طبقه بندی(classification) :

در یک سیستم نرم افزاری انواع گوناگونی از اشیا قرار دارد. این اشیا در سطوح مختلفی از انتزاع قرار دارند.

مقوله بندی(stereotyping) و **لایه بندی**(layering) کمک می کند تا دسته بندی مناسبی برای اشیا ارایه دهیم.

برای دسته بندی و توصیف دیدگاههای مختلف دو اصل وجود دارد:

1) از دید چه کسی این سیستم توصیف می گردد؟ ==> لایه بندی
2) میزان پرداختن به جزییات چقدر است؟==> مقوله بندی

لایه بندی

در لایه بندی یک سیستم نرم افزاری به صورت تعدادی از لایه ها تقسیم بندی می گردد. هر لایه از تعدادی مولفه تشکیل شده که همکاری گروهی این مولفه ها بوجود آورنده رفتار لایه است.

استفاده از لایه بندی،وابستگی ها را کاهش می دهدبه طوریکه لایه های پایینتراز جزییات لایه های بالاتراطلاع ندارند.

معماری های متمرکز،معماریclient/server ،معماری 3-Tier نمونه هایی از لایه بندی در دنیای نرم افزارهستند.

لایه بندی

یک برنامه کاربردی از نظر منطقی به سه قسمت تقسیم می شود:

1) واسط کاربر(user interface)
2) منطق حرفه یا منطق برنامه(business logic)
3) سرویس های داده ای(data services)

تفاوت این معماری ها در نحوه ارتباط این قسمتها با یکدیگر می باشد.

معماری متمرکز

در معماری متمرکز این سه قسمت با هم ترکیب می شوندو باعث ساخت سیستمهای غیرقابل انعطاف و غیر قابل نگهداری می شود.

مزیت این روش،سادگی آن در طراحی و پیاده سازی است که در سالهای اولیه توسعه نرم افزار مورد استفاده قرار می گرفت.

معماری client/server

این معماری شامل دو لایه است:

لایه client و لایه server

واسط کاربرو منطق حرفه سرویس های داده ای

مزیت معماری client/server

- واسط کاربر قابل استفاده مجدد است.
- استفاده از مدل برنامه نویسیevent-driven
- وجود محیط های برنامه نویسی مانند .NET,Delphi,Visual Studio,...
- امکان استفاده مجدد از منطق حرفه در قسمت سرویس دهنده

معایب معماری client/server

- محل منطق حرفه مشخص نیست.
- تنها بخشی از منطق حرفه قابل استفاده مجدد است.
- منطق حرفه قابل استفاده مشخص،معمولا به صورت روالهای ذخیره شده که متعلق به یک پایگاه داده معینی است، وجود دارد.

معماری 3-Tier

در این معماری این لایه ها از هم جدا هستند. این ویژگی قابلیت استفاده مجدد در این سیستم ها را بالا می برد.

منطق حرفه بر روی سرویس دهنده مستقلی به نام سرویس دهنده کاربردی قرار می گیرد.

متدولوژی شی گرای perspective از این معماری استفاده می کند.

معماری perspective

در این معماری چهار لایه وجود دارد:

1) فرایندهای حرفه(business process) :فرایند حرفه عبارتست از مجموعه ای از فعالیت ها که یک یا چند ورودی دریافت کرده و خروجی را برای کاربر ایجاد می کنند.
2) سرویس های کاربر:فراهم نمودن سرویس های ارتباط با کاربر
3) سرویس های حرفه:سرویس های عمومی که منطق حرفه سیستم را تشکیل می دهند. این سرویس ها داده ها رااز سرویس های کاربر و سرویس های داده ای گرفته و مورد پردازش قرار می دهند.
4) سرویس های داده ای: این سرویس داده های موردنیازفرایندهای حرفه متفاوت را فراهم می نماید. سرویس های داده ای عمل پردازش داده ها رابه صورت مستقل از نحوه ذخیره سازی آنها انجام می دهند.

مقوله بندی

مقوله بندی اشیا کمک می کندکه سطوح انتزاعی را شناخته و به این ترتیب تمام افراد تیم در یک سطح از انتزاع فعالیت کنند.

مثال: در متدولوژیUSDP کلاس ها به سه گروه تقسیم می شوند:

1) کلاسهای مرزی
2) کلاسهای کنترلی
3) کلاسهای موجودیتی

مقوله بندی

مثال: در متدولوژیperspective سه نوع کلاس وجود دارد:

1) اشیا واسط/کاربری:برقراری ارتباط کاربربا سیستم
2) اشیا حرفه:سرویس های مورد احتیاج نیازمندیهای حرفه
3) اشیا داده ای:فراهم نمودن سرویس های داده ای

فرایند توسعه نرم افزار:

مجموعه ای از فعالیت های نیمه مرتبی که برای توسعه نرم افزار به کار گرفته می شوند را می توان تعریف ساده ای از فرایند توسعه نرم افزار دانست.

نقش های اساسی در فرایند توسعه نرم افزار:

1) مشخص نمودن ترتیب فعالیت ها
2) چه فرایندهایی باید تولید شوند و در چه زمانی
3) تعیین روش اداره وظایف توسعه دهندگان منفرد و تیمی
4) معیارهایی برای اندازه گیری کیفیت محصولات پروژه

مشخصات فرایندهای توسعه موفق:

- استفاده از روش تکرارو توسعه تدریجی
- مدیریت نیازمندیها
- استفاده از معماری مبتنی بر مولفه ها
- راهبری بر مبنای موارد کاربری
- مدلسازی تصویری نرم افزار
- کنترل تغییرات
- بررسی کیفیت نرم افزار