



دانشگاه آزاد اسلامی مرکز ملارد

Software Quality Assurance

تضمین کیفیت نرم افزار

استاد : جناب آقای مهندس مروجی



فهرست :

5.....	مقدمه
6.....	مهندسی نرم افزار و تولید نرم افزاری با کیفیت بالا
8.....	تعریف رسمی SQA (Formal SQA Definition)
8.....	کنترل کیفیت نرم افزار (SQC)
9.....	ادغام مدل بلوغ قابلیت (CMMI)
10.....	تضمین کیفیت
11.....	هزینه کیفیت
12.....	مراحل یک فرآیند تضمین کیفیت معمولی
12.....	کیفیت نرم افزار
14.....	مشخصه های اصلی کیفیت نرم افزار
15.....	سنجش کیفیت نرم افزار (Software Quality Measurement)
17.....	ویژگی های کیفی نرم افزار
17.....	مدل کیفیت مک کال
18.....	مدل کیفیت بوهم
20.....	مقایسه این دو مدل
21.....	SQC در مقابل SQA
22.....	استانداردهای ساخت و مستندسازی محصولات نرم افزاری
22.....	استاندارد ISO 9001
23.....	حوزه کاربرد

23.....	ساختار استاندارد
25.....	استاندارد های محصول
33.....	مدیریت پروژه نرم افزار
34.....	مدیریت پیکر بندی نرم افزار
37.....	وارسی و اعتبار سنجی
38.....	تضمین کیفیت نرم افزار
40.....	استاندارد IEEE/EIA 12207
41.....	معرفی کلی استاندارد
43.....	ساختار استاندارد
49.....	طرح تضمین کیفیت نرم افزار
50.....	مدیریت
50.....	مستند سازی
50.....	استانداردهای رویه ای
52.....	بازبینی و ممیزی
52.....	فعالیت های آزمون
53.....	گزارش مشکلات و انجام اصلاحات
53.....	ابزارها و روش ها
53.....	کنترل برنامه و رسانه ها
53.....	کنترل پیمانکاری
54.....	گردآوری و ذخیره سازی
54.....	آموزش
54.....	مدیریت ریسک
55.....	منابع

مقدمه

تولید نرم افزارهای کاربردی روزبه روز گسترش می یابد و لزوم بکارگیری روش ها و اصول مهندسی نرم افزار در مراحل توسعه ، مدیریت و پشتیبانی آنها بیشتر نمود پیدا می کند . کیفیت نرم افزار (Software Quality) شاخص حیاتی برای تولید نرم افزار های با کیفیت بالاست که ضمن بالا بردن بهره وری در تولید نرم افزار ها ، به ایجاد نرم افزار های قدرتمند و شکست ناپذیر منجر می گردد .

همانطور که شاهدیم در سال های اخیر نگاه علمی به مقوله اقتصاد ، معادلات حاکم بر روابط اقتصادی و تجارت سنتی را تحت تاثیر قرار داده و راهبرد توسعه اقتصادی متکی به توسعه صنعتی ، جای خود را به توسعه اقتصادی بر مبنای توسعه علمی داده است . به گونه ای که در اقتصاد دانش محور ، توسعه علم و فن آوری بویژه فن آوری اطلاعات و ارتباطات (IT) به عنوان شیوه ای مطمئن و قدرتمند برای تولید ثروت و توسعه اقتصادی شناخته می شود . در این بین و با الگو برداری از پیشگازان عرصه تولید نرم افزار همچون کشور های هند و ایرلند و شیوه های بکار رفته توسط ایشان در توسعه اقتصادی مبتنی بر نرم افزار ، متوجه خواهیم شد که در پس همه این طرح ها و موفقیت ها باید به جستجوی زیر ساخت ها بر آئیم ، زیر ساخت هایی که منجر به تولید و ساخت نرم افزار های قدرتمند و بر پایه اصول علمی در سطح جهانی شده است .

از این رو وجود زیر ساخت های بسیار قوی علمی ، فنی و مهندسی بکار گرفته شده توسط این کشورها را می توان به عنوان الگویی برای سایر کشورها در نظر گرفت و با بهره گیری از این تجربیات به ارتقاء جایگاه نرم افزار و صادرات آن و نیز تعامل مثبت آن با اقتصاد پرداخت . مدل سازی نرم افزار ، بکارگیری تکنیک های پیشرفته آزمایش نرم افزار ، مدیریت ریسک نرم افزار ، تضمین کیفیت نرم افزار ، مهندسی محصول و تنها عناوینی از فهرست گسترده زیر ساخت های مرتبط با توسعه نرم

افزار های قوی و مهندسی ساز است که در نوشتار حاضر به طور خاص به بررسی علمی و فنی یکی از این زیر ساخت ها با عنوان "تضمین کیفیت نرم افزار" و در زیرشاخه ی آن به کیفیت نرم افزار و راه های تضمین و بهبود آن پرداخته خواهد شد. این نوشتار با ارائه تعاریف و اهمیت نقش کیفیت نرم افزار در تولید سیستم های نرم افزاری مهندسی ساز و ارائه یک نمونه از استاندارد های کاربردی در این خصوص ، سعی دارد نشان دهد که توجه به ساخت نرم افزار با کیفیت بالا در جهت بهره گیری مطلوب می تواند کارآیی سیستم ها را افزایش داده و دستیابی به ظرفیت های جدیدی را فراهم آورد .

مهندسی نرم افزار و تولید نرم افزاری با کیفیت بالا

مهندسی نرم افزار، یک روش علمی ، ریاضی و اقتصادی برای تولید نرم افزارها است که بر اساس آن، نرم افزار در طی یک فرایند علمی، تجزیه و تحلیل، طراحی، پیاده سازی، آزمایش و پشتیبانی می شود. بکارگیری مهندسی نرم افزار برای پیاده سازی نرم افزارهایی که اهداف مهم و حیاتی دارند یک ضرورت است. در مهندسی نرم افزار برای ساخت یک سیستم نرم افزاری سه فرایند مهم تاثیر گذار می باشند:

✓ فرایند توسعه (Development Process): سازماندهی فعالیت ها است برای ساخت یک سیستم.

✓ فرایند مدیریت (Management Process): انتخاب افراد، تجهیزات و فرایند هاست برای توسعه یک سیستم و کنترل و نظارت بر روند اجرای پروژه .

✓ فرآیند پشتیبانی (Maintenance Process): کنترل و پشتیبانی نرم افزار پس از تولید آن.

در فرآیند توسعه هدف آن است که یک سیستم با مشخصات خواسته شده تولید شود. فرآیند توسعه از مرحله طرح یک راه حل مفهومی برای مساله خواسته شده (مطالعه امکان سنجی) آغاز شده، پس از دریافت خواسته ها و بررسی سیستم ، طراحی صورت گرفته و در نهایت این طراحی با کمک ابزارهای پیاده سازی تبدیل به یک سیستم واقعی می شود. هدف این فرآیند آن است که از یک سو برآورده ساختن نیازهای کاربران و از سوی دیگر کیفیت مناسب عملکرد سیستم تضمین گردد و بنابراین بایستی حاوی مکانیسم هایی برای اعتبار سنجی نرم افزار (خروجی مطابق با خواسته ها (Validation) و واریسی پذیری نرم افزار (صحت عملکرد خروجی (Verification) باشد.

با فرض اینکه تمامی نرم افزار های ایجاد شده بر اساس ،فرآیند مهندسی نرم افزار تولید شده باشند ، باز هم با هم تفاوت هایی دارند . مسئله تفاوت بین نمونه ها برای تمام محصولات تولید شده توسط انسان وجود دارد . تفاوت های بین نمونه ها ممکن است بدون کمک تجهیزات دقیق اندازه گیری ابعاد فنی و مهندسی آن امکان پذیر نباشد اما حتی با دستگاه هایی که به اندازه کافی هم دقیق و حساس نیستند باز هم به این نتیجه می رسیم که هیچ دو نمونه نرم افزاری شبیه هم نیستند . آنچه در این میان اهمیت دارد و باعث وضوح این تفاوت ها می شود، کیفیت نرم افزارهاست.

تعریف رسمی SQA (Formal SQA Definition)

تضمین کیفیت نرم افزار (SQA) متشکل از ابزار نظارت بر فرآیندهای مهندسی نرم افزار و روش های مورد استفاده برای اطمینان از کیفیت می باشد. در واقع تضمین کیفیت نرم افزار، تابعی از کیفیت نرم افزار است که اطمینان می یابد استانداردها، فرایندها و روشهای مناسب برای این پروژه به درستی انتخاب شده و به صورت صحیح اجرا می شود. روش هایی که توسط آن انجام میگیرد متعدد و متنوع هستند و ممکن است شامل حصول اطمینان از انطباق با یک یا چند استاندارد از قبیل ISO 9000 یا مدل CMMI باشند.

SQA شامل تمامی فرایندهای توسعه می باشد، فرآیندهایی مانند تعریف الزامات، طراحی نرم افزار، برنامه نویسی، کنترل کد منبع، بررسی کد، مدیریت تغییر، مدیریت پیکربندی، تست، مدیریت انتشار، و ادغام محصول. SQA به اهداف، تعهد، توانایی ها، فعالیت ها، اندازه گیری، و تصدیق ها متکی است.

کنترل کیفیت نرم افزار (SQC)

تابعی از کیفیت نرم افزار است و کنترل میکند که پروژه بر اساس استانداردها، فرایندها و رویه ها صورت گیرد. کنترل کیفیت شامل مجموعه ای از بازبینی ها، مرورها و آزمایشات استفاده شده طی فرآیند مهندسی نرم افزار به منظور اطمینان از انطباق محصول با نیازمندی هایی است که برای آن ساخته شده است. کنترل کیفیت می تواند به صورت خودکار، به صورت دستی و یا به صورت ترکیبی از تکنیک ها و روش های خود کار و دستی صورت پذیرد.

ادغام مدل بلوغ قابلیت (CMMI)

بهبود روند رویکرد به توسعه نرم افزار میباشد. در حقیقت SQA و SQC، در درون یک چارچوب CMMI پیاده سازی میشوند. SQA و SQC فرآیندهای تعریف شده در درون CMMI هستند.

CMMI شامل مجموعه ای از هسته اصلی مهندسی نرم افزار است :

- ✓ مورد نیاز توسعه
- ✓ الزامات مدیریت
- ✓ راه حل های فنی
- ✓ یکپارچه سازی محصول
- ✓ تأیید کننده
- ✓ اعتبار سنجی
- ✓ مدیریت فرآیند
- ✓ مدیریت پروژه
- ✓ ...

مدل CMMI (ادغام مدل بلوغ قابلیت) که به طور گسترده استفاده می شود، به پیاده سازی تضمین کیفیت (PPQA) در یک سازمان می پردازد. سطح بلوغ CMMI را می توان به 5 مرحله تقسیم کرد، که یک شرکت می تواند بر اساس نیازهای خود از آن بهره گیرد. CMMI فرآیندهای عالی QA هستند که توسط شرکت های بزرگی مانند ناسا به کار گرفته شده اند.

تضمین کیفیت

تضمین کیفیت نرم افزار (SQA) عبارت از يك " طرح برنامه ریزی شده منظم از کلیه عملیات لازم برای حصول اطمینان کافی و مناسب در خصوص انطباق عنصر یا محصول تولید شده با مشخصات فنی مورد نظر " می باشد . تضمین کیفیت نرم افزار با " تضمین محصول " نرم افزاری مترادف بوده و در این مجموعه از استانداردها این دو اصطلاح به جای یکدیگر به کار می روند .

فعالیت تضمین کیفیت ، فرآیند اثبات صحت به کارگیری این استانداردها است . در پروژه های كوچك این فعالیت می تواند توسط گروه تولید نرم افزار انجام پذیرد ، ولی در پروژه های بزرگ می بایست افراد مشخص و متخصصی به انجام این وظیفه تخصیص یابند.

تضمین کیفیت (QA) اشاره به اندازه گیری سیستماتیک، مقایسه آن با استاندارد، نظارت بر روند ها و حلقه ی بازخورد همراه خود (برای پیشگیری از خطا) دارد. این مورد را می توان با کیفیت "کنترل" در تضاد دانست چرا که بر روی خروجی فرایند متمرکز شده است. هدف اصلی از تضمین نرم افزار این است که اطمینان حاصل شود که فرایندها، رویه ها و محصولات مورد استفاده برای تولید و حفظ این نرم افزار مطابق با تمامی الزامات و استانداردهای تعیین شده است. هدف دوم از تضمین نرم افزار این است که مطمئن شویم سیستم های نرم افزاری فشرده که ما تولید میکنیم امن تر هستند.

هدف تضمین کیفیت فراهم نمودن روشی به منظور اطلاع از کیفیت محصول است . توجه به این نکته حائز اهمیت است که رضایت مندی مشتری در بکارگیری سیستم های نرم افزاری بر اساس بررسی های صورت گرفته به ترتیب زیر تعریف گردیده است :

**رضایت مندی مشتری = محصول خواسته شده + تحویل بر مبنای بودجه و زمان بندی
+ کیفیت خوب**

دو اصل اساسی در QA عبارتند از :

✓ متناسب برای هدف، محصول باید برای هدف مورد نظر مناسب باشد.

✓ درست اولین بار، اشتباهات باید حذف شود.

QA شامل مدیریت کیفیت مواد خام ، مجامع، محصولات و قطعات، خدمات مربوط به تولید، و فرایندهای مدیریت ، تولید و بازرسی است.

هزینه کیفیت

هزینه کیفیت شامل تمام هزینه های صرف شده برای رسیدن به کیفیت و یا انجام فعالیت های مربوط به آن است . هزینه های کیفیت می توانند به هزینه های مربوط به پیشگیری ، شکست و ارزیابی تقسیم شوند . هزینه های پیشگیری شامل آموزش ، وسائل و تجهیزات آزمایش و برنامه ریزی کیفیت است . هزینه شکست مربوط به دوباره کاری ، تعمیر و تحلیل وضعیت شکست خواهد شد و هزینه ارزیابی مشتمل بر بازبینی فرایندها و آزمایش خواهد بود .

مراحل یک فرآیند تضمین کیفیت معمولی

اشکال بسیاری از فرآیندهای QA، تفاوت دامنه و عمق آنهاست که باعث می شود استفاده از آن برای هر فرآیند خاص و شاید منحصر به فرد باشد و اغلب برای یک فرایند، سفارشی شده است. یک فرآیند معمولی ممکن است شامل موارد زیر باشد :

- ✓ آزمایش مقالات قبلی
- ✓ طرح به منظور بهبود
- ✓ طراحی شامل بهبود و الزامات
- ✓ تولید با بهبود
- ✓ بررسی آیتم های جدید و بهبود
- ✓ آزمون آیتم جدید

کیفیت نرم افزار

مدیران و خبرگان دنیای نرم افزار بر این عقیده اند که کیفیت بالای محصول نرم افزاری به صرفه جویی در هزینه و ارتقاء همیشگی سطح نرم افزار منتج می شود . این در حالیکه تمامی توسعه دهندگان نرم افزاری توافق دارند که دستیابی به نرم افزار های با کیفیت ، بالاترین هدف در ایجاد وساخت سیستم های نرم افزاری است . اما کیفیت نرم افزار چگونه تعریف می شود ؟ کیفیت نرم افزار مطابق با نیازهای عملیاتی و استاندارد های توسعه نرم افزار تعریف و تدوین می گردد و در این میان توجه به سه اصل زیر اهمیت دارد .

- ✓ استانداردها ، مجموعه ای از معیارهای توسعه را تعریف می کنند و چنانچه این معیارها بدرستی دنبال نشوند ، نتیجه آن فقدان کیفیت خواهد بود .

✓ چنانچه یک نرم افزار منطبق بر نیازهای اصلی خود باشد اما نیازهای جانبی خود را (مانند سهولت کاربری و پشتیبانی مناسب) را برآورده نسازد ، کیفیت نرم افزار حاصل نگریده است.

✓ نیازمندی های نرم افزار و آنچه که نرم افزار برای آن طراحی و پیاده سازی گردیده است ، مبنای اندازه گیری کیفیت است . عدم تطابق نرم افزار با نیازمندی های آن موجب عدم کیفیت نرم افزار خواهد شد .

کیفیت در معنی عام آن به مفهوم خصوصیت یا صفتی از یک شی است. درمورد یک شی، کیفیت اشاره به خصوصیتی از از قبیل رنگ ، شکل ، اندازه و ... دارد و در مورد یک نرم افزار شامل درجه پیچیدگی درونی الگوریتم های آن ، تعداد خطوط برنامه نرم افزاری ، ارتباطات داخلی زیر برنامه ها و ... می شود . کیفیت نرم افزار در زمینه مهندسی نرم افزار، اشاره به دو مفهوم مرتبط اما مجزا دارد:

✓ کیفیت طراحی

به ویژگی هایی برمی گردد که طراح برای محصول مشخص می کند. پایه مواد اولیه، تولرانسها و مشخصات کارایی در کیفیت طراحی سهم هستند. اگر محصول به موازات این موارد تولید شود کیفیت طراحی محصول افزایش می یابد.

✓ کیفیت مطابقت

حدپیروی از مشخصات در طول ساخت است. هر چه میزان مطابقت بیشتر باشد سطح کیفیت مطابقت بالاتر خواهد بود.

مشخصه های اصلی کیفیت نرم افزار

قابلیت اطمینان یک نرم افزار بخش مهمی از کیفیت آن است . اگر یک نرم افزار به دفعات متعدد با شکست در اجرا مواجه شود آنگاه یکی از مهمترین عوامل کیفی آن از بین رفته است . قابلیت اطمینان نرم افزار به این شکل تعریف می شود : عملکرد بدون شکست یک نرم افزار در محیطی خاص و برای مدت زمانی معین . توجه به این نکته حائز اهمیت است که شکست نرم افزار در اثر اشکالات طراحی است . در واقع تمام شکست های نرم افزاری به مشکلات طراحی یا پیاده سازی منتهی می شوند . امنیت نرم افزار یکی از شاخص های تضمین کیفیت نرم افزار است که متضمن تشخیص خطرات احتمالی است که بر روی نرم افزار تأثیرات منفی دارند و موجبات شکست سیستم را بوجود می آورند . اگر این خطرات احتمالی زود تشخیص داده شوند ، اشکالات و نواقص طراحی نرم افزار قابل تشخیص بوده و حذف آنها از فرآیند مهندسی نرم افزار امکان پذیر می گردد .

در زیر تعریف موجز و کوتاهی از نحوه تولید نرم افزار مطلوب و با کیفیت قابل قبول بیان شده است . مطالب زیر مختصر و مفید و قابل درک بوده و بر اصولی عملی و دقیق استوار می باشند . همچنین ضمن اینکه جنبه های اصلی و ضروری هر پروژه ای را در بر می گیرند ، در قالب مجموعه ای از چارچوب ها و ضرورتها ، حداکثر انتخاب ممکن را نیز برای مدیر پروژه فراهم می آورند . استانداردهای مهندسی نرم افزار ESA تحت نظارت هیئت کنترل و استاندارد سازی نرم افزار (BSSC) وابسته به آژانس فضایی اروپا تنظیم و بازبینی شده است .

سنجش کیفیت نرم افزار (Software Quality Measurement)

بررسی ارتباط بین ویژگی های نرم افزار مطلوب (سمت راست تصویر) و ویژگی های قابل اندازه گیری (سمت چپ تصویر) در حقیقت اندازه گیری کیفیت نرم افزار است که بررسی میکند تا چه اندازه یک نرم افزار یا سیستم دارای ویژگی های مطلوب است. این مهم می تواند از طریق ابزارهای کیفی یا کمی و یا ترکیبی از هر دو انجام شود. در هر دو مورد، برای هر یک از مشخصه های مطلوب، مجموعه ای از ویژگی های قابل اندازه گیری وجود دارد. به عنوان مثال، یک ویژگی همراه با قابلیت حمل، تعدادی از بیانیه های وابسته به هدف در یک برنامه است.

✓ برنامه تمرین معماری (Application Architecture Practices)

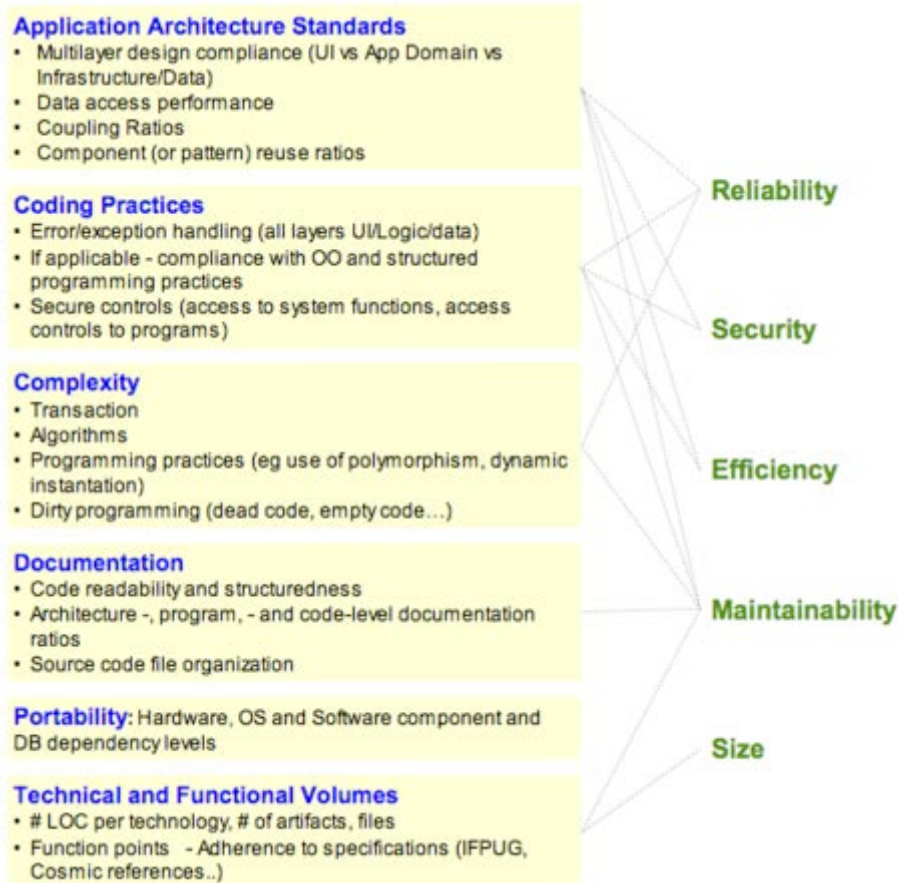
✓ برنامه نویسی تمرین (Coding Practices)

✓ پیچیدگی برنامه (Application Complexity)

✓ مستندات (Documentation)

✓ قابلیت حمل (Portability)

✓ دوره های فنی و تابعی (Technical & Functional Volumes)



در تصویر بالا درخت وابستگی بین ویژگی های کیفیت نرم افزار و ویژگی های قابل اندازه گیری آنها (در سمت راست) و ویژگی های قابل اندازه گیری (سمت چپ) نشان داده شده است.

ویژگی های کیفی نرم افزار

با توجه به ماهیت ناملموس و انتزاعی نرم افزار، محققان و پژوهشگران به دنبال راه هایی برای مشخص کردن نرم افزار به منظور ایجاد مزایا و هزینه های قابل اندازه گیری هستند. در ادامه دو مدل مهم کیفیت را بررسی میکنیم:

✓ مک کال (1977)

✓ بوهم (1978)

مدل کیفیت مک کال – 1977 (McCall's Quality Model)

جیم مک کال، این مدل را برای نیروی هوایی ایالات متحده تولید کرد و هدف اصلی او از بین بردن شکاف بین کاربران و توسعه دهندگان بود. مک کال سه دیدگاه اصلی برای توصیف ویژگی های با کیفیت از محصول نرم افزار دارد :

✓ ویرایشهای محصول (قابلیت تغییر)

✓ انتقال محصول (سازگاری با محیط جدید)

✓ عملیات محصول (ویژگی های اساسی عملیاتی)

ویرایشهای محصولات

دیدگاه ویرایشهای محصول با هدف شناسایی عوامل با کیفیت که توانایی تغییر محصول نرم افزاری را دارا هستند شکل گرفت. این عوامل عبارتند از:

✓ نگهداری ، توانایی پیدا کردن و رفع نقص است.

✓ انعطاف پذیری ، توانایی ایجاد تغییرات مورد نیاز کسب و کار.

✓ تست ، توانایی اعتبارسنجی نرم افزار مورد نیاز است.

انتقال محصول

چشم انداز انتقال محصول شناسایی عوامل با کیفیت است که بر توانایی انطباق نرم افزار با محیط های جدید تاثیرگذارند :

- ✓ قابلیت حمل ، توانایی برای انتقال نرم افزار از یک محیط به محیط دیگر
- ✓ قابلیت استفاده مجدد ، سهولت استفاده از اجزای نرم افزار موجود در زمینه های مختلف

✓ متقابل ، حد ، و یا سهولت ، که اجزای نرم افزار ها با یکدیگر همکاری می کنند

عملیات محصولات

دیدگاه عملیات محصول شناسایی عوامل با کیفیت در نفوذ محصول را بر عهده دارد:

- ✓ صحت ، قابلیت بازی با خصوصیات است
- ✓ قابلیت اطمینان ، تا چه حد که سیستم قابل اطمینان است
- ✓ بهره وری ، استفاده از منابع سیستم (شامل پردازنده ، دیسک ، حافظه ، شبکه)
- ✓ تمامیت ، حفاظت از دسترسی های غیر مجاز
- ✓ قابلیت استفاده ، سهولت استفاده

مدل کیفیت بوهم – 1978 (Boehm's Quality Model)

بوهم یک مدل سلسله مراتبی از ویژگی های با کیفیت نرم افزار ارائه میدهد، در واقع تلاش برای تعریف کیفیت نرم افزار به عنوان مجموعه ای از صفات و متریک (اندازه گیری).

بوهم از سه تعریف اولیه استفاده می کند:

- ✓ به عنوان ابزار ، تا چه حد نرم افزار را می توان مورد استفاده قرار داد (یعنی سهولت استفاده ، قابلیت اطمینان و کارایی)
- ✓ نگهداشت پذیری، سهولت شناسایی نیاز و همچنین سهولت اصلاح و retesting تغییر
- ✓ قابلیت حمل ، سهولت تغییر نرم افزار به یک محیط جدید

بوهم هفت عامل کیفیت را به صورت زیر معرفی میکند:

- ✓ قابلیت حمل ، تا چه حد نرم افزار تحت پیکربندی کامپیوترهای مختلف کار می کنند (یعنی سیستم عامل، پایگاه داده و ...)
- ✓ قابلیت اطمینان، ، عدم وجود نقص در نرم افزار
- ✓ بهره وری، استفاده بهینه از منابع سیستم در حین اجرای صحیح
- ✓ قابلیت استفاده، سهولت استفاده
- ✓ تست، سهولت اعتبار سنجی نرم افزار مطابق با الزامات.
- ✓ فهم، توجه به هدف و ساختار درک
- ✓ انعطاف پذیری ، سهولت تغییر نرم افزار به الزامات تجدید نظر شده

این عوامل کیفیت بیشتر به ساختارهای می پردازد که قابل اندازه گیری هستند.

مقایسه این دو مدل

در اینجا می‌خواهیم به طور خلاصه دو مدل را با هم مقایسه کنیم:

- ✓ هر دو مدل مک کال و بوهم به دنبال یک ساختار مشابه ، با یک هدف مشابه هستند.
 - ✓ هر دوی آنها عواملی را بررسی میکنند که می تواند اندازه گیری شود.
 - ✓ برخی از عوامل کیفیت در هر دو مشترک است مانند :قابلیت استفاده، قابلیت حمل ، کارایی و قابلیت اطمینان
 - ✓ ارزش این دو مدل و سایر مدل ها، به عملی شدن آنهاست نه در معانی و یا تفاوت های ساختاری
 - ✓ ارزش هر دوی آنها به اندازه گیری دقیق (هزینه و سود) نرم افزار وابسته است
- ظهور یک مدل به عنوان بهترین مدل تقریباً غیرممکن و دور از دسترس است و همواره نقایص و مشکلاتی در هر مدل وجود دارد و شاید به همین دلیل است که استانداردها و مدل های دیگر پیشنهاد شده است که این موضوع نشان دهنده ی ماهیت ناملموس نرم افزار و مشکلاتی به ارمغان می آورد است .

SQC در مقابل SQA

در این قسمت به تعریف اصطلاح کنترل کیفیت نرم افزار (SQC) در رابطه با تضمین کیفیت نرم افزار (SQA) میپردازیم و نقش SQA را در چارچوب روند بهبود مستمر، به منظور گسترش در تعریف SQA بررسی میکنیم. این شرایط مشابه به نظر می رسد اما یک مثال ساده، تفاوت اساسی این دو موضوع را برجسته میسازد. یک پروژه نرم افزاری که شامل الزامات، طراحی رابط کاربر و پیاده سازی پایگاه داده SQL است را در نظر بگیرید. تیم SQA یک طرح با کیفیت دارای همه ی استانداردها، فرایندها و رویه هایی که در بهبود پروژه مطرح است را ارائه میکند. این استانداردها ممکن است شامل IEE XYZ طرح مشخصات (مورد نیاز)، سبک راهنمای ABC (برای طراحی رابط کاربر) و استاندارد SQL (برای اجرای SQL) باشند. این موارد همه استانداردها، فرایندها و روش هایی است که باید به دنبال شناسایی و مستندسازی در طرح کیفیت، توسط SQA صورت گیرد. هنگامی که محصول تولید می شود (در این مثال) تیم کنترل کیفیت نرم افزار اطمینان حاصل میکند که محصول از استاندارد مستندسازی (در این مورد IEEE XYZ) پیروی میکند. SQC نیز همین وظیفه را در برابر طراحی رابط کاربر و اجرای SQL دارد که هر دوی این موارد به دنبال استاندارد شناسایی شده توسط SQA انجام شده است. به این ترتیب SQA اجرا می شود، و به دنبال آن SQC به وضوح می تواند کار خود را انجام دهد. علاوه بر این کنترل کیفیت عبارت از یک سری بازرسی ها و آزمون هاست که در سرتاسر فرآیند نرم افزار به کار می روند تا اطمینان حاصل شود که هر محصول کاری خواسته های مربوط را برآورده میسازد.

استانداردهای ساخت و مستندسازی محصولات نرم‌افزاری

استاندارد ISO 9001

ISO 9001 یک استاندارد تضمین کیفیت است که در مهندسی نرم افزار کاربرد دارد . این استاندارد حاوی نیازمندی هایی لازم برای تضمین کیفیت یک سیستم نرم افزاری است . کنترل طراحی ، کنترل مستندات ، شناسایی محصول نرم افزاری ، کیفیت ، امکان سنجی و نیازمندی های وظیفه ای و غیر وظیفه ای ، بازبینی و آزمایش نرم افزار و بررسی کیفیت های داخلی از عناوینی است که در این استاندارد مورد توجه قرار می گیرد .

ارائه یک مثال :

طرح آژانس فضایی اروپا (ESA) و ارائه استاندارد نرم افزار این استاندارد تعریف کوتاهی از نحوه تولید نرم افزار مطلوب و با کیفیت قابل قبول ارائه می دهند. آنها مختصر و مفید و قابل درک بوده و بر اصولی عملی و دقیق استوار می باشند. آنها ضمن اینکه جنبه های اصلی و ضروری هر پروژه ای را در بر می گیرند، در قالب مجموعه ای از چارچوب ها و ضرورتها، حداکثر انتخاب ممکن را نیز برای مدیر پروژه فراهم می آورند. استانداردهای مهندسی نرم افزار ESA تحت نظارت هیئت کنترل و استانداردسازی نرم افزار وابسته به آژانس فضایی اروپا تنظیم و بازبینی شده است.

حوزه کاربرد

این استاندارد کلیه نرم‌افزارهای شرکتی یا صنعتی را تحت پوشش قرار می‌دهد. در این استاندارد، نرم‌افزار به عنوان برنامه‌ها، روال‌ها، قوانین و کلیه اسناد و مدارک مربوط به عملکرد و بهره‌برداری از یک سیستم کامپیوتری، تعریف شده است. این استانداردها کلیه وجوه نرم‌افزاری یک سیستم، به انضمام واسطه‌های آن با سخت‌افزار کامپیوتر و دیگر مؤلفه‌های سیستم را در بر می‌گیرند. نرم‌افزار ممکن است زیر سیستمی از یک سیستم پیچیده تر و یا سیستم مستقل باشد.

ساختار استاندارد

استانداردهای مهندسی نرم‌افزار ESA به سه بخش اصلی تقسیم می‌شوند.

✓ بخش اول، استانداردهای محصول Products :

استانداردها، توصیه‌ها، و رهنمودهای مربوط به محصول را شامل می‌شود.

✓ بخش دوم، استانداردهای رویه Procedure :

رویه‌های مورد استفاده در مدیریت یک پروژه نرم‌افزاری را تشریح می‌کند.

✓ بخش سوم، پیوست‌ها Appendices :

شامل خلاصه‌ها، جداول، فرمها، و فهرستی از روشهای ضروری می‌باشد.

سه گروه روش استاندارد در این مجموعه استفاده شده‌اند که عبارتند از :

✓ روشهای ضروری : در این روشها از عبارت "الزامی است" استفاده شده است.
از این روشها باید بدون هیچگونه استثنایی در کلیه پروژههای نرمافزاری پیروی شود.

✓ روشهای توصیه شده : در این روشها از عبارت "می‌بایست" استفاده شده است و
به روشهای قویاً توصیه شده اشاره می‌کند. عدم پیروی از این روشها نیاز به
توجیه مناسب دارد.

✓ رهنمودها : در این روشها از عبارت "ممکن است" یا "می‌توانند" استفاده شده
است و به رهنمودها اشاره می‌کنند. عدم پیروی از رهنمودها، نیازی به توجیه
ندارد.

استانداردهای مهندسی نرمافزار ESA ، کاربران را به استفاده از هیچیک از روشهای
ویژه مهندسی نرمافزار و یا ابزار نرمافزاری وادار نمی‌کنند، بلکه تنها روشهای
ضروری توصیه شده و رهنمودها را برای پروژههای مهندسی نرمافزار تشریح کرده و
به هر پروژه امکان می‌دهد که بهترین روش را جهت پیاده‌سازی انتخاب نماید.
در ادامه، توصیف خلاصه‌ای از استانداردهای مهندسی نرمافزار ESA ارائه می‌گردد.
همانطور که ذکر گردید این استانداردها در دو بخش ارائه شده است : استانداردهای
محصول و استانداردهای رویه که هر کدام به صورت اجمالی معرفی می‌گردد.

استانداردهای محصول

این بخش، کلیات چرخه حیات نرم افزار را تشریح می کند. در این استانداردها غالباً از عبارت "پروژه نرم افزاری" استفاده می شود. واضح است که تولید نرم افزار جنبه های سخت افزاری کامپیوتر را نیز در بر می گیرد. محصولات نرم افزاری لازم است با روش معینی طراحی و اجرا گردند. یک "مدل چرخه حیات"، فعالیتهای پروژه را در قالب مراحل مشخص سازماندهی می کند و تعیین می نماید کدام یک از فعالیتها باید در کدام مرحله انجام گیرد. بر اساس این استاندارد شش مرحله بایستی در چرخه حیات یک نرم افزار طی شود که عبارتند از:

- ✓ مرحله UR : تعیین نیازهای کاربر (User Requirements)
 - ✓ مرحله SR : تعیین نیازهای نرم افزار (Software Requirements)
 - ✓ مرحله AD : طراحی معماری (Architectural Design)
 - ✓ مرحله DD : طراحی تفصیلی و تولید برنامه (Detailed Design)
 - ✓ مرحله TR : انتقال و واگذاری نرم افزار برای بهره برداری (Transfer of Software the
 - ✓ مرحله OM : بهره برداری و نگهداری (Maintenance & Operation)
- چهار مرحله اول با یک بازبینی خاتمه می یابند. پنج مقطع کاری مهم، پیشرفت چرخه حیات نرم افزار را مشخص می نمایند که این مقاطع عبارتند از :

- ✓ تصویب سند نیازهای کاربر (URD)
- ✓ تصویب سند نیازهای نرم افزار (SRD)
- ✓ تصویب سند طراحی معماری (ADD)

✓ تصویب سند طراحی تفصیلی (DDD)، راهنمای کاربر نرم افزار (SUM) ،

اعلام آمادگی جهت پذیرش آزمونهایی پذیرش موقت

✓ اعلام پذیرش موقت و تحویل سند انتقال نرم افزار (STD)

آخرین مرحله کاری، نه در پایان این مرحله، بلکه در پایان دوره تضمین، خاتمه می یابد. این مقاطع به عنوان حداقل مراحل لازم جهت یک ارتباط کاری که باید در کلیه پروژه ها ارائه گردند انتخاب شده اند. در ادامه، هر یک از مراحل شش گانه بطور مختصر شرح داده می شود.

مرحله UR : تعیین نیازهای کاربر

مرحله UR را می توان "مرحله تعریف مسئله" چرخه حیات نامید. هدف این مرحله تدقیق یک فکر در مورد کاری که باید انجام گیرد در قالب تعریفی است از آنچه که از یک سیستم کامپیوتری انتظار می رود. نتیجه مرحله UR، سند نیازهای کاربر (URD) است. این سند برای کل پروژه نرم افزار سند مهمی به شمار می آید. زیرا مفاهیم اساسی که بر پایه آنها نرم افزار مورد پذیرش قرار می گیرد را تعریف می نماید.

ورودی ها :

هیچ گونه ورودی رسمی مورد نیاز نیست. اگرچه نتایج مصاحبه ها، بررسی ها، و مطالعات می توانند جهت تنظیم خواسته های کاربر مفید باشند.

فعالیت‌ها:

- ✓ دریافت نیازهای کاربر
- ✓ تعیین محیط عملیاتی
- ✓ تشخیص نیازهای کاربر
- ✓ بازبینی‌ها

خروجی‌ها :

- ✓ سند نیازهای کاربر
- ✓ طرح‌های آزمون پذیرش
- ✓ طرح مدیریت پروژه برای مرحله SR (محدوده پروژه، برآورد هزینه و طرح مدیریت)
- ✓ طرح مدیریت پیکربندی برای مرحله SR (روالهای مدیریت، ابزارهای CASE)
- ✓ طرح واریسی و اعتبار سنجی برای مرحله SR
- ✓ طرح تضمین کیفیت برای مرحله SR

مرحله SR : تعیین نیازهای نرم‌افزار

مرحله SR را می‌توان "مرحله تحلیل مسئله" چرخه حیات نامید. یک بخش ضروری در این مرحله، ساخت الگویی است که بیانگر آن باشد که نرم‌افزار چه کاری را باید انجام دهد، نه اینکه چگونه باید آنرا انجام دهد. در این حالت ممکن است ساخت نمونه‌های اولیه به منظور روشن نمودن نیازهای نرم‌افزار ضروری باشد.

ورودی‌ها:

تمامی خروجیهای مرحله قبل به عنوان ورودی در این مرحله استفاده می‌شوند.

فعالیت‌ها :

✓ ساخت مدل منطقی

✓ تشخیص نیازهای نرم‌افزار

✓ بازبینی‌ها

خروجی‌ها :

✓ سند نیازهای نرم‌افزار

✓ طرح‌های آزمون سیستم

✓ طرح مدیریت پروژه برای مرحله AD

✓ طرح مدیریت پیکربندی برای مرحله AD

✓ طرح واریسی و اعتبارسنجی برای مرحله AD

✓ طرح تضمین کیفیت برای مرحله AD

✓

مرحله AD : طراحی معماری

هدف مرحله طراحی معماری، تعیین ساختار نرم‌افزار است. الگوی ساخته شده در مرحله SR نقطه شروع این مرحله است. این الگو با تخصیص وظیفه‌مندی‌ها به مؤلفه‌های نرم‌افزار و تعیین گردش اطلاعات و عملیات بین آنها به طرح معماری نرم‌افزار تغییر می‌یابد. در این مرحله ممکن است طرح چندین بار تکرار شود. در واقع امکان استفاده از مدل‌های مختلف آبشاری، افزایشی و ... در این مرحله وجود دارد.

در این مرحله ممکن است نمونه‌سازی بخش‌های حساس نرم‌افزار جهت تأیید فرضیات طرح اصلی ضروری باشد.

ورودی‌ها :

کلیه خروجی‌های مرحله SR

فعالیت‌ها :

✓ ساختار مدل فیزیکی شامل (تجزیه نرم‌افزار به مولفه‌ها ، پیاده‌سازی الزامات غیر وظیفه‌مندی، معیارهای کیفیت طرح ، بررسی مزیت‌های نسبی مابین طرح‌های جایگزین)

✓ ساختار مدل معماری شامل (تعیین وظیفه مؤلفه‌ها، تعیین ساختمان داده‌ها، تعیین میزان استفاده از منابع کامپیوتری ، انتخاب زبان برنامه‌نویسی، بازبینی طرح

خروجی‌ها :

✓ سند طراحی معماری

✓ طرح‌های آزمون یکپارچگی

✓ طرح مدیریت پروژه برای مرحله DD

✓ طرح مدیریت پیکربندی برای مرحله DD

✓ طرح واریسی و اعتبارسنجی برای مرحله DD

✓ طرح تضمین کیفیت برای مرحله DD

مرحله DD : طراحی تفصیلی و تولید برنامه

هدف این مرحله، طراحی تفصیلی نرم افزار، برنامه نویسی، مستندسازی و آزمون آن است. سند طراحی تفصیلی و راهنمای کاربر نرم افزار همزمان با برنامه نویسی و آزمون آن تولید می شود. در آغاز، سند طراحی تفصیلی و راهنمای کاربر، بخش های متناظر با سطوح فوقانی سیستم را در بر می گیرند. همچنانکه طرح به سطوح پایین تر جریان می یابد، زیربخش های مربوطه افزوده می شوند. در پایان این مرحله، مستندات تکمیل می شوند و به همراه برنامه ها، خروجی های این مرحله را تشکیل می دهند.

ورودی ها :

تمامی خروجی های مرحله AD به عنوان ورودی استفاده می شود.

فعالیت ها :

✓ طراحی تفصیلی تولید شامل (برنامه نویسی، مجتمع سازی، آزمون ، بازبینی)

خروجی ها :

✓ برنامه ها

✓ سند طراحی تفصیلی

✓ راهنمای کاربر نرم افزار

✓ طرح مدیریت پروژه برای مرحله TR

✓ طرح مدیریت پی کربندی برای مرحله TR

✓ ویژگی های آزمون پذیرش

✓ طرح تضمین کیفیت برای مرحله TR

مرحله TR : انتقال و واگذاری نرم افزار براي بهره برداري

هدف مرحله TR نصب نرم افزار در محيط عملياتي است و نشان دادن اينكه كليه قابليت هاي تشریح شده در سند نیازهاي کاربر (URD) در نرم افزار مربوطه موجود است.

ورودي ها :

✓ همه خروجي هاي مرحله DD

فعاليت ها :

✓ نصب

✓ آزمون هاي موقت

خروجي ها :

✓ اعلاميه پذيرش موقت

✓ سيستم نرم افزاري پذيرفته شده موقت

✓ سند انتقال نرم افزار

مرحله OM : بهره‌برداری و نگهداری

در مرحله بهره‌برداری، ابتدا استفاده از نرم‌افزار آغاز می‌شود. بهره‌برداری از نرم‌افزار فراتر از حدود این استاندارد است. لذا این فصل فقط به بحث در خصوص نگهداری می‌پردازد. هدف از نگهداری نرم‌افزار اطمینان از آن است که محصول در طول دوره حیات خود کماکان نیازهای کاربر را برآورده خواهد نمود. خروجی عمده این مرحله سند تاریخچه پروژه (PHD) می‌باشد که مراحل تولید، بهره‌برداری و نگهداری محصول را به طور خلاصه بیان می‌کند.

ورودی‌ها :

✓ خروجی‌های مرحله TR

فعالیت‌ها :

✓ پذیرش نهایی

✓ نگهداری

خروجی‌ها :

✓ اعلامیه پذیرش نهایی

✓ سند تاریخچه پروژه

✓ سیستم نرم‌افزاری پذیرفته شده نهایی

✓ استانداردهای رویه

بخش دوم این استاندارد، فعالیتهایی را که برای مدیریت چرخه حیات نرم افزار بسیار ضروري است، تشریح می کند. هدف از مدیریت فعالیتهای تولید محصول در قالب بودجه تعیین شده، بر طبق زمانبندی و با کیفیت لازم است. به منظور دستیابی به این هدف باید طرحهایی برای موارد زیر تهیه شوند :

- ✓ مدیریت پروژه نرم افزار
- ✓ مدیریت پیکر بندی نرم افزار
- ✓ واریسی و اعتبار سنجی نرم افزار
- ✓ تضمین کیفیت نرم افزار

مدیریت پروژه نرم افزار

مدیریت پروژه نرم افزار (Software Project Management) "فرآیند طراحی، سازماندهی، تعیین کارکنان، نظارت، کنترل، رهبری و هدایت یک پروژه نرم افزاری" می باشد که در استاندارد ANSI/IEEE Std 1058.1-1987 تعریف شده است.

فعالیت ها:

- ✓ سازماندهی پروژه
- ✓ رهبری پروژه
- ✓ مدیریت ریسک
- ✓ مدیریت فنی
- ✓ برنامه ریزی، زمانبندی و بودجه بندی کار
- ✓ گزارش پیشرفت پروژه

طرح مدیریت پروژه نرم افزار سندی است که مدیریت يك پروژه نرم افزاري را کنترل می کند و به چهار قسمت شامل طرحهاي مدیریت مراحل SR، AD، DD و TR تقسیم می گردد که جزئیات مندرجات آن در استاندارد درج گردیده است.

مدیریت پیکربندی نرم افزار

مدیریت پیکربندی نرم افزار عبارت است از یک فعالیت جامع که در طول فرایند نرم افزاری اجرا می گردد. از آن جایی که تغییر می تواند در هر زمانی رخ دهد، فعالیت های SCM برای شناسایی تغییر، کنترل تغییر، حصول اطمینان از این که تغییر به درستی اجرا می شود، گزارش تغییر به کسانی که علاقمند به دانستن آن هستند، انجام می گردد. SCM اصلاحاتی را که همواره به وقوع می پیوندد، در زمان ساخت نرم افزار و پس از ارائه آن به مشتریان، شناسایی کنترل، بررسی و گزارش می نماید. بین پشتیبانی از نرم افزار و مدیریت پیکربندی نرم افزار تفاوت وجود دارد. پشتیبانی یعنی مجموعه ای از فعالیت های مهندسی نرم افزار که پس از تحویل نرم افزار به مشتری و استفاده از آن به وقوع می پیوندد. مدیریت پیکربندی نرم افزار یعنی مجموعه ای از فعالیت های پیگیری و کنترل که زمانی شروع می شوند که یک پروژه مهندسی نرم افزاری آغاز می گردد و تنها زمانی پایان می یابند که نرم افزار از کار می افتد.

متغیر دیگری در فرایند نرم افزاری وجود دارد بنام "تغییر". تغییر می تواند در هر زمانی و به هر دلیلی روی دهد. در واقع اولین قانون مهندسی سیستم بیان می دارد که: مهم نیست که شما در کجای چرخه زندگی سیستم قرار دارید، سیستم تغییر خواهد کرد و تمایل به تغییر آن در تمام چرخه زندگی ادامه پیدا خواهد کرد.

چهار منبع تغییر اساسی وجود دارد:

✓ شرایط جدید کاری و یا تجاری که تغییر در شرایط محصول و یا قوانین تجاری را دیکته می کند

✓ نیازهای جدید مشتریان که اصلاحات داده هایی را که توسط سیستم های اطلاعاتی ایجاد می شوند، عملکردی که توسط محصولات عرضه می شوند، و یا خدماتی که توسط یک سیستم کامپیوتری ارائه می شوند را عرضه می نمایند

✓ سازماندهی دوباره و یا گسترش/کاهش تجارت که سبب بروز تغییرات در اولویت پروژه و یا ساختار تیم مهندسی نرم افزار می شود.

✓ مشکلات مربوط به بودجه و یا برنامه ریزی که سبب تعریف دوباره سیستم و یا محصول می گردد.

مدیریت پیکربندی تکنیکی است که در مدیریت کردن بهبود ویژگی و روش های کار در پروژه های توسعه ای مورد استفاده قرار می گیرد. این تکنیک در دهه 1950 در صنایع دفاعی امریکا بوجود آمد و امروزه تبدیل به ابزاری نسبتاً ضروری در کنترل کارایی و کیفیت اجزا در ساختار شکست محصول و روش های کار در ساختار شکست کار شده است و در انواع مختلف پروژه ها قابل کاربرد می باشد. مدیریت پیکربندی ویژگی های ساختار شکست محصول را کنترل می کند، و محصولی که توسط یک پروژه تولید می شود را به عنوان ترکیبی از اجزا معرفی می کند، که این پیکربندی می تواند به اشکال مختلفی باشد: یک ماشین، یک فضاپیما، نقشه، سیستم های نرم افزاری و غیره. هر جزیی از این اجزا پیکره بندی را می توان به عنوان یک پیکربندی در نظر گرفت که خود از اجزای دیگری تشکیل شده است.

مدیریت پیکربندی یک سری از روش های کاری مناسب می باشد که به منظور غلبه کردن بر عدم یقین ها و تغییرها و بدست آوردن تعهد ارکان های مختلف پروژه در حین این که پروژه در حال شکل گیری می باشد، مورد استفاده قرار می گیرد. به منظور این که مدیریت پیکربندی بخصوص در پروژه های پیچیده موثر باشد، باید دارای رویکردی استوار و سیستماتیک باشد و از همان آغاز، ساختار به گونه ای باشد که از ان حمایت کند، که این بدین معنی می باشد که افراد مشخصی باید مسئولیت مدیریت پیکربندی را بر عهده داشته باشند و همچنین وجود یک سری از روش های کاری که مورد تایید مدیریت ارشد باشد. وجود یک شورای نظارت با مسئولیت تایید ویژگی های محصول و تایید تغییرات ویژگی های محصول ضروری است.

مدیریت پیکربندی نرم افزار عبارت است از نظام به کارگیری مسیر فنی و اجرایی و نظارت بر :

- ✓ تعیین و مستندسازی مشخصات فیزیکی و وظیفه مندی يك عنصر پیکربندی
- ✓ کنترل تغییرات حاصله در مشخصات مزبور
- ✓ ثبت گزارش فرآیند تغییرات و وضعیت پیاده سازی
- ✓ بررسی پذیرش محصول با توجه به نیازهای تعیین شده
- ✓ تشخیص پیکربندی
- ✓ ذخیره سازی عنصر پیکربندی
- ✓ کنترل و نظارت بر تغییر پیکربندی
- ✓ گزارش وضعیت پیکربندی

طرح مدیریت پیکربندی نرم افزار به چهار قسمت برای مراحل SR ، AD ، DD ، و TR تقسیم می شود.

وارسي و اعتبار سنجي نرم افزار

اصطلاح "وارسي" بسته به مفهومي كه از آن انتظار مي رود، داراي معاني متعددي است كه سه معنای متداول آن عبارتند از :

✓ عمل بازبيني، بازرسى، آزمون، بررسى، ممیزی و به عبارت دیگر تصدیق و مستندکردن آنكه آیا عناصر فرآیندها، خدمات یا مستندات منطبق با نیازهای تعیین شده هستند یا خیر.

✓ فرآیند ارزیابی يك سیستم یا يك مؤلفه به منظور تعیین آنكه آیا محصولات يك مرحله تولید مشخص، شرایط وضع شده آغازین آن مرحله را ارضا می نمایند یا خیر.

✓ اثبات رسمي صحت برنامه

اعتبارسنجي نیز بنا بر تعریف ANSI/IEEE عبارت است از "ارزیابی نرم افزار در پایان تولید به منظور حصول اطمینان از رعایت نیازهای کاربر". بنابراین اعتبارسنجي، وارسي نهایی است.

فعالیت ها :

- ✓ بازبيني ها و بررسى هاي فني و بازرسى هاي نرم افزار
- ✓ بررسى اينكه نیازهای نرم افزار متناسب با خواسته های کاربر قابل ردیابی هستند
- ✓ بررسى اينكه مؤلفه های طراحی متناسب با نیازهای نرم افزار قابل ردیابی هستند
- ✓ بررسى تأییدیه های رسمي و الگوریتم ها
- ✓ آزمون واحدها
- ✓ آزمون یکپارچگی
- ✓ آزمون سیستم

✓ آزمون پذیرش

✓ ممیزی

طرح واریسی و اعتبار سنجی نرم افزار در هفت بخش که شامل برنامه های واریسی برای مراحل SR، AD، DD و شرح مشخصات آزمون های واحد، یکپارچگی، سیستم و پذیرش می باشند تقسیم می گردد.

تضمین کیفیت نرم افزار

تضمین کیفیت نرم افزار (SQA: Software Quality Assurance) عبارت است از "طرح برنامه ریزی شده منظم از کلیه عملیات لازم برای حصول اطمینان کافی و مناسب در خصوص انطباق عنصر یا محصول تولید شده با مشخصات فنی مورد نظر". فعالیت تضمین کیفیت، فرآیند اثبات صحت به کارگیری این استانداردهاست. در پروژه های کوچک این فعالیت می تواند توسط گروه تولید نرم افزار انجام پذیرد ولی در پروژه های بزرگ، افراد مشخص و متخصص می بایست برای انجام این وظیفه منسوب گردند.

فعالیت ها :

✓ مدیریت مستندسازی

✓ نظارت بر استانداردها، رویه ها، پیمان ها و معیارها

✓ بازبینی ها و ممیزی ها

✓ فعالیتهای آزمون

✓ گزارش مشکلات و انجام اصلاحات

✓ بررسی مناسب بودن ابزارها، شیوه ها و روش ها

- ✓ کنترل برنامه‌ها و رسانه‌ها
- ✓ کنترل پیمانکارها
- ✓ گردآوری، ذخیره‌سازی و نگهداری گزارشات
- ✓ آموزش
- ✓ مدیریت ریسک

توجه به کیفیت نرم افزار ها و بکارگیری فن آوری های نوین برای تعالی آن ، زیر ساخت و الزام مورد نیاز جهت تولید نرم افزارهای مهندسی ساز ، مشتری پسند و کاربردی است که نقش بسزائی در روان سازی بازار صادرات این گونه محصولات ایفا می کند . اهداف اصلی رعایت شده در تضمین کیفیت نرم افزار و اثرات آن بر چرخه تولید و صادرات آنها به شرح ذیل ذکر می گردد :

- ✓ صرفه جویی در هزینه های تولید ، توسعه و نگهداری نرم افزار های کاربردی که توسط کارشناسان داخلی طراحی و پیاده سازی می گردد . منظور از این صرفه جویی هم در ساختار داخلی کشور است و هم در سطح بین المللی .
- ✓ پیاده سازی نرم افزارهای با کیفیت بالا .
- ✓ پیاده سازی نرم افزار های با ارزش افزوده بالا .
- ✓ قابلیت ساخت و طراحی نرم افزار بر اساس استانداردهای جهانی .
- ✓ قابلیت ساخت و طراحی نرم افزار های شکست ناپذیر .

✓ آمادگی برای ورود به بازارهای جهانی نرم افزار از طریق رعایت قوانین و استانداردهای حاکم بر فرآیند مهندسی نرم افزار

طرح تضمین کیفیت نرم افزار در چهار مرحله UR، SR، AD و DD انجام می گیرد.

استاندارد IEEE/EIA 12207

مؤسسه های IEEE (Institute of Electrical and Electronic Engineers) و EIA (Electronic Industries Association) سازمانهای معتبر تدوین کننده استاندارد هستند. این دو سازمان برای تدوین استانداردهای مهندسی نرم افزار در آمریکا با سازمان صنایع دفاع آمریکا (Department of Defense) همکاری می کنند. سه هدف اساسی از تبیین استاندارد ISO/IEC 12207 در آمریکا مطرح بوده است که عبارتند از :

- ✓ استاندارد بایستی سیستم های تجاری را به بهترین نحو پوشش دهد
 - ✓ استاندارد بایستی با برنامه های کاربردی با نیازهای پیچیده و شرایط سیستم های دفاعی سازگاری داشته باشد
 - ✓ استاندارد بایستی سازگاری خود را به منظور فروش جهانی نرم افزار حفظ کند.
- بر مبنای این اهداف دو استاندارد در این سازمانها تدوین شد که عبارتند از :

IEEE / EIA 12207

يك استاندارد استراتژيك می باشد که اهداف ذکر شده در پاراگراف قبل را می پوشاند. این استاندارد برگرفته از استاندارد بین المللی ISO / IEC 12207 می باشد و می تواند برای سیستمهای تجاری و نظامی هم برای مشتریان داخلی و هم برای مشتریان خارجی استفاده شود.

EIA/IEEE J-STD-016

يك استاندارد تاکتیکی می باشد که برای سازمانهایی که در پروسه های نرم افزاری تحت استانداردهای نظامی سرمایه گذاری می کنند تهیه شده است. يك تفاوت اساسی میان استانداردهای نظامی و استانداردهای تجاری این است که استانداردهای نظامی معمولاً به صورت قبول داوطلبانه می باشند در صورتیکه استانداردهای تجاری بر مبنای قرارداد فی مابین صورت می گیرد.

معرفی کلی استاندارد IEEE/EIA 12207

همانطور که در قسمتهای قبل اشاره شد ISO/ IEC 12207 به عنوان يك استاندارد بین المللی در سال 1995 ارائه گردید. این استاندارد يك چهارچوب برای پروسه های چرخه حیات نرم افزار ارائه می نماید و برای استفاده در تقاضا، عرضه، ساخت، نگهداری و عملیات نرم افزار بسیار مناسب می باشد.

این استاندارد بصورت عمومی و کلی ارائه شده است و کشورهای آمریکا، ژاپن، و تعدادی از کشورهای اروپایی به منظور پیاده‌سازی این استاندارد در کشورهای خود اقدام به اعمال تغییراتی در آن نموده‌اند. در این راستا، یک گروه کاری از مؤسسات IEEE و EIA در آمریکا اقدام به آماده‌سازی استاندارد 12207 برای بکارگیری آن در کشور آمریکا کردند. به منظور حفظ سازگاری استاندارد IEEE/EIA 12207 در آمریکا با نسخه ISO آن، بجای تمرکز بر روی سطوح پروژه‌های مجزا، سطوح سازمانی مدنظر قرار گرفت بدین صورت که پیشنهاد شد سازمانها، مجموعه پروسه‌ها و رویه‌های خود را با اصول IEEE/EIA 12207 تطبیق دهند. استاندارد IEEE/EIA 12207 منافع و مزایای مهمی را برای استانداردهای موجود و در حال استفاده در زمینه پروسه‌های چرخه حیات ارائه نمود که می‌توان به چند مورد آن اشاره کرد :

✓ این استاندارد در مقایسه با روشهای نوعی که فقط پیاده‌سازی پروسه‌ها را مدنظر قرار داده اند، می‌تواند کل چرخه حیات را بپوشاند.

✓ برخلاف روشهای جاری که ثبت رویه‌ها و داده‌های تولید شده بر روی مستندات کاغذی انجام می‌شود این انعطاف پذیری را دارد که روش ثبت را به گونه ای تغییر دهد که قابل استفاده توسط ابزار CASE باشند.

✓ این استاندارد قابلیت همکاری با دیگر استانداردهای تدوین شده در آمریکا را به منظور پیاده‌سازی جزئی تر فراهم می‌سازد.

✓ این استاندارد می‌تواند پروسه‌ها و داده‌های هدفمندی را تولید کند که برای سازگاری نیازها در شرایط غیر معمول مفید خواهد بود.

✓ این استاندارد سازگاری خود را با روشهای ISO 9000 برای سیستمهای کیفیت، مدیریت کیفیت و تضمین کیفیت حفظ می کند.

✓ سازگاری کامل این استاندارد با نسخه بین المللی باعث میشود که شرکت های آمریکایی یک مجموعه واحدی از پروسه ها را تولید کنند که می تواند هم برای پروژه های جهانی و هم برای پروژه های داخلی استفاده گردد.

ساختار استاندارد IEEE/EIA12207

استاندارد IEEE/EIA 12207 در سه بخش 12207.0، 12207.1، و 12207.2 ارائه شده است. بخش 12207.0 در واقع یک نسخه از همان استاندارد ISO/IEC 12207 می باشد که مطالب الحاقی به شرح زیر به آن اضافه شده است:

✓ یک ضمیمه که لیست خطاهای کشف شده در متن بین المللی استاندارد در آن ارائه شده است.

✓ مجموعه ای از Data objectives و Process objectives که بصورت مشخص، داده های (کاغذی یا الکترونیکی) تولید شده توسط پروسه های مشخص شده در استاندارد را تعیین می کنند.

بخش 12207.1

يك متن راهنمایی است که توصیه‌هایی را برای objectives Data ارائه میکند. برای کسانی که علاقه دارند ارتباط بین پروسه‌های مختلف با مستندات یا داده‌های الکترونیکی را تنظیم نمایند این بخش راهنمای مفیدی خواهد بود.

بخش 12207.2

يك متن راهنمایی است که توصیه‌هایی را در رابطه با نحوه پیاده‌سازی پروسه‌ها ارائه می‌کند. سازماندهی مطالب در این بخش به صورتی است که ابتدا عین رویه‌های بخش 12207.0 در داخل يك کادر آورده می‌شود، سپس بخش راهنمایی شامل دوره زمانی مناسب، نکات راهنمایی، و توصیه‌های پیاده‌سازی ارائه می‌گردد.

بخش 12207.0

در قسمت معرفی استاندارد ISO/ IEC 12207 ارائه گردید. موارد الحاقی به این بخش شامل چند متمم و ضمیمه می‌باشد که فهرست آنها به شرح زیر می‌باشد.

A - Annex : مناسب سازی (Tailoring) پروسه‌ها

B - Annex : راهنمای Tailoring

C - Annex : راهنمایی برای پروسه‌ها و سازمانها

D - Annex : راهنمای مراجع

E - Annex : مفاهیم پایه‌ای در ISO/IEC 12207

Annex - F : موافقت

Annex - G : اهداف پروسه‌هاي Life cycle

Annex - H : اهداف داده‌هاي Life cycle

Annex - I : ارتباطات

Annex - J : خطاها و تغییرات

همانطور که اشاره گردید بخش 12207.1 برای راهنمایی ثبت داده‌های منتج شده از بخش اول (12207.0) یابه عبارت دیگر چرخه داده‌ای تدوین شده است. این بخش شامل شش زیر بخش می‌باشد که عبارتند از:

✓ حوزه استاندارد

✓ ارجاعات

✓ تعاریف

✓ چرخه حیات داده (توسعه Annex H مربوط به 12207.0)

✓ محتوای عناصر اطلاعاتی عمومی شامل :

- توصیف

- طرح

- رویه

- ثبت

- گزارش

- درخواست

- جزئیات و مشخصات

بخش 12207.2 براي راهنمايي بكارگيري پروسه‌هاي بخش 12207.0 تدوين شده است. اين بخش از استاندارد نيز شامل چندين زير بخش و چندين Annex مي‌باشد كه عناوين آنها عبارتند از :

Annex - A تا Annex - F :

راهنمايي ضمايم متناظر اضافه شده در 12207.0
11 : Annex – G نوع بازبيني (review) را پيشنهاد مي‌كند كه شامل موارد زير مي‌باشد :

- ✓ بازبيني برنامه‌ريزي نرم‌افزار (Software Plan Reviews)
- ✓ بازبيني مفاهيم عملياتي ((Operational Concept Reviews)
- ✓ بازبيني نيازهاي سيستم/ زير سيستم (System / subsystem Reviews)
- ✓ بازبيني طراحي سيستم / زير سيستم (System / subsystem Design Reviews)
- ✓ بازبيني نيازهاي نرم‌افزار (Software Requirement Reviews)
- ✓ بازبيني طراحي نرم‌افزار (Reviews Software Design)
- ✓ بازبيني آماده بودن آزمايشها (Test Readiness Reviews)
- ✓ بازبيني نتايج آزمايشها (Test Result Reviews)
- ✓ بازبيني قابليت استفاده از نرم‌افزار (Software Usability Reviews)
- ✓ بازبيني نگهداري از نرم‌افزار (Maintenance Reviews Software)
- ✓ بازبيني نيازهاي بحراني (Reviews Critical Requirements)

Annex - H : طبقه‌بندی کمی نرم‌افزار که شامل طبقه‌بندی‌های زیر می‌باشد :

- ✓ برنامه‌ریزی و پیشرفت
- ✓ منافع و هزینه‌ها
- ✓ رشد و ثبات
- ✓ کیفیت تولیدات
- ✓ کارایی توسعه نرم‌افزار
- ✓ تکافوی تکنیکی

Annex - I : راهنمای استراتژی‌های توسعه و طراحی بر مبنای build

در این قسمت مدل‌های ساخت نرم‌افزار اعم از آبشاری، افزایشی، و... مقایسه و بررسی می‌شوند.

Annex - J : طبقه‌بندی اولویت‌ها برای گزارش مشکلات (اولویت‌های 1 تا 5 برای بیان

مشکلات تعیین شده است.)

Annex - K : ارزیابی تولیدات نرم‌افزاری. این قسمت 11 معیار برای ارزیابی محصولات نرم‌افزاری ارائه می‌کند که عناوین آنها عبارتند از :

- ✓ دقت
- ✓ کافی بودن موارد آزمون، رویه‌ها، داده‌ها و نتایج
- ✓ سازگاری با بقیه تولیدات
- ✓ پوشش
- ✓ امکان سنجی
- ✓ پیروی از طرح نرم‌افزار
- ✓ ثبات داخلی
- ✓ برآورده کردن نیازها
- ✓ آزمون پذیری
- ✓ قابل فهم بودن
- ✓ نمایش

Annex - L : مدیریت ریسک. این قسمت شامل پیشنهاداتی برای طرح ریسک، مشخص‌سازی، آنالیز و کنترل ریسک می‌باشد.

Annex - M : مرجع پروسه‌های چرخه حیات. پروسه‌های چرخه حیات را برای ISO 12207 (/ IEC 12207) بیان می‌کند.

طرح تضمین کیفیت نرم افزار (Software Quality Assurance Plan)

طرح تضمین کیفیت نرم افزار (SQAP) نحوه نظارت بر رعایت این استانداردها را معین می کند. فهرست مندرجات SQAP، فهرستی از فعالیت هایی است که باید به منظور حصول اطمینان از کیفیت محصول انجام پذیرند. برای هر فعالیت می بایست مسئولین تضمین کیفیت نرم افزار طرح های لازم برای نظارت بر آن فعالیت را به دقت تشریح نمایند. گواهی رعایت این استانداردها می بایست برای کلیه مراحل چرخه حیات نرم افزار درخواست شود. مستنداتی که توسط این مجموعه از استانداردها ضروری تشخیص داده شده اند می بایست فراهم گردیده و تامین شوند. متن اصلی برنامه ها می بایست برای اطمینان از رعایت استانداردهای برنامه نویسی مورد بررسی قرار گیرند. هر جا که امکان پذیر است می بایست با بهره گیری از معیارهای قابل قبول، جنبه های گوناگون کیفیت (مانند پیچیدگی، قابلیت اطمینان، نگهداشت پذیری، ایمنی، میزان نواقص، میزان مشکلات، میزان مغایرت ها) به طور کمی اندازه گیری شوند. بخش های بعدی به مجموعه فعالیت هایی اختصاص دارند که از ANSI/IEEE Std 730 استنتاج گردیده اند و اجرا و رعایت آنها برای آنکه يك عنصر نرم افزاری متناسب با هدف خود باشد ضروری است. هر بخش به بحث در خصوص نحوه بررسی یکی از فعالیت ها می پردازد.

مدیریت

یکی از فعالیت های تصمیم کیفیت نرم افزار ، تحلیل ساختار مدیریتی است که بر کیفیت نرم افزار و نحوه نظارت بر آن اثر می گذارد . وجود یک ساختار سازمانی مناسب می بایست مورد بررسی قرار گیرد و تایید شود که افراد یا هر یک از واحدهای مجزای آن ساختار دارای وظایف و مسئولیت های کاملاً تعریف شده ای هستند . سازمان ، وظایف و مسئولیت ها باید در طرح مدیریت پروژه نرم افزار آورده شوند .

مستند سازی

کلیه رویه های مستند سازی که در طرح مدیریت پروژه نرم افزار تعیین گردیده اند می بایست رعایت شوند. هر گونه انحراف از برنامه مستند سازی تعریف شده در این مجموعه استانداردها می بایست با مدیر پروژه مورد بحث و بررسی گیرد .

استانداردها ، رویه ها ، قرار ها و معیارها

رعایت کلیه استانداردها ، رویه ها و قرار ها می بایست تحت نظارت و کنترل قرار گیرد . انحرافات و عدم تبعیت از موارد مذکور می بایست یادداشت شده و جهت ملاحظه مدیر پروژه ارسال شود . کارکنان تضمین کیفیت نرم افزار می توانند مدیر پروژه را با تعبیر و تفسیر صحیح استانداردها ، رویه ها و قرار ها یاری رسانند . ”معیار “ ، یک مقیاس کمی برای اندازه گیری میزان یک مشخصه تعیین شده ، در یک سیستم ، مولفه یا فرآیند است . معیارها برای مدیریت موثر بسیار ضروری هستند. معیارها باید ساده باشند تا به سهولت فهمیده شوند و همچنین کاربردی باشند تا مفید واقع گردند .

معیارهای اندازه گیری کیفیت، به ویژه قابلیت اطمینان و نگهداشت پذیری ، می بایست در سند نیازهای نرم افزار مشخص شوند. این معیارها می بایست برای کاربران معنی دار بوده و منعکس کننده خواسته های آنان باشند. معیارهای اضافی مورد نیاز می توانند در پروژه تعریف شوند . به عنوان مثال به منظور محدود کردن پیچیدگی طرح ، می توان میزان معیارهای پیچیدگی را در استانداردهای طراحی تعیین نمود . معیارها می توانند به عنوان راهنمایی برای تصمیم گیری در طرح مدیریت پروژه نرم افزار آورده شوند (مثلاً اگر در آزمون يك مولفه نرم افزاري بیش از سه مورد اشکال مشاهده شد ، باید مورد بررسی مجدد قرار گیرد) . معیارها می توانند بر حسب اهداف پروژه تغییر کنند ، بنابراین هرکدام از اهداف پروژه می بایست معیارهایی به همراه خود داشته باشند ، در غیر این صورت ممکن است وزن غیر ضروری به معیارهای تعریف شده داده شود . به عنوان مثال ، پروژه ای که تعداد خطوط برنامه نوشته شده را محاسبه کرده ، اما میزان خرابی را اندازه گیری نمی کند ، احتمالاً منجر به تولید يك برنامه حجیم و بزرگ ولي غیر قابل اعتماد خواهد شد .

بازبینی ها و ممیزی ها

استانداردهای تضمین کیفیت ، باز بینی های سند نیازهای کاربر ، سند نیازهای نرم افزار ، سند طراحی معماری ، سند طراحی تفصیلی ، طرح واریسی و اعتبار سنجی نرم افزار ، طرح مدیریت پیکربندی نرم افزار و همچنین بازبینی و ممیزی برنامه ها در حین تولید نرم افزار را ضروری می داند . به مراحل بازبینی و ممیزی می بایست مطابق آنچه که در SVVP تشریح گردیده توجه شود.

بازبینی های متنوعی (مانند بازبینی فنی ، بازرسی و بررسی) می توانند انجام پذیرند ، اگر چه می بایست ابتدا مناسب بودن مکانیزم ها با نوع پروژه مورد قبول قرار گیرند . کارکنان تضمین کیفیت نرم افزار می بایست در فرآیند بازبینی شرکت نمایند .

فعالیت های آزمون (تست)

انجام آزمون های واحد ، یکپارچگی ، سیستم و پذیرش برای حصول اطمینان از کیفیت نرم افزار نهایی قابل اجرا ضروری است . برنامه ها ، طرح ها ، رویه ها و گزارش های آزمون در طرح واریسی و اعتبار سنجی نرم افزار تشریح می شوند و می بایست توسط کارکنان تضمین کیفیت نرم افزار مورد بازبینی قرار گیرند . آنها می بایست بر انجام فعالیت های آزمون ، از جمله آزمون اجرا ، که توسط گروه تولید نرم افزار صورت می پذیرند نظارت نمایند . علاوه بر آن ، آزمون های دیگری نیز می توانند در طرح تضمین کیفیت نرم افزار پیشنهاد شده و توسط کارکنان تضمین کیفیت به اجرا گذاشته شوند .

گزارش مشکلات و انجام اصلاحات

رویه مدیریت مشکلات که باید در این استاندارد ها ایجاد گردد ، به منظور گزارش و پیگیری مشکلات از مرحله تشخیص تا رفع و حل نهایی آنها ، طراحی می گردد . کارکنان تضمین کیفیت نرم افزار می بایست بر اجرای رویه های تشریح شده در طرح مدیریت پیکربندی نرم افزار نظارت داشته و تعداد دفعات بروز خطا را مورد توجه قرار دهند .

ابزارها ، شیوه ها و روش ها

استانداردهای تضمین کیفیت ایجاب می نماید که ابزارها ، شیوه ها و روش های تولید نرم افزار متناسب با سطح پروژه ، تعیین و به کار گرفته شوند . بررسی مناسب بودن ابزارها ، شیوه ها و روش های انتخابی و نظارت بر کاربرد صحیح آنها ، از جمله فعالیت های تضمین کیفیت نرم افزار محسوب می شود . کارکنان تضمین کیفیت نرم افزار می توانند جهت نظارت دقیق تر خود ، در خصوص ضرورت استفاده از ابزارها ، شیوه ها و روش های اضافی تصمیم گیری نمایند ، که می بایست نتایج آن در طرح تضمین کیفیت نرم افزار آورده شود .

کنترل برنامه ها و رسانه ها

استانداردهای تضمین کیفیت ایجاب می نماید که روال هایی برای روش ها و تسهیلات استفاده شده به منظور نگهداری ، ذخیره سازی ، ایمنی و مستند سازی نسخ نهایی نرم افزار مورد نظر در طرح مدیریت پیکربندی نرم افزار تعریف و تعیین شوند . کارکنان تضمین کیفیت نرم افزار می بایست مناسب بودن روال های تعریف شده در طرح مدیریت پیکربندی نرم افزار و نحوه اجرای آنها را مورد بررسی قرار دهند .

کنترل پیمانکار ها

اقدام نرم افزاری تهیه شده از طریق پیمانکاران خارج از گروه تولید نرم افزار همواره باید در مقابل رعایت استانداردهای تعیین شده در پروژه مورد ارزیابی قرار گیرند . هر یک از پیمانکارها باید یک طرح تضمین کیفیت تهیه و ارائه نمایند . نرم افزار های تجاری ارائه شده به بازار نیازی به طرح تضمین کیفیت نرم افزار ندارند .

گردآوری ، ذخیره سازی و نگهداری گزارشات

این استانداردها مجموعه مستنداتی را که باید در هر پروژه تهیه و تولید شوند تعیین می نمایند . ضمن آنکه مستندات اضافی مانند صورت جلسات و گزارشات باز بینی هم می توانند ارائه گردند . کارکنان تضمین کیفیت نرم افزار باید مناسب بودن روش ها و تسهیلات استفاده شده در گرد آوری ، حفظ و نگهداری کلیه مستندات مذکور را ، دست کم برای دوره حیات پروژه ، مورد بررسی قرار دهند . روال های کنترل مستندات در طرح مدیریت پیکربندی آورده می شوند .

آموزش

کارکنان تضمین کیفیت نرم افزار می بایست بررسی کنند که کارکنان گروه تولید نرم افزار به طرز مناسبی برای انجام وظایف خود آموزش دیده اند و هر گونه آموزش دیگری که ضروری می دانند را نیز مشخص نمایند . برنامه های آموزشی در طرح مدیریت پروژه نرم افزار آورده می شوند .

مدیریت خطر (ریسک)

کلیه پروژه ها باید عوامل مهم و حیاتی در پیشرفت و موفقیت خود را تعیین و کنترل نمایند . این عمل " مدیریت خطر " نامیده می شود . مدیر پروژه باید همواره خطراتی را که بر پروژه اثر می گذارند تحلیل کرده و یافته های خود را در طرح مدیریت پروژه نرم افزار مستند نماید . کارکنان تضمین کیفیت نرم افزار می بایست بر فعالیت مدیریت خطر نظارت کرده و در خصوص روش ها و روال های تشخیص ، ارزیابی ، نظارت و کنترل محدوده های خطر توصیه های لازم را به مدیریت پروژه ارائه نمایند .

منابع

www.SQA.net

www.wikipedia.com

www.well.com

bagerian.blogfa.com

standards.ieee.org

www.amazon.com