

مهندسی نرم افزار

مدرس: مسعود معاونی

منبع تدریس: کتاب مهندسی نرم افزار مهندس عین اله جعفرنژاد قمی و کتاب
مهندسی نرم افزار سجاد نقدعلی زنجانی

سرفصل‌های درس

- فصل اول: معرفی و آشنایی با ماهیت نرم‌افزار
- فصل دوم: فرایند
- فصل سوم: مهندسی سیستم و تحلیل نیازهای نرم‌افزار
- فصل چهارم: مدل‌های تحلیل
- فصل پنجم: مفاهیم طراحی
- فصل ششم: مدل‌های طراحی
- فصل هفتم: تعاریف و مفاهیم شی‌گرایی
- فصل هشتم: تحلیل و طراحی شی‌گرا
- فصل نهم: تعاریف و روش آزمون نرم‌افزار
- فصل دهم: راهبردهای آزمون نرم‌افزار
- فصل یازدهم: مدیریت پروژه و ارائه متریک‌های نرم‌افزاری
- فصل دوازدهم: تخمین و زمانبندی پروژه
- فصل سیزدهم: مباحث پیشرفته در مدیریت نرم‌افزار
- فصل چهاردهم: متدولوژی‌های نرم‌افزار

فصل پنجم: مفاهیم طراحی

- پس از آن که نیازهای نرم افزار شناخته شد، باید برای پیاده سازی نرم افزار آماده شد. بدیهی است برای داشتن نرم افزاری قدرتمند باید مقدمات طراحی را در نظر بگیریم..
- طراحی نرم افزار دقیقا جایی است که همزمان بر روی دو دنیا ایستاده اید: دنیای فناوری و دنیا اهداف مشتریان!
و مهندس نرم افزار تلاش می کند این دو دنیا را به هم نزدیک کند.
- طراحی نرم افزار به چهار مدل طراحی داده، طراحی معماری، طراحی واسط کاربری و طراحی در سطح اجزا تقسیم می شود.

ابعاد طراحی

- **فاز طراحی داده (Data Design):** داده ها همواره به عنوان بخش مهمی از نرم افزار محسوب می شوند.. طراحی داده به معنای انتخاب یک ساختمان داده مناسب است. همچنین طراحی ساختار پایگاه داده براساس نمودار ERD نیز در همین مرحله معنا می یابد.
- **طراحی معماری (Architecture Design):** هر نرم افزاری از بخش های کوچکتری به نام مولفه (Component) تشکیل می شود. در طراحی معماری، تمرکز بر روی ساختار و رابطه بین این مولفه ها است. سبک های مختلف معماری نرم افزار نقش مهمی در طراحی براساس محدودیت ها دارند.
- **طراحی رابط (Interface Design):** به طور عمومی نرم افزارها با عوامل خارجی مثل سخت افزارها، دیگر نرم افزارها و کاربران در ارتباطند. هدف اصلی از طراحی رابط کاربری، ارتقا سطح کیفی ارتباطات نرم افزار با محیط اطراف و خصوصا با کاربر است.
- **طراحی در سطح اجزا (Component Level Design):** در این مرحله به دنبال ساخت اجزا و تعیین رویه های ارتباطی بین مولفه ها هستیم تا عملکرد درستی در نرم افزار ایجاد شود.

اهداف و اصول طراحی

- طراحی باید تمام نیازهای صریح نرم افزار را در بر گرفته و بتواند نیازهای ضمنی آن را نیز پوشش دهد.
- طراحی باید قابل درک و خوانا باشد، تا هم بخش پیاده سازی و هم بخش آزمون و پشتیبانی بتوانند از آن به بهترین نحو استفاده کنند.
- طراحی باید نرم افزار را به شکل کامل به تصویر بکشد.
- طراحی باید پیمانه ای (ماژولار) باشد و هر بخش بزرگ به چند بخش کوچکتر تقسیم شود.
- طراحی باید هم زمان در هر چهار بعد داده، معماری، واسط و اجزا توسعه یابد.

عوامل کیفی موثر در طراحی نرم افزار

- قابلیت عملیاتی
- قابلیت استفاده
- قابلیت اطمینان
- کارایی
- قابلیت پشتیبانی

مفاهیم طراحی

- زمانی که یک خودرو که یک خودرو ساخته می‌شود، حرکت کردن آن لزوماً به معنای ساخت صحیح خودرو نیست.
در نرم‌افزار نیز چنین است.
- **اگر نرم‌افزاری سرویس‌های کاربر را به درستی برآورده کند، باز هم نمی‌توان ادعا کرد که این محصول درست ساخته شده است.**
- بنابراین باید برای طراحی نرم‌افزار به مفاهیمی مانند تجرید، پالایش، پیمانه‌سازی، معماری نرم‌افزار، سلسه مراتب، ساختمان داده، پنهان‌سازی اطلاعات، استقلال عملکردی، انسجام و اتصال توجه کرد.

۱. تجرید یا abstraction

- در تجرید تلاش می‌شود تا از سطوح کلی به جزئی برسیم. به عنوان مثال وقتی می‌گوییم درب را باز کن، یک جمله تجریدی است. هر چند که در آن جزییات را نگفته‌ایم، اما بسیاری از مفاهیم را منتقل کرده‌ایم. در مثال بالا منظور ما این بود که بلندشو، به سمت درب حرکت کن، دستگیره درب را به پایین فشار بده و درب را به سمت خودت بکش.
- مفهوم تجرید این امکان را ایجاد می‌کند که شخص در سطح کلی و بدون توجه به جزییات نامرتبط سطح پایین، روی مسئله تمرکز کند.

۲- پالایش Refinement

- درک کلی مسئله همیشه به سادگی امکان پذیر نیست. در پالایش با تعیین ریز عملیات، جزئیات مسئله آشکار می شود. در پالایش هر چه از سطح بالا به پایین حرکت می کنیم با جزئیات بیشتری از مسئله روبه رو می شویم.
- پالایش و تجرید دو مفهوم مکمل یکدیگر هستند. تجرید بر عملکرد کلی تاکید دارد و پالایش بر روی جزئیات متمرکز است.

۳. پیمانه‌سازی Modularity

- پیمانه‌سازی براساس تکنیک تقسیم شکل می‌گیرد. پیمانه‌ها یا ماژول‌ها اجزایی هستند که وظایف مشخصی را انجام می‌دهند و در ارتباط با سایر اجزا در سیستم نقش مکملی را برعهده دارند. در طراحی‌های ماژولار، امکان استفاده از قطعات طراحی شده در پروژه‌های مشابه وجود دارد.

۴. معماری نرم افزار

- معماری نرم افزار نشان می دهد که اجزا و پیمانه ها تحت چه ساختاری باید با یکدیگر مجتمع شوند تا نرم افزار مورد نظر ساخته شده و عملکرد درستی داشته باشد.. در واقع در معماری نرم افزار، نحوه اتصال و ادغام ماژول ها با یکدیگر را بررسی می کنیم.

۵. سلسله مراتب کنترل (Control Hierachy)

- سلسله مراتب کنترل که به آن ساختار برنامه یا Program Structure نیز گفته می‌شود، ترتیب قراگیری پیمانه‌ها و سلسله مراتب کنترل را نشان می‌دهد. در سلسله مراتب کنترل از ساختارهای بازگشتی و درخت‌های تصمیم‌گیری استفاده می‌شود.

۶. ساختمان داده

- ساختار داده‌ها همیشه بر طراحی نهایی نرم‌افزار تاثیر می‌گذارد.. در ساختار داده‌ها با مفاهیمی مانند آرایه، لیست پیوندی، پشته، انواع درخت و... سروکار داریم. به‌عنوان مثال در ساختار تجرید داده، پشته محلی برای ایجاد صف و نگهداری اطلاعات محسوب می‌شود، اما در لایه‌های پایینی از آرایه برای پیاده‌سازی استفاده می‌کنیم.

۷. پنهان سازی اطلاعات

- یکی از جذاب ترین مفاهیم طراحی، پنهان سازی اطلاعات است. در حقیقت تعیین سطوح دسترسی و نمایش اطلاعات به کاربران در این گام تعیین می شود. در مهندسی نرم افزار یک اصل کلیدی داریم **به هر کسی همان قدر اطلاعات نشان بده که به آن نیاز دارد!**

۸. استقلال عملکردی

- هر پیمانه باید تا جای ممکن تنها یک عمل واحد انجام دهد و تا جای ممکن از ارتباط با پیمانه‌های دیگر یا تاثیرگذاری بر روی آنان جلوگیری کند. استقلال عملکردی بر کاهش ارتباط بین ماژول‌ها تاکید دارد و کاهش ارتباط بین ماژول‌ها یعنی کاهش احتمال انتشار خطا و تغییرات. همچنین امکان استفاده یک ماژول در برنامه‌های دیگر را داریم.

۹. انسجام و اتصال

- از طراحی ماژول‌هایی اجتناب می‌کنیم که چند کار غیر مرتبط را انجام می‌دهند.
- اتصال وابستگی بین ماژول‌ها برای رسیدن به اهداف‌شان را نشان می‌دهد. اتصال نقطه مقابل استقلال عملکردی است. اما باید توجه داشت برخی از ماژول‌ها برای عملکرد بهتر نیاز به دریافت داده‌هایی از ماژول‌های دیگر دارند.

سوالات پایان فصل ۵

- ابعاد طراحی را نام برده و هر کدام را شرح دهید
- اهداف طراحی نرم افزار چیست
- چه عواملی بر روی کیفیت نرم افزار اثر می گذارند
- تجرید چیست و چرا اهمیت دارد
- منظور از پیمانه سازی را بیان کرده و مزیت کلیدی آن را بگویید
- چرا در طراحی نرم افزار به سراغ پنهان سازی اطلاعات میرویم