک صفحه نوشته است