

اصول مهندسی نرم افزار و کاربرد آن در مکانیزاسیون سیستم های لجستیکی

تهیه و تدوین: مهندس علی کریمی

عضو هیأت علمی دانشگاه امام حسین (ع) و مدیر انفورماتیک مرکز مطالعات و پژوهش های لجستیکی

واژه های کلیدی: مهندسی نرم افزار - کیفیت نرم افزار - تولید نرم افزار - نگهداری نرم افزاری - زمان تولید نرم افزار - هزینه تولید نرم افزار - هزینه نگهداری نرم افزار - ابزارهای مهندسی نرم افزار - تست نرم افزار - مجتمع سازی نرم افزار - دوره حیات نرم افزار - قابلیت استفاده مجدد نرم افزار

چکیده

با توجه به پیشرفت های شگرف در زمینه مهندسی نرم افزار و فنون نوین آن در تولید و توسعه سیستم های نرم افزاری، شناخت اصول، مبانی و بکارگیری روش های پیشرفته آن در تجزیه و تحلیل و طراحی سیستم، می تواند به افزایش غنای سیستم و محکم شدن پایه های علمی آن بیفزاید. نقش اساسی و راهبردی مهندسی نرم افزار در فرآیندهای تولید، توسعه و نگهداری سیستم های نرم افزاری، کاهش «زمان» و «هزینه» می باشد که از عوامل مهم و تعیین کننده در فرآیندهای سه گانه یاد شده به شمار می آید.

۱- مقدمه

آمار تجربه های نرم افزاری سال های اخیر نتایج زیر را در اختیار ما قرار می دهد.
- سالانه بیش از سی میلیارد دلار صرف تولید سیستم های نرم افزاری می شود.
- بیش از ۶۰٪ هزینه های نرم افزار مربوط به دوره نگهداشت آن است.
- بیش از ۶۵٪ سیستم های نرم افزاری با شکست مواجه می شوند و به بهره برداری کامل نمی رسند.
- حدود ۲٪ از نرم افزارها به بهره برداری کامل می رسند.
- حدود ۳٪ از نرم افزارها پس از اصلاحات لازم به بهره برداری می رسند.
- حدود ۴۵٪ از نرم افزارها برنامه نویسی می شوند، ولی به مرحله اجرا و عملیات نمی رسند.
- حدود ۳۰٪ از نرم افزارها آنالیز و طراحی می شوند، ولی به مرحله کدینگ نمی رسند.
آمارهای فوق لزوم توجه به اصول و فنون مهندسی نرم افزار را تبیین می کند و ما را به این نتیجه می رساند که در فرآیند تولید و توسعه سیستم های نرم افزاری باید از ابزارهای مطمئن و قابل اتکا بهره بجوییم.
در این شرایط فنون مهندسی نرم افزار به کمک ما می شتابند و یک دید نظام مند مبتنی بر پایه های علمی را در اختیار ما قرار می دهند.

۲- چرا مهندسی نرم افزار؟

دلایل زیر لزوم توجه به مبانی مهندسی نرم افزار را توجیه می کند.
افزایش روزافزون هزینه تولید و نگهداری سیستم های نرم افزاری
تغییر و تبدیل برنامه زمان بندی پروژه های نرم افزاری
نامطلوب بودن کیفیت نرم افزار تولید شده (از نظر قابلیت اطمینان)
تجدید نظر در روش های برنامه ریزی و کنترل پروژه های نرم افزاری
استفاده از روش های جدید در تولید نرم افزار
مهندسی نرم افزار یک دید نظام مند و همه جانبه نسبت به کل فرآیند تولید، توسعه و نگهداری سیستم های نرم افزاری ارائه می کند.

تهیه و جایگزینی استانداردهای مطمئن به جای استفاده از روش های غیراستاندارد.

۳- دلایل روند رو به رشد هزینه های نرم افزار

نرم افزار یک محصول تک منظوره است، لذا برای هر کاربرد جدید، نیاز به تهیه نرم افزار جدید وجود دارد ولی در مقابل، سخت افزار یک محصول همه منظوره است. البته با استفاده از مفاهیم مبنایی مهندسی نرم افزار، مانند بخش پذیری ۱ و قابلیت استفاده مجدد، ۲ می توان از نرم افزارهای تولید شده قبلی، در پروژه های جدید با اعمال حداقل تغییرات استفاده نمود و این مسئله در کاهش هزینه و زمان تولید نرم افزار بسیار مؤثر است.

طراحی و تولید سخت‌افزارهای جدید تولید نرم‌افزارهای جدید را افزایش می‌دهد. کاربردهای جدید، بزرگ‌تر و پیچیده‌تر می‌شود، لذا تولید نرم‌افزار برای آنها پیچیده‌تر و نیاز به افراد خبره و با تجربه بیشتر می‌شود.

مجموعه پیشرفت‌های فناوری در زمینه شبکه‌های رایانه‌ای ۳، سیستم‌های توزیع شده ۴، سیستم‌های جاسازی شده ۵ مثل ROM، ابزارهای تولید خودکار نرم‌افزار ۶، سیستم‌های خبره ۷ و شبکه‌های عصبی ۸ باعث تشدید بحران نرم‌افزار گردیده است.

۴- تعریف مهندسی نرم‌افزار

تعاریف مختلفی برای مهندسی نرم‌افزار ارائه شده است که برخی از آنها را بر می‌شماریم. مهندسی نرم‌افزار یک دید نظام‌مند و همه جانبه نسبت به کل فرآیند تولید، توسعه و نگهداری سیستم‌های نرم‌افزاری ارائه می‌کند.

علمی است که ایده تحلیل، طراحی، کدگذاری، آزمایش، مستندسازی و مدیریت را برای تکمیل موفقیت‌آمیز سیستم‌های بزرگ نرم‌افزاری به کار می‌گیرد.

علمی است که تولید سیستم‌های نرم‌افزاری اقتصادی و قابل اطمینان که به طور مؤثر روی ماشین واقعی کار می‌کند را امکان‌پذیر می‌سازد.

مهندسی نرم‌افزار کاربرد مجموعه‌ای از اصول برای بهینه‌سازی فرآیند تولید و نگهداشت نرم‌افزار با هدف کاهش هزینه و افزایش قابلیت اطمینان است.

۵- دلایل بحران نرم‌افزار

برخی از عوامل مؤثر در بحران سیستم‌های نرم‌افزاری را می‌توان به ترتیب زیر برشمرد.

هزینه روز افزون تولید سیستم‌های نرم‌افزاری

تأخیر در تولید و تحویل به موقع نرم‌افزار به کارفرما

عدم تأمین نیازهای واقعی کارفرما

کیفیت پایین، نامطمئن بودن و کارآیی کم نرم‌افزار

هزینه بالای توسعه و نگهداری نرم‌افزار [۲]

۶- شناخت بهتر ماهیت و طبیعت نرم‌افزار

با دقت در ماهیت نرم‌افزار می‌توان دریافت که تفاوت عمده در ابعاد مختلف بین نرم‌افزار و سخت‌افزار وجود دارد. این تفاوت‌ها را می‌توان به صورت زیر تبیین کرد.

۶-۱- تفاوت در فرآیند تولید

فرآیند تولید نرم‌افزار یک فرآیند مهندسی است تا محصول قابل قبولی تولید شده و تحویل کارفرما شود. در صورتی که فرآیند تولید سخت‌افزار از نوع فرآیند ساخت ۹ می‌باشد، به طوری که می‌توان خط تولید راه انداخت و بر اساس یک مدل و نقشه خاصی، محصولی را تولید انبوه کرد. در تولید نرم‌افزار ظرافت کار بیشتر است و معمولاً از قطعات و ابزارهای استاندارد خبری نیست. در نرم‌افزار اگر اشکالی باشد، در مراحل اولیه تولید خودش را نشان نمی‌دهد، ولی در سخت‌افزار با یک تست می‌توان در مراحل اولیه اشکال را تشخیص داد و برطرف نمود.

در تولید نرم‌افزار ظرافت کار بیشتر است و معمولاً از قطعات و ابزارهای استاندارد خبری نیست.

در نرم‌افزار مشکل کنترل فرآیند تولید مطرح است و این که تعریف مسئله به وسیله افراد غیرمتخصص انجام می‌شود ولی طراحی و ساخت آن به وسیله افراد متخصص صورت می‌گیرد.

تولید نرم‌افزارهای بزرگ به صورت تیمی انجام می‌شود و مدیریت اعضای تیم باید به طور صحیح انجام گیرد و مسئله هماهنگی اعضای تیم همیشه به عنوان یک مشکل مطرح است.

۶-۲- پیچیدگی خود مسئله نرم‌افزار

وجود نیازهای مختلف و گاهی متضاد، عدم وجود زبان شکل ۱: دیدگاه کاربران مختلف نسبت به کیفیت نرم افزار مشترک بین مهندس نرم افزار و کارفرما و عدم درک صحیح یکدیگر، تغییر نیازها در فرآیند تولید و حتی پس از تولید باعث پیچیدگی مسئله نرم افزار شده است.

۳-۶- استاندارد نبودن نرم افزار

به میزان استانداردهای سخت افزاری، در مقوله نرم افزار استاندارد وجود ندارد.

۴-۶- مشکل توصیف رفتار سیستم های پیوسته توسط سیستم های گسسته

سخت افزار یک سیستم گسسته است و با ابزار گسسته می خواهیم رفتار سیستم پیوسته (نرم افزار) را توصیف کنیم. [۲]

۷- اهداف اساسی مهندسی نرم افزار

مهندسی نرم افزار همواره سه هدف مهم زیر را دنبال می کند.

تولید سیستم های نرم افزاری با کیفیت مطلوب که از اهداف اولیه مهندسی نرم افزار به شمار می آید.

تولید سیستم های نرم افزاری با هزینه مطلوب.

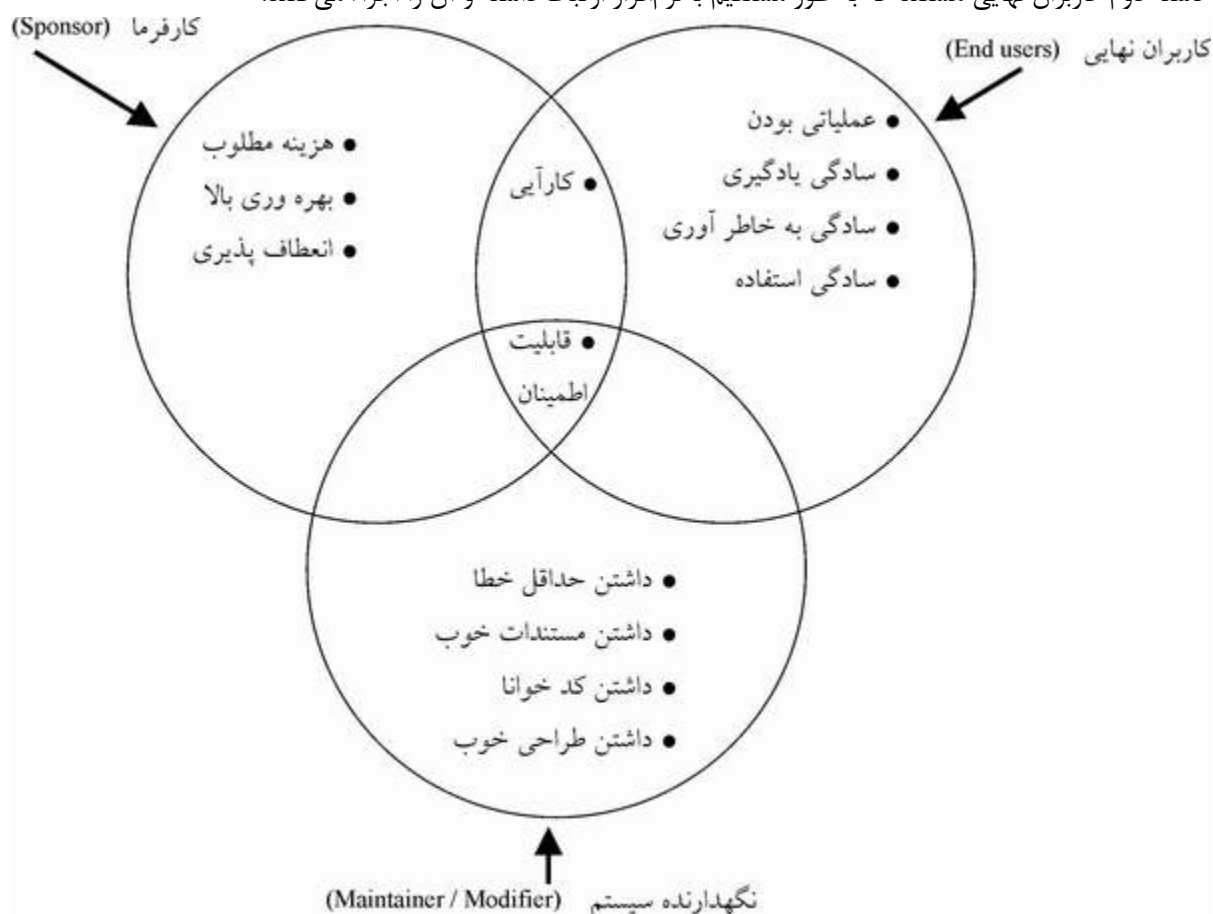
تولید سیستم های نرم افزاری در یک زمان معقول.

۸- کیفیت نرم افزار

منظور از کیفیت نرم افزار چیست؟ پاسخ این سؤال ساده نیست و بستگی به کاربری دارد که به این سؤال پاسخ می دهد. از دیدگاه مهندسی نرم افزار، کاربران مرتبط با سیستم های نرم افزاری به سه دسته تقسیم می شوند.

دسته اول کارفرمای سیستم نرم افزاری است و او فرد یا سازمانی است که هزینه های نرم افزار را می پردازد و برای هزینه های بعدی سیستم هم برنامه ریزی کرده و پاسخگو می باشد. هدف او بیشتر کاهش هزینه تولید و افزایش بهره دهی نرم افزار است.

دسته دوم کاربران نهایی هستند که به طور مستقیم با نرم افزار ارتباط داشته و آن را اجرا می کنند.



دسته سوم نگهدارنده‌ها و توسعه دهندگان سیستم هستند که مسئولیت اصلاح و بهینه‌سازی سیستم برای سرویس دهی بهتر را به عهده دارند. کیفیت نرم‌افزار چیزی نیست که بعد از تولید بتوان به آن اضافه کرد. باید از همان مراحل اولیه به فکر آن بود، قدم به قدم آن را بررسی کرد و در تمام مراحل فرآیند تولید به آن دقت نمود. حال به جاست که کیفیت را از دید کاربران فو مورد بررسی قرار دهیم.

همان طوری که در شکل «۱» ملاحظه می‌شود، بیشترین توجه کارفرما در خصوص کیفیت یک سیستم نرم‌افزاری به هزینه تولید و بهره‌وری بالای آن است؛ در حالی که تمام ذهن و توجه یک کاربر عادی به سادگی استفاده از سیستم معطوف می‌باشد. پر واضح است که عمده زمان و هزینه یک سیستم نرم‌افزاری مربوط به دوره نگهداشت آن می‌باشد؛ بنابراین از دید نگهدارنده‌ها و توسعه دهندگان سیستم، وجود مستندات خوب پر واضح است که عمده زمان و هزینه یک سیستم نرم‌افزاری مربوط به دوره نگهداشت آن می‌باشد؛ بنابراین از دید نگهدارنده‌ها و توسعه دهندگان سیستم، وجود مستندات خوب و کد خوانا از اهمیت ویژه‌ای برخوردار است.

و کد خوانا از اهمیت ویژه‌ای برخوردار است. از دیدگاه مهندسی نرم‌افزار دوره نگهداری و توسعه سیستم‌های نرم‌افزاری از مهم‌ترین و حساس‌ترین دوره‌های تولید و بهره‌برداری محسوب می‌شود؛ زیرا این دوره، مرحله‌ای است که نرم‌افزار عملیاتی می‌شود و کلیه قابلیت‌ها و کارآیی خود را نشان می‌دهد. در این دوره سه فعالیت مهم انجام می‌گیرد که عبارتند از:

رفع اشکالات احتمالی سیستم در حین اجرا ۱۰

تصحیح و بهینه‌سازی سیستم ۱۱

بسط و توسعه سیستم ۱۲

بدیهی است سیستمی که خطا داشته باشد قابل اطمینان نیست و سیستمی که قابل اطمینان نباشد نمی‌تواند عملیاتی باشد و سیستم غیرعملیاتی نمی‌تواند در افزایش بهره‌وری مؤثر باشد.

۹- توزیع هزینه‌های نرم‌افزار از دیدگاه مهندسی نرم‌افزار

۹-۱- هزینه‌های تولید و نگهداری سیستم

هزینه مرحله تولید نرم‌افزار ۳۳٪ کل هزینه پیش‌بینی می‌شود.

هزینه مرحله نگهداری نرم‌افزار ۶۷٪ کل هزینه پیش‌بینی می‌شود.

۹-۲- توزیع هزینه در مراحل مختلف تولید سیستم

هزینه "تحلیل و طراحی سیستم" ۴۰٪ کل هزینه را تشکیل می‌دهد.

هزینه "پیاده‌سازی سیستم" ۲۰٪ کل هزینه را تشکیل می‌دهد.

هزینه "تست سیستم" ۴۰٪ کل هزینه را تشکیل می‌دهد.

۹-۳- توزیع هزینه در مراحل مختلف تولید و نگهداری سیستم

هزینه تحلیل نیازمندی‌ها ۶٪ هزینه کل.

هزینه طراحی سیستم ۵٪ هزینه کل.

هزینه کدینگ ۷٪ هزینه کل.

هزینه تست واحد ۱۳ ۸٪ هزینه کل.

هزینه تست سیستم ۱۴ ۷٪ هزینه کل.

هزینه نگهداری سیستم ۶۷٪ هزینه کل.

۱۰- دلایل سختی و پیچیدگی تولید نرم‌افزار

گفته شد که تولید نرم‌افزار یک فرآیند پیچیده و هزینه بر است. عوامل مختلفی در پیچیدگی و سختی تولید نرم‌افزار دخالت دارند که به طور فشرده به پنج عامل مهم اشاره می‌شود.

۱۰-۱- نحوه ارتباطات تولیدکننده و کارفرمای نرم‌افزار

عموماً تولیدکننده و کارفرما هر کدام اطلاعات و نقطه نظرانی در خصوص مسائل خاص خود دارند که دیگری به اندازه کافی اشرافیتی به آنها ندارد. هر کدام با زبان تخصصی خاص خویش صحبت می‌کنند که برای دیگری مبهم و غیرقابل فهم است.

معمولاً در پروژه‌های بزرگ نرم‌افزاری، ارتباطات کارفرما و تولیدکننده مشکل‌تر می‌شود. همچنین ارتباط بین تیم‌های تولیدکننده، که خود جزو مسائل اصلی تولید و نگهداری سیستم‌های نرم‌افزاری است و هماهنگی بین اعضای تیم مدیریت دقیق و صحیحی را طلب می‌کند. مسیرهای ارتباطی بین اعضا، از فرمول $n(n-1)2$ قابل محاسبه است و روشن است که هر چه تعداد اعضای تیم بیشتر باشد، تعداد مسیرهای ارتباطی بیشتر و در نتیجه بدفهمی و انتقال ناقص مفاهیم بین اعضا بیشتر می‌شود. راه حل مهندسی نرم‌افزار برای حل این مشکل، مستندسازی دقیق و استاندارد کلیه مراحل تولید نرم‌افزار می‌باشد و برای ایجاد ارتباط صحیح بین افراد دخیل در پروژه‌های بزرگ، استفاده از مستندسازی اجتناب‌ناپذیر است.

۱۰-۲- ترتیبی بودن طبیعت تولید نرم‌افزار

حقیقتی که در تولید سیستم‌های نرم‌افزاری وجود دارد، طبیعت ترتیبی بودن آن است. بسیاری از کارهایی که در مراحل تولید انجام می‌گیرد، دارای ترتیب خاصی است؛ مثلاً آنالیز نیازمندی‌ها قبل از طراحی، طراحی قبل از کدینگ و کدینگ قبل از تست و... انجام می‌گیرد. با این دیدگاه باید در نظر داشت که اضافه کردن پرسنل به یک پروژه نرم‌افزاری، همیشه باعث افزایش سرعت پیشرفت آن نمی‌شود، بلکه در بعضی اوقات اثر سوء در پیشرفت آن به جای می‌گذارد. آقای بروکس ۱۵ عقیده دارند که افزایش پرسنل، زمانی باعث افزایش سرعت پیشرفت سیستم‌های رزرواسیون هوایی، سرویس‌گر بانک و سایر سیستم‌های باید از پاسخ مناسب زمانی برخوردار باشند و سیستم‌هایی، از قبیل سیستم اکسیژن دهی به بیمار، بورس، فروش و... باید از اطمینان‌پذیری بالایی برخوردار باشند.

پروژه می‌شود که ماهیت کار قابل تقسیم به چند کار مستقل باشد و هر کار توسط یک نفر با برقراری ارتباط صحیح بین آنها قابل انجام باشد.

بنابراین، افزایش پرسنل جدید به پروژه باعث بروز مشکلات ناشی از ارتباطات بین آنها، آموزش پرسنل جدید و رساندن آنها به سرعت مطلوب کار خواهد شد. آقای بروکس می‌گوید: افزودن پرسنل جدید به پروژه‌های معمو، باعث معوتر شدن آنها می‌شود.

۱۰-۳- ویژگی‌های خاص پروژه

عوامل تعیین‌کننده در این مورد را می‌توان به شرح زیر بیان نمود.

اندازه پروژه: به این معنی که پروژه چه تعداد عملیات را پردازش می‌کند؟ حجم داده‌ها و پیچیدگی ساختمان داده‌های آن چقدر است؟

بدیعی بودن مسئله و روش‌های حل آن: برای مثال پیاده‌سازی سیستم‌هایی، از قبیل سیستم عامل، کنترل ورودی/خروجی دستگاه‌ها و انتقال داده، خیلی مشکل‌تر از برنامه‌های کاربردی، از قبیل کامپایلرها، اسمبلرها و ادیتورهاست.

متغیر بودن نیازمندی‌ها و خواسته‌های کارفرما.

تغییر در پیکره‌بندی‌های سخت‌افزاری و نرم‌افزاری

در نظر گرفتن امنیت، زمان پاسخ، قابلیت اطمینان و تأمین نیازمندی‌های حیاتی توسط سیستم نرم‌افزاری مورد نظر امنیت سیستم، با چگونگی کنترل و دستیابی به برنامه‌ها، فایل‌ها، رکوردها و نقل و انتقال داده‌ها در ارتباط تنگاتنگ است؛ طراحی سیستمی که امنیت را کنترل می‌کند، بسته به نوع و درجه امنیت می‌تواند پیچیدگی بسیار بالایی داشته باشد. در برخی از سیستم‌ها، سرعت پاسخ، از اهمیت خاصی برخوردار است. سیستم‌های رزرواسیون هوایی، سرویس‌گر بانک و سایر سیستم‌های *Online* باید از پاسخ مناسب زمانی برخوردار باشند و سیستم‌هایی، از قبیل سیستم اکسیژن دهی به بیمار، بورس، فروش و... باید از اطمینان‌پذیری بالایی برخوردار باشند. سیستم‌های دیگری به عنوان سیستم‌های بحرانی مطرح هستند که این گونه سیستم‌ها در مقابل خطا و خرابی، قابلیت ترمیم خودکار را دارا می‌باشند. مثلاً جنگنده $F-16$ به طور کامل توسط نرم‌افزار رایانه‌ای هدایت می‌شود؛ بنابراین باید از درجه اطمینان‌پذیری بالایی برخوردار باشد که برای تولید نرم‌افزار آن ۸۵ میلیون دلار و برای نگهداری و توسعه آن ۲۵۰ میلیون دلار هزینه شده است.

۱۰-۴- ویژگی‌های خاص پرسنل

علاوه بر عامل ارتباطات بین پرسنل، ویژگی‌های فردی پرسنل نیز در مشکلات تولید نرم‌افزار تأثیرگذار است.

از جمله این ویژگی‌ها می‌توان به موارد ذیل اشاره نمود.

توان انجام کار؛

تجربه و آموزش قبلی؛

توانایی در برقراری ارتباط بین یکدیگر و با کارفرما؛

توانایی کار به شکل گروهی و تیمی.

۱۰-۵- ملاحظات و مسائل مدیریتی

در این خصوص امور مدیریتی داخل و خارج پروژه مطرح می‌باشد. زیرا بعضی مواقع مدیر داخلی پروژه‌های نرم‌افزاری، مدیریت کارهای خارج از پروژه را نیز بر عهده دارد. احتمال دارد مدیریت یک پروژه توسط فردی انجام شود که از منطقه دیگری آمده است و تجربه و مهارت (دانش) کافی برای هدایت پروژه را ندارد. یکی از مسائل اساسی در مهندسی نرم‌افزار، عدم توازن مابین جنبه‌های فنی و جنبه‌های مدیریتی نرم‌افزار می‌باشد.

۱۱- دوره حیات سیستم‌های نرم‌افزاری

هر سیستم نرم‌افزاری دارای سه دوره حیات است.

۱۱-۱- دوره طفولیت ۱۶

در این دوره پس از طی مراحل مختلف، نرم‌افزار تولید و تست می‌شود.

عدم رعایت اصول مهندسی نرم‌افزار در ابعاد مختلف اعم از مسائل مدیریتی، روش‌های مستندسازی، هماهنگ‌سازی تیم فنی، ملحوظ کردن ویژگی‌های خاص پروژه، روش‌های تست و... در مرحله تولید، باعث تضعیف استحکام نرم‌افزار شده و قابلیت اطمینان و کارایی آن را دچار خدشه می‌کند.

۱۱-۲- دوره بلوغ ۱۷

در این دوره نرم‌افزار، اجراء و عملیاتی شده و سرویس‌ها و خدمات پیش‌بینی شده را به کارفرما ارائه می‌دهد. در همین مرحله فعالیت‌های مربوط به رفع خطا، اعمال تغییرات مورد نیاز و توسعه قابلیت‌های نرم‌افزار انجام می‌گیرد. بدیهی است این دوره از حساس‌ترین مراحل تولید و توسعه نرم‌افزار محسوب می‌شود و اگر دقت کافی، بکارگیری روش‌های استاندارد مستندسازی و تست دقیق در مرحله تولید انجام شود، این دوره طولانی خواهد شد و با کمترین هزینه، کارفرما می‌تواند از قابلیت‌های نرم‌افزار بهره‌مند شود. عدم رعایت اصول مهندسی نرم‌افزار در ابعاد مختلف اعم از مسائل مدیریتی، روش‌های مستندسازی، هماهنگ‌سازی تیم فنی، ملحوظ کردن ویژگی‌های خاص پروژه، روش‌های تست و... در مرحله تولید، باعث تضعیف استحکام نرم‌افزار شده و قابلیت اطمینان و کارایی آن را دچار خدشه می‌کند. از این رو نرم‌افزار تولید شده با خطا و اشکالات فراوانی مواجه شده و ضمن کاهش بهره‌وری، هزینه زیادی را برای رفع خطا و تطبیق‌پذیری با نیازمندی‌های واقعی به کارفرما، تحمیل خواهد کرد و این مسئله کاهش طول عمر نرم‌افزار و از رده خارج شدن آن را به دنبال دارد.

۱۱-۳- دوره انفصال از خدمت ۱۸

عواملی که سبب بازنشستگی و انفصال از خدمت نرم‌افزار می‌شود به قرار زیر است.

* عدم تأمین نیازهای واقعی کارفرما

* داشتن خطای زیاد در حین اجرا و عدم قابلیت اطمینان

* تغییر در شیوه عملکرد سازمان و عدم انعطاف‌پذیری نرم‌افزار

* مطرح شدن نرم‌افزارهای جدید با قابلیت بهتر و سریع‌تر

* تغییر نسل‌ها و نسخه‌های نرم‌افزارهای پایه، از قبیل سیستم عامل و پایگاه داده

* تغییر در ساختار و پیکره‌بندی شبکه و سخت‌افزار

۱۲- خلاصه‌ای از مشکلات مطرح شده

نرم‌افزار در مقابل آنچه کاربر می‌خواهد شکست می‌خورد.

تولید و نگهداری نرم‌افزار گران است.

همیشه به اندازه کافی سریع نیست و از کارایی کمتری برخوردار است.

همیشه قابل انتقال از یک محیط به محیط دیگر نیست.

قابلیت اعتماد آن پایین است.

به موقع تحویل نمی‌شود.

تولید انبوه نرم‌افزار میسر نیست.

تولید نرم‌افزار به عوامل انسانی وابسته است و این امر باعث افزایش هزینه‌های آن می‌شود. نرم‌افزار حاصل فکر انسان است و نسبت به تولید قطعات فیزیکی (سخت‌افزار) پیچیدگی بیشتری دارد.

۱۳- ایده‌ها و راه‌حل‌های مهندسی نرم‌افزار

ایده‌ها و راه‌حل‌های مهندسی نرم‌افزار به شرح زیر است:

تأکید بر انجام مراحل تولید، توسعه و نگهداری نرم‌افزار به صورت نظام‌مند

کمک گرفتن از ابزارهای مهندسی نرم‌افزار برای توسعه آن

تمرکز برای دستیابی به نیازهای واقعی کاربر

مستندسازی دقیق و استاندارد مشخصات سیستم در مراحل مختلف تولید و توسعه

ارائه نرم‌افزار با ایجاد نمونه ۱۹

حتی‌الامکان استفاده از زبان‌های برنامه‌نویسی جدید با انعطاف‌پذیری بالا

تست نرم‌افزار با ارائه الگوهای تست ۲۰ دقیق و تلاش در جهت به صفر رساندن خطا

به طور کلی هدف مهندسی نرم‌افزار دستیابی به یک سیستم مستند، ساخت یافته و با کیفیت مطلوب است که در زمان مشخص، با هزینه معین و با صرف نیروی انسانی مجرب و دیگر منابع پیش‌بینی شده، حاصل می‌شود.

۱۴- نتیجه‌گیری

برای تولید، توسعه و نگهداری سیستم‌های نرم‌افزاری، لازم است از اصول، قواعد و روش‌های استاندارد پیروی کنیم تا محصول تولید شده از کارایی بیشتر و قابلیت اطمینان بالاتر برخوردار باشد. بکارگیری روش‌های مناسب، علاوه بر افزایش کیفیت محصول، به صرفه جویی در زمان و کاهش هزینه نیز کمک خواهد کرد. پر واضح است که به صرف پیروی از روش‌ها و اصول استاندارد نمی‌توان به موفقیت کامل در پروژه‌های نرم‌افزاری نائل شد. بلکه ملحوظ داشتن عوامل پیچیدگی تولید نرم‌افزار، از قبیل مسائل ارتباطات، ماهیت نرم‌افزار، ویژگی‌های خاص هر کدام از پروژه‌ها، تجربیات و توان کاری تیم‌های فنی تولیدکننده نرم‌افزار و مسائل مدیریتی، از اهمیت ویژه‌ای برخوردار است. اصول مهندسی نرم‌افزار به هیچ وجه استفاده از راه‌های میان‌بر و غیر روشمند را به طراحان سیستم‌های نرم‌افزاری توصیه نمی‌کند، زیرا این کار ممکن است رضایت کوتاه مدت کاربر را در اتمام سریع نرم‌افزار فراهم کند، لکن مشکلات عدیده بلند مدت، از قبیل افزایش هزینه‌های نگهداری، غیرقابل انعطاف بودن نرم‌افزار، مراجعات مکرر برای حل مشکل سیستم و نهایتاً اتلاف زمان و هزینه‌های بیشتری را در پی خواهد داشت. اصول مهندسی نرم‌افزار به دنبال آن است که کاربران در هر مرحله به خواسته‌هایشان و پاسخ‌هایی که می‌دهند، دقت داشته باشند، زیرا این نگرش سبب می‌شود که زمان و هزینه صرف شده برای تجزیه و تحلیل نیازمندی‌ها، در مرحله پیاده‌سازی و نگهداری سیستم جبران شود. این روش همچنین سبب می‌شود که در مرحله تست و تنظیم، سیستم با تغییرات ناخواسته و تجدید نظرهای ریشه‌ای مواجه نشود. به طور کلی هدف مهندسی نرم‌افزار دستیابی به یک سیستم مستند، ساخت یافته و با کیفیت مطلوب است که در زمان مشخص، با هزینه معین و با صرف نیروی انسانی مجرب و دیگر منابع پیش‌بینی شده، حاصل می‌شود. با توجه به مطالب فوق، بدیهی است که هر چه سیستم‌های نرم‌افزاری بزرگ‌تر و تعاملات بین زیرسیستم‌های آنها بیشتر می‌شود، بکارگیری اصول و مبانی مهندسی نرم‌افزار برای تولید، توسعه و نگهداری آنها اجتناب‌ناپذیرتر می‌شود.

سیستم‌های یکپارچه لجستیکی در ابعاد سازمانی، نیروی انسانی و استقرار جغرافیایی دارای ویژگی‌های خاص و پیچیدگی‌های منحصر به فردی است که برای دستیابی به موفقیت در امر مکانیزاسیون و ماشینی کردن آنها لاجرم باید همه شرایط، ملاحظات، نقاط قوت و ضعف خودمان را به طور عمیق مورد بررسی و ارزیابی قرار دهیم و با آگاهی کامل تصمیم اصول مهندسی نرم‌افزار به دنبال آن است که کاربران در هر مرحله به خواسته‌هایشان و پاسخ‌هایی که می‌دهند، دقت داشته باشند، زیرا این نگرش سبب می‌شود که زمان و هزینه صرف شده برای تجزیه و تحلیل نیازمندی‌ها، در مرحله پیاده‌سازی و نگهداری سیستم جبران شود.

بگیریم. به نظر می‌رسد، اولین گام، فراهم کردن مقدمات لازم برای جذب و سازماندهی نیروی انسانی ماهر، مجرب، توانمند و آشنا به اصول و مبانی کار و آگاه به سختی و پیچیدگی‌های مسئله باشد. چرا که در طول تاریخ، همیشه انسان‌های بصیر و درد آشنا توانسته‌اند راه‌های پرپیچ و خم را بیمایند و گره‌های ناگشودنی را بگشایند. در این قسمت به همین اجمال بسنده می‌کنیم و تفصیل بحث را به مجال دیگر می‌گذاریم.

پی نوشت ها

- Modularity - ۱
- Reuseability - ۲
- Computer Networks - ۳
- Distributed Systems - ۴
- Embedded Systems - ۵
- Case Tools - ۶
- Expert Systems - ۷
- Neural Networks - ۸
- Manufacturing - ۹
- Fixing Error - ۱۰
- Making Modification - ۱۱
- Adding Enhancements - ۱۲
- Unit Test - ۱۳
- System Test - ۱۴
- Brooks - ۱۵
- Childhood Cycle - ۱۶
- Adulthood Cycle - ۱۷
- Retirement Cycle - ۱۸
- Prototype - ۱۹
- Test Patterns - ۲۰

منابع و مأخذ

- USA: Prentice hall, , ۱ Vol ;Projects Engineering with student Software; Barbee ,Mynnat - ۱
1995
- USA: , ۱ vol ;Software Engineering with Practitioner`s Approach ; Pressman, Roger - ۲
Mcgrawhill, 1997