

مهندسی نرم افزار

فصل چهارم : اصول راهنما در مهندسی نرم افزار

مدرس:

فرشید شیرافکن

دانشجوی دکتری دانشگاه تهران

(کارشناسی و کارشناسی ارشد : کامپیوتر نرم افزار) (دکتری: بیو انفورماتیک)

نگاهی گذار

اصول راهنما چیستند؟

مهندسی نرم افزار، آرایه وسیعی از اصول، مفاهیم، روش ها و ابزارهاست که باید در برنامه ریزی برای توسعه یک نرم افزار در نظر گرفت.

چرا اهمیت دارد؟

فرایند نرم افزار برای رسیدن به هدفی موفق، یک نقشه راه در اختیار همه ی افراد دخیل در ایجاد یک سیستم یا محصول کامپیوتری قرار می دهد. کار مهندسی، جزئیات لازم برای طی این مسیر را برای شما فراهم می سازد. در حیطه ی مهندسی نرم افزار، کار مهندسی چیزی است که در طول روز انجام می دهید تا نرم افزار را از یک ایده به واقعیت برسانید.

کار مهندسی نرم افزار چیست؟

از یک دیدگاه کلی، کار مهندسی به مجموعه ای مفاهیم، اصول، روش ها و ابزارها گفته می شود که یک مهندس نرم افزار در طول روز با آنها سرو کار دارد.

کار مهندسی به مدیران این امکان را می دهد که پروژه های نرم افزاری را مدیریت کنند و به مهندسان نرم افزار این امکان را می دهد که برنامه های کامپیوتری بسازند.

دانش مهندسی نرم افزار

نظر مک کانل:

توده ی دانش مهندسی نرم افزار به یک هسته ی پایدار متکامل شده است که نشان گر ۷۵٪ از دانش مورد نیاز برای توسعه یک سیستم پیچیده است.

اصول هسته ای اکنون بستری فراهم می سازند که مدل های نرم افزاری، روش ها و ابزار ها را در آن بستر می توان به کاربرد و ارزیابی کرد.

اصول هسته ای

مجموعه ای از اصول هسته ای وجود دارد که راهنمای مهندسی نرم افزار است و به استفاده از یک فرایند نرم افزار با معنی و اجرای اثربخش روش های مهندسی نرم افزار کمک می کند.

در سطح **فرایند**، اصول فرایند، یک بنیاد فلسفی ایجاد می کنند که :

- تیم نرم افزاری را هدایت می کند،
- جریان فرایند را مورد کاوش قرار می دهد و
- مجموعه ای از محصولات کاری مهندسی نرم افزار تولید می کند.

در سطح **کاری**، اصول مهندسی نرم افزار، مجموعه ای از ارزش ها و قواعد را تعیین می کند که شما را در تحلیل یک مساله، طراحی راهکار، پیاده سازی و آزمون آن راهکار و سرانجام استقرار نرم افزار در جامعه ی کاربری راهنمایی می کند.

اصول راهنمای فرایند مهندسی

مجموعه اصول هسته ای زیر را می توان برای چار چوب و با بسط دادن آن برای هر فرایند نرم افزار به کار گرفت:

اصل ۱. چابک باشید.

اصل ۲. در هر مرحله کیفیت را در کانون توجه قرار دهید.

اصل ۳. آمادگی انطباق را داشته باشید.

اصل ۴. تیمی اثر بخش تشکیل دهید.

اصل ۵. سازو کارهایی برای برقراری ارتباط و هماهنگی ایجاد کنید.

اصل ۶. مدیریت تغییرات.

اصل ۷. ارزیابی ریسک.

اصل ۸. ایجاد محصولات کاری که برای دیگران ارزش فراهم می کنند.

اصول راهنمای کار مهندسی نرم افزار

۱. اصل تقسیم و حل
۲. اصل درک به کارگیری انتزاع
۳. اصل تلاش برای سازگاری
۴. اصل توجه ویژه به انتقال اطلاعات
۵. اصل توسعه نرم افزاری که ساختار پیمانه ای اثر بخش داشته باشد.
۶. اصل جستجو به دنبال الگوها
۷. اصل هرگاه که امکان دارد، مساله و راهکار آن را از چند دیدگاه متفاوت به نمایش بگذارید.
۸. اصل به خاطر داشته باشید که نرم افزار را نگهداری خواهید کرد.

اصول راهنمای فعالیت های چارچوبی

۱- اصول ارتباطی

۲- اصول برنامه ریزی

۳- اصول مدل سازی

۴- اصول ساخت

۵- اصول استقرار

اصول ارتباطی

اصول ارتباطی : بسیاری از این اصول در سایر شکل‌های ارتباطی که در یک پروژه ی نرم افزاری رخ می دهند کاربرد دارند:

اصل ۱. گوش سپردن.

اصل ۲. خود را قبل از برقراری ارتباط آماده کنید.

اصل ۳. یک نفر باید این فعالیت را تسهیل کند.

اصل ۴. بهترین راه، ارتباط رودرروی است.

اصل ۵. یادداشت بردارید و تصمیم گیری را مستند کنید.

اصل ۶. تلاش برای همکاری.

اصل ۷. توجه خود را معطوف کنید، بحث خود را پیمان‌ه ای کنید.

اصل ۸. اگر چیزی واضح نبود، یک تصویر بکشید.

اصل ۹. الف) هنگامی که بر سر بحثی به توافق رسیدید، به مبحث دیگر بپردازید.

ب) اگر به توافق نرسیدید، به مبحث دیگر بپردازید.

پ) اگر ویژگی یا قابلیت‌ای واضح نیست و نمی‌توان در حال حاضر آن را واضح کرد، باز هم به مبحث دیگر بپردازید.

اصل ۱۰. مذاکره، یک مسابقه یا بازی نیست. وقتی بهترین نتیجه را می‌دهد که هر دو طرف برنده باشند.

اصول برنامه ریزی

فعالیت برنامه ریزی شامل مجموعه ای از امور مدیریتی و فنی می شود که تیم نرم افزاری را قادر به تعریف نقشه راه در سفر به سوی اهداف راهبردی و مقاصد تاکتیکی اش می سازد.

فلسفه های متفاوت برای برنامه ریزی:

- **کمینہ گرا** : تغییر، غالباً نیاز به برنامه ریزی مفصل را منتفی می سازد.
- **سنت گرا** : برنامه ریزی، یک نقشه راه اثربخش فراهم می آورد و هرچه جزئیات آن بیشتر باشد، احتمال گم شدن تیم کمتر می شود.
- **چابک گرایان** : یک "بازی برنامه ریزی" سریع ممکن است ضروری باشد، ولی نقشه راه چیزی است که با "کار واقعی" روی نرم افزار شروع می شود.

برنامه ریزی با هر میزان سخت گیری که اجرا شود، اصولی که به دنبال خواهد آمد، همواره کاربرد خواهد داشت:

اصل ۱. شناخت حوزه ی پروژه.

اصل ۲. طرف های ذی نفع را در فعالیت برنامه ریزی دخالت دهید.

اصل ۳. این را بدانید که برنامه ریزی ماهیتی مبتنی بر تکرار دارد.

اصل ۴. برآوردهای خود را براساس آنچه که می دانید، انجام دهید.

اصل ۵. همزمان با برنامه ریزی، ریسک را هم در نظر بگیرید.

اصل ۶. واقع بین باشید. مردم هر روز صددرصد کار نمی کنند.

اصل ۷. هنگام تعریف برنامه ریزی، گرانولیتة (granularity) را تعیین کنید.

اصل ۸. تعیین کنید که چگونه می خواهید از کیفیت اطمینان یابید.

اصل ۹. چگونگی انجام دادن تغییرات را شرح دهید.

اصل ۱۰. برنامه ریزی را به وفور پیگیری کنید و در صورت نیاز، تنظیماتی به عمل آورید.

اصول مدل سازی

ما مدل ها را برای درک بهتر یک موجودیت واقعی که قرار است ساخته شود، ایجاد می کنیم.

مدل ها با توجه به نوع موجودیت متفاوت اند.

در کار مهندسی نرم افزار، دو نوع مدل ایجاد می شود:

- ۱- مدل های خواسته ها (مدل تحلیل)
- ۲- مدل های طراحی

مدل خواسته ها، خواسته های مشتری را با تصویر کردن نرم افزار در سه دامنه متفاوت به نمایش می گذارند.

۱- دامنه اطلاعاتی

۲- دامنه عملیاتی

۳- دامنه رفتاری

مدل های طراحی: نشان گر خصوصیتی از نرم افزارند که به نرم افزار نویس کمک می کنند تا آن را بهتر بسازد:

۱- معماری

۲- واسط کاربر

۳- جزئیات در سطح مولفه ها

اصول مدل سازی

اصل ۱ - هدف اصلی تیم نرم افزاری ساخت نرم افزار است نه ایجاد مدل .

اصل ۲ - سبک بار سفر کنید. مدل های بیش از نیاز خود ایجاد نکنید.

اصل ۳ - بکوشید ساده ترین مدلی را بسازید که مساله یا نرم افزار را توصیف کند.

اصل ۴ - مدل ها را طوری بسازید که قابل تغییر باشد.

اصل ۵ - توانایی بیان صریح هدف هر مدل ایجاد شده را داشته باشید.

اصل ۶- مدل هایی را که توسعه می دهید بر سیستم مورد نظر مطابقت دهید.

اصل ۷- سعی کنید مدل های مفید بسازید، ولی ساخت مدل های کامل را فراموش کنید .

اصل ۸- در مورد قالب و نحو مدل، تعصب به خرج ندهید، اگر در انتقال مفاهیم موفق است، نمایش در مرحله ی دوم اهمیت قرار دارد.

اصل ۹- اگر گزینه شما می گوید مدلی درست نیست، هر چند که روی کاغذ درست به نظر می رسد، احتمالاً دلیلی برای این نگرانی دارید.

اصل ۱۰- به محض اینکه توانستید ، باز خورد بگیرید.

در مدل سازی خواسته ها(تحلیل) ، ویژگی های نرم افزاری زیر مورد توجه قرار می گیرند :

۱- اطلاعاتی که باید پردازش شوند.

۲- قابلیت هایی که باید تحویل شوند.

۳- رفتاری که باید به نمایش گذارده شود.

اصول مدل سازی خواسته ها

اصل ۱ - دامنه ی اطلاعاتی یک مساله باید نمایش داده و درک شود.

اصل ۲- عملکردهای نرم افزار باید تعریف شوند.

اصل ۳- رفتار نرم افزار(بعنوان نتیجه ای از رویداد های خارجی) باید نمایش داده شوند.

اصل ۴- مدل هایی که اطلاعات، قابلیت ها و رفتارها را تصویر می کنند باید به شیوه ای تقسیم بندی

شوند که جزئیات را به گونه ای لایه ای(یا سلسله مراتبی) نمایش دهند.

اصل ۵- وظیفه ی تحلیل باید از اطلاعات ضروری به سمت جزئیات پیاده سازی حرکت کند.

اصول مدل سازی طراحی

اصل ۱ - طراحی باید تا مدل خواسته ها قابل ردگیری باشد.

اصل ۲ - همواره معماری سیستمی را که قرار است ساخته شود، در نظر داشته باشید.

اصل ۳ - طراحی داده ها به اندازه طراحی عملکردها اهمیت دارد .

اصل ۴ - واسط ها(چه درونی و چه بیرونی) باید با احتیاط طراحی شوند.

اصل ۵ - طراحی واسط کاربر باید مطابق با نیاز های کاربر نهایی تنظیم گردد، ولی در هر مورد باید بر

سهولت کاربرد نیز تاکید ورزیده شود.

اصل ۶- طراحی در سطح مولفه ها باید مستقل از عملکرد باشد.

اصل ۷- مولفه ها باید با یکدیگر و با محیط خارجی ارتباطی سست داشته باشند .

اصل ۸- مدل های طراحی باید به آسانی قابل درک باشند.

اصل ۹- طراحی باید به صورت تکراری توسعه یابد. در هر دور تکرار، طراحی باید بکوشد تا سادگی بیشتر

شود.

اصول ساخت

فعالیت ساخت شامل مجموعه ای از وظایف کدنویسی و آزمایش است که نتیجه ی آن نرم افزاری عملیاتی و آماده ی تحویل به کاربر نهایی است.

کدنویسی می تواند یکی از موارد زیر باشد

۱- ایجاد مستقیم کد منبع در زبان برنامه نویسی.

۲- تولید خودکار کد منبع با استفاده از نمایش شبه طراحی از مؤلفه ای باشد که قرار است ساخته باشد

۳- تولید خودکار کد قابل اجرا با استفاده از یک زبان برنامه نویسی نسل چهارم

در مرحله ی آزمون، توجه به سیستم ابتدا در سطح مؤلفه ها رخ می دهد، این رویکرد را آزمون واحدها می نامند.

سایر سطوح آزمون عبارتند از:

۱- آزمون انسجام

۲- آزمون اعتبارسنجی

۳- آزمون پذیرش

اصول کد نویسی

اصولی راهگشا در وظیفه کدنویسی ، بستگی تنگاتنگی با موارد زیر دارند:

- سبک برنامه نویسی
- زبان برنامه نویسی
- شیوه برنامه نویسی موردنظر

اصول آماده سازی

قبل از نوشتن حتی یک خط از برنامه اطمینان حاصل کنید که :

- می دانید چه مسأله ای را قرار است حل کنید.
- اصول و مفاهیم طراحی پایه را می دانید.
- زبانی برای برنامه نویسی انتخاب کنید که نیازهای نرم افزاری که قرار است در آن کار کند، برآورده سازد.
- محیطی برای برنامه نویسی انتخاب کنید که ابزارهای لازم برای آسان تر کردن کار را در اختیار قرار دهد.
- مجموعه ای از آزمون های پایه را ایجاد کنید که با کامل شدن کدنویسی مؤلفه، بتوانید آنها را به کار ببرید.

اصول برنامه نویسی

با شروع کدنوسی اطمینان حاصل کنید که:

- ۱- الگوریتم هایتان را با دنباله روی از برنامه سازی ساخت یافته، مقید کنید.
- ۲- استفاده از برنامه نویسی جفتی را در نظر داشته باشید.
- ۳- انتخاب ساختمان داده هایی که نیازهای طراحی را برآورده کند.
- ۴- شناختن معماری نرم افزار و ساختن واسط هایی سازگار با آن
- ۵- ساده نگه داشتن منطق شرطی تا حد امکان
- ۶- نوشتن حلقه های تودرتو به شیوه ای که به آسانی قابل آزمون باشند.
- ۷- انتخاب نام های با معنی برای متغیرها و پیروی از سایر استانداردهای کدنویسی محلی
- ۸- نوشتن کدهایی که خود مستندسازی شده باشند.
- ۹- ایجاد یک چیدمان بصری (مثلا با تورفتگی و خطوط خالی) که به فهم کدها کمک کند.

اصول اعتبارسنجی

پس از به پایان رساندن اولین دور کدنویسی حتما:

- در صورت امکان گشتی در میان کدها بزنید.
- آزمون واحدها را اجرا کنید و خطاهایی را که کشف می شوند را تصحیح کنید.
- کدها را بازآرایی کنید.

اصول آزمون

چند قاعده که می توان آن ها را به عنوان اهداف آزمون در نظر گرفت:

- آزمون، فرآیند اجرای برنامه به قصد یافتن خطاهاست.
- یک مورد آزمون خوب باید خطاهای کشف نشده را با احتمال زیادی کشف کند.
- آزمون موفق، آزمونی است که خطای کشف نشده تاکنون را کشف کند .

آزمون نمی تواند نبودن خطاها و نقایص را نشان بدهد، فقط می تواند نشان دهد که خطاها و نقایص وجود دارند.

اصول آزمون

اصل ۱: همه آزمون ها تا خواسته های مشتری قابل ردگیری باشند.

اصل ۲: آزمون ها را باید مدت ها قبل از شروع آزمون برنامه ریزی کرد.

اصل ۳: اصل پارتو در آزمون نرم افزار کاربرد دارد.

اصل پارتو : اثر ۸۰٪ از همه خطاهای کشف شده طی آزمون را احتمالاً در ۲۰٪ از کل مؤلفه های نرم افزار می توان پیدا کرد.

اصل ۴: آزمون باید در مقیاس کوچک آغاز شود و به سمت مقیاس بزرگ پیش رود.

اصل ۵: آزمون کامل امکان پذیر نیست.

اصول استقرار

فعالیت استقرار شامل سه کنش است: تحویل، پشتیبانی و بازخورد.

چون مدل های فرآیند نرم افزار مدرن، ماهیتی تکاملی یا افزایشی دارند، به یکباره استقرار رخ نمی دهد بلکه با حرکت نرم افزار به سوی تکامل، چند با تکرار می شود.

چرخه تحویل: یک نسخه عملیاتی از نرم افزار در اختیار کاربر نهایی قرار می دهد که قابلیت ها و ویژگی های جدیدی فراهم می سازد.

چرخه پشتیبانی: کمک انسانی و مستند سازی برای کلیه قابلیت ها و ویژگی های ارائه شده طی همه چرخه های استقرار تا آن زمان را فراهم می سازد.

چرخه بازخورد: راهنمایی های مهمی را در اختیار تیم نرم افزاری قرار می دهد که به اصلاح قابلیت ها، ویژگی ها و رویکرد در نظر گرفته شده برای نسخه بعدی نرم افزار می انجامند.

در حالی که تیم نرم افزاری آماده تحویل یک نسخه جدید می شود چند اصل کلیدی را باید رعایت کند:

اصل ۱: انتظارات مشتری برای نرم افزار باید مدیریت شود .

اصل ۲: پکیج تحویل کامل باید مونتاژ و آزمایش شود.

اصل ۳: قبل از تحویل نرم افزار، یک روال پشتیبانی باید مشخص کرد.

اصل ۴: مواد آموزشی مناسب باید برای کاربران نهایی تهیه شود.

اصل ۵: نرم افزار مشکل دار ابتدا باید اصلاح و بعدا تحویل داده شود.

اشتباه در برقراری ارتباط

SAFEHOME



team workspace

Communication Mistakes

The scene: Software engineering

The players: Jamie Lazar, software team member;
Vinod Raman, software team member; Ed Robbins,
software team member.



پایان فصل ۴