

## ترجمه صفحه ۷۳ تا ۷۷ کتاب pressman

### جدول ۵,۳ وظایف نرم افزار برنامه ریزی

موقعیت	وظایف نرم افزار برنامه ریزی
1.5.1, 5.1.1, 5.3.2, 6.6.3, 10.1, 13.3	روش های توسعه و نگهداری SDP
4.4, 5.1.10, 10.2, 9.5, 12.5, 13.4	مدیریت داده های نرم افزار (DM)
3.2.2, 3.4, 3.7, 5.1.3, 9.5.1, 13.4, Chapter 14	اندازه نرم افزار و برآورد منابع
3.4, 5.1.4, 8.2, 8.7	برنامه ریزی ساخت نرم افزار
2.3.9, 3.6, 3.7, 10.3, 8.5, 8.7	برنامه ریزی یکپارچه سازی و آزمایش نرم افزار (T & ۱)
6.3, 12.1, 12.3, 12.11	توسعه نرم افزار
5.1.9, 8.8, 8.9, 9.8.2, 16.2, 16.3	محیط و ابزارهای پشتیبانی
2.4.3, 5.1, 6.3, 6.4, 8.6, 12.3, 16.1	پذیرش نرم افزار ، تحویل ، نصب ، انتقال ، عملیات ، برنامه ریزی
2.4.5, 6.1.2, 6.2, 6.5, 13.4	پایداری و از رده خارج شدن
2.3.5, 2.4.2, 5.1, 6.2, 10.8.2, 13.4	مدیریت پیکربندی نرم افزار
5.1, 5.2.3, 6.2, 6.4, 10.3.4, 10.4.5, 8.5.8, 9.10.5, 12.4, 15.2	ارزیابی نرم افزار با بررسی های رسمی و غیررسمی
1.6.3, 2.3.2, 5.2, 6.4.3, 6.6.3, 9.5.2, 9.6.3, 13.4, 16.5	تضمین کیفیت نرم افزار (SQA)
1.5.1, 2.3.3, 3.10.8, 5.1.4, 6.1.2, 13.4, Chapter 15	روشهای حل مسئله و اقدام پیشگیرانه
3.10.2, 5.2.5, 5.4.1, 10.4.3, 9.5, 9.8.2	مدیریت ریسک نرم افزار
5.1.2, 5.5, 11.2, 11.3.2	معیارهای نرم افزاری که محصولات و فرآیندها را پوشش می دهد
2.3.4, 2.6, 4.4, 5.1, 8.6, 8.8, 9.8.2, 11.4, Appendix D	مسائل امنیتی و حریم خصوصی
1.5.1, 1.6, 8.1.4, 8.2.3, 8.4.3, 8.5.3, 8.6.3, 10.1, Appendix C	نظارت بر قراردادهای فرعی نرم افزار
10.1, 10.2, 10.3, 11.3	برنامه های نرم افزاری با وابستگی های متقابل حیاتی
2.3.7, 4.3.9, 6.1.3, 8.9.9, 10.5, 10.8.4, 10.9, 11.3.5	سازمان نرم افزار ، نقش ها و مسئولیت ها
Chapter 16	منابع مورد نیاز ، مهارت ها و برنامه کارکنان
	برنامه های آموزشی و الزامات آموزشی
	عملیات و نگهداری نرم افزار

فرآیند اقدام اصلاحی (CAP). به روزرسانی های برنامه ریزی نشده SDP باید همانطور که در SDP یا برنامه اقدام اصلاحی توضیح داده شده است از طریق CAP انجام شود. همه تغییرات در SDP باید تأیید CSWE و برنامه SEPG را اجباری کنند. این کتاب راهنما فرض می کند که پروژه دارای CSWE یا موقعیت معادل آن است ، همانطور که در بخش ۱,۶,۴ شرح داده شده است.

تغییرات در SDP نیز باید تأیید CCB همانطور که در بخش ۶,۴,۳ بحث شده را ملزم نماید. SEPG در بخش ۶,۱,۱ شرح داده شده است.

بانک اطلاعات موجودیت نرم افزار (SWED). برای تهیه مکانیزمی برای شناسایی و ردیابی کلیه موارد نرم افزاری (SI) در پروژه ، باید یک پایگاه داده که می توان آن را Database Software Entity نامید ، تولید و به روز شود. هر زیر سیستم باید مسئول ورود داده های خود به SWED باشد ، اما CSWE باید مسئول جمع آوری این اطلاعات در یک پایگاه داده متمرکز و کنترل شده واحد برای پروژه یا برنامه باشد. این پایگاه داده ممکن است شامل هر SI در پروژه باشد که مجموعه ای از موارد زیر است:

شرح عملکردی SI ، کلاس ، گروه ، اندازه ، درصد کد جدید در مقابل استفاده مجدد ، توسعه دهنده (ها)ی مسئول و اطلاعات تماس آنها و زبان (های) مورد استفاده.

۵,۱,۳ برآورد منابع نرم افزار

منابع نرم افزاری ، شامل منابع فیزیکی ، پرسنلی ، هزینه ای و رایانه ای ، باید قبل از شروع توسعه نرم افزار تامین شود. این برآوردها برای ایجاد برنامه های توسعه نرم افزار ، برنامه های کاهش خطر و تعهدات استفاده می شود و باید در یک برنامه ارزیابی منابع نرم افزاری ثبت شود. جزئیات مدیریت منابع نرم افزاری در قسمت ۳ کتاب راهنما آورده شده است. پرسنل نرم افزار باید با سایر گروههای آسیب دیده (مهندسی سیستم ، SQA ، SCM ، آزمون و غیره) در برنامه ریزی کلی برنامه به عنوان اعضای تیم های محصول یکپارچه نرم افزار (SWIPT) شرکت کنند.

SWIPT ها در بخش ۱,۶,۵ بررسی شده. تعهدات ، یا تغییر در تعهدات ، افراد و گروه های خارجی باید به طور منظم با مدیریت بررسی شود.

برآورد کارکنان: برای تعیین سطح کارکنان مورد نیاز ، عملکرد برنامه ریزی باید محدودیت های برنامه شامل نقاط عطف ، بررسیها ، تحویل اسناد و محصولات ، نقاط عطف داخلی ، ساخت های افزایشی ، محدودیت های فنی و هرگونه تغییر در دامنه را در نظر بگیرد. برآورد Source Lines Of Code و بهره وری توسعه نرم افزار نقش مهمی در تخمین کارکنان مورد نیاز دارد.

برنامه ریزی مجدد: گروههای نرم افزاری باید در صورت لزوم در فعالیتهای برنامه ریزی مجدد برای رسیدگی به تغییرات قرارداد ، بهبود فرآیند یا هنگامی که عملکرد اندازه گیری شده از عملکرد برنامه ریزی شده متفاوت باشد ، شرکت کنند. داده های مربوطه تولید شده باید حفظ و در پرونده های توسعه نرم افزار (SDF) یا دفترچه مهندسی نرم افزار قابل استفاده باشند. پرسنل نرم افزار همچنین باید در فعالیت های اصلاح قرارداد / پیمانکاری فرعی (مانند پیشنهادهای تغییر مهندسی) شرکت کنند. فصل ۱۴ شامل اصول تخمین نرم افزار ، روش ها ، رشد اندازه و ابزار تخمین می باشد.

## ۵.۱.۴ برنامه ریزی ساخت نرم افزار

ساخت نرم افزار بخشی از سیستم است که به طور جزئی یا کامل ، یک زیر مجموعه قابل شناسایی از کل مورد نهایی یا سیستم مورد نیاز و توابع را برآورده می کند. اغلب چندین ساختار داخلی وجود دارد که منجر به ساخت قابل تحویل برای افزایش چرخه فعالیت می شود. الزاماتی که در یک ساختار افزایشی برآورده می شوند ، در تمام مراحل متوالی نیز برآورده می شوند. آخرین ساخت شامل سیستم نرم افزاری کامل است. نسخه ، ورژن ساختی است که برای آزمایش پذیرش تحویل داده می شود و متعاقباً ممکن است برای استفاده عملیاتی منتشر یا تحویل شود. ساخت های افزایشی را می توان برای هر SI یا گروهی از SI ها برنامه ریزی کرد.

برنامه ساخت نرم افزار. برای نشان دادن برنامه تحویل نرم افزار ، یک جدول ، مانند نمونه جدول ۵.۴ ، باید در SDP یا سند جداگانه ای که توسط SDP به آن ارجاع شده باشد ، وجود داشته باشد. این جدول باید برای هر نرم افزار یک عدد منحصر به فرد داشته باشد که اغلب به آن شناسه منحصر به فرد برنامه (PUI) برای هر نرم افزار، مورد و نام آن ، سازمان در حال توسعه و خطوط کد معادل (ESLOC) برنامه ریزی شده برای هر ساخت می گویند. در بخش ۱۴.۱.۳ توضیحی در مورد چگونگی استخراج ESLOC ارائه شده است. طبق جدول ۵.۴ ، نسخه (اولیه ، ابتدایی ، بروزرسانی ، رفع اشکال) را می توان برای هر تحویل مشخص کرد.

برنامه ساخت مستر نرم افزار (SMBP). برای ترسیم قابلیت های افزایشی ، توانایی ها و نیازهای اختصاص یافته به هر ساخت ، باید یک SMBP جامع ارائه شود. CSWE یا Build Manager معمولاً SMBP را

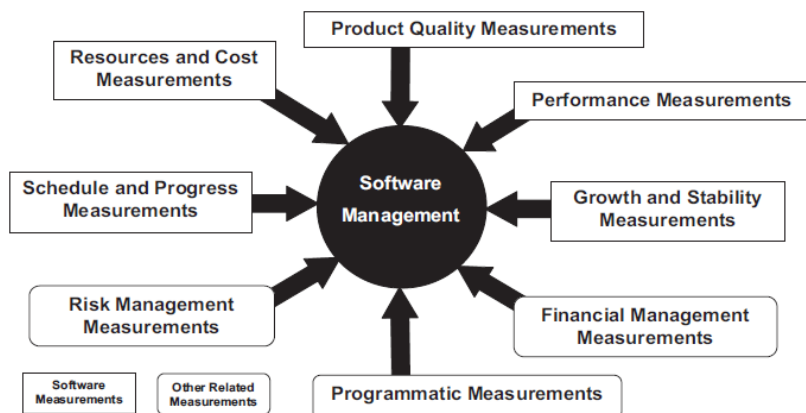
با تأیید نرم افزار CCB حفظ می کند. پس از تأیید ، SMBP باید توسط SCM کنترل شود. در یک برنامه بزرگ ، این یک مسئله پیش پا افتاده نیست.

به عنوان مدیر پروژه ، شما ، CSWE یا مدیر ساخت نرم افزار ، باید به طور مرتب وضعیت و تغییرات را به مدیر برنامه (سطح) ارشد گزارش دهید. SMBP همچنین ممکن است "برنامه اصلی یکپارچه سازی و تأیید نرم افزار" نامیده شود یا ممکن است به عنوان "ماتریس عملکرد ساخت" شناخته شود. SMBP یک سند نرم افزاری است و باید با Master Test Plan (MTP) که یک سند در سطح سیستم است و سخت افزار و نرم افزار را شامل می شود همگام سازی شود.

جدول ۵,۴ برنامه تحویل ساخت SI – مثال

PUI	نام آیتم نرم افزار	توسعه دهنده	Build-1	Build-2	Build-3	کل
	مجموع برای ۱۸ SIS:		102,000	65,000	130,000	297,000
1.0	پشتیبانی تصمیم گیری		45,000	28,000	48,000	121,000
1.1	تحلیل تصمیم	Able Corp	I	UWDR	U	
1.2	الگوریتم های تحلیلی	Able Corp	-	I	U	
1.3	تحلیل سناریو	Able Corp	-	I	U	
1.4	کنترل بستر	Baker Co.	P	I	U	
1.5	کنترل ترافیک	Baker Co.	I	UWDR	U	
1.6	تحلیل شبیه سازی	Baker Co.	P	I	U	
2.0	پشتیبانی خدمات		30,000	18,000	12,000	60,000
2.1	تجزیه و تحلیل مسیریابی	Charlie Co.	P	I	U	
2.2	پشتیبانی کاربر	Charlie Co.	-	I	UWDR	
2.3	خدمات XYZ	Charlie Co.	I	UWDR	UWDR	
	و غیره.	etc.				

P = نسخه مقدماتی ؛ I = تحویل اولیه ؛ U = تحویل به روز شده - = بدون تحویل ؛ UWDR = به روزرسانی ها برای گزارش های عدم اختلاف "###, ###" = تعداد ESLOC در هر ساخت. PUI = شناسه منحصر به فرد برنامه.



شکل ۵,۲ مدیریت نرم افزار از دیدگاه اندازه گیری.

## بروزرسانی طرح برنامه ریزی

برنامه ریزی ساخت نرم افزار باید برای هر برنامه رخ داده و هر ساخت قابل ارائه و بروز شده در کل برنامه اجرا شود. برنامه های ساخت معمولاً فقط هنگامی تغییر می یابند که محتوای برنامه به طور قابل توجهی تغییر کند که توسط SWIPT Lead تعیین شده است. هنگام برنامه ریزی برای ساخت ، برنامه پروژه ، ESLOC و برآورد محتوای عملکردی باید در نظر گرفته شود. با تکمیل برنامه ، طراحی اضافی ، الزامات ، محتوای فنی و رویکردهای آزمایشی باید اضافه شوند. فعالیت های ساخت باید در برنامه های دقیق ثبت شده و سپس به همراه کارکنان و اطلاعات مربوط به برنامه بودجه در IMS گنجانده شود.

## ۵,۱,۵ توسعه نرم افزار

### پیگیری و نظارت

پس از اتمام برنامه ریزی نرم افزار ، تلاش برای ردیابی و نظارت بر نرم افزار آغاز می شود. زیر سیستم SWIPT ها ، کلیدهای نرم افزار ، مهندس ارشد نرم افزار و تضمین کیفیت نرم افزار ، می توانند وضعیت توسعه نرم افزار را از طریق:

جمع آوری و ارزیابی داده های اندازه گیری نرم افزار (نگاه کنید به فصل ۱۵)

انجام ممیزی های کیفیت محصول و فرآیند (به ۶,۱,۲ و ۶,۲ مراجعه کنید)

پشتیبانی از بررسی نرم افزار (نگاه کنید به ۵,۳)

انجام فعالیتهای مدیریت ریسک (نگاه کنید به ۲,۳,۲ ، ۳,۹,۴ و ۵,۲)

نظارت بر اندازه گیری نرم افزار. در طول فرایند توسعه ، داده های اندازه گیری نرم افزار باید برای مقایسه اندازه واقعی نرم افزار ، هزینه ، برنامه و پیشرفت در برابر برنامه تعیین شده استفاده شود تا بتوانید اقدامات اصلاحی به موقع را انجام دهید (بخش ۶,۴ را ببینید). اگر این معیارها شرایط نامتعارفی را نشان میدهند ، شما یا اعضای زیر سیستم SWIPT باید یک تجزیه و تحلیل انجام دهید تا مشکل را شناسایی کنید. در آن مرحله ، شما باید اقدام اصلاحی مناسب را انجام دهید و خطرات احتمالی از جمله تأثیرات هزینه و برنامه را شناسایی کنید. طرح اندازه گیری نرم افزار (یا متریک) یک ضمیمه مهم به SDP است. فصل ۱۵ کاملاً به اندازه گیری نرم افزار اختصاص یافته است.

وضعیت نرم افزار باید در جلسات سطح زیر سیستم و در جلسات ماهانه وضعیت برنامه بررسی شود (معمولاً هفتگی). علاوه بر این ، وضعیت نرم افزار باید ماهانه و همچنین در بررسی های سه ماهه به مشتری ارائه شود. مدیریت و کنترل نرم افزار باید در طرح کلی مدیریت برنامه شما ادغام شود. شکل ۵,۲ تصویری از مدیریت نرم افزار از منظر اندازه گیری است.

همانطور که در شکل ۵,۲ نشان داده شده، نظارت بر اندازه گیری نرم افزار شامل موارد زیر است:

اندازه گیری های خاص نرم افزار:

برنامه ریزی و اندازه گیری پیشرفت

منابع و اندازه گیری هزینه ها

اندازه گیری کیفیت محصول

اندازه گیری عملکرد

اندازه گیری رشد و پایداری

اندازه گیری های غیر نرم افزاری خاص:

اندازه گیری های مدیریت ریسک

اندازه گیری های مدیریت مالی

اندازه گیری های برنامه ای

**نظارت بر هزینه**



بسته های نرم افزاری معمولاً حساب های هزینه ای در سیستم مدیریت ارزش کسب شده (EVMS) است که توسط بسیاری از پیمانکاران استفاده می شود (EVMS در بخش ۸,۵ شرح داده شده است) حساب هزینه باید در سطح جزئیات کافی باشد تا کنترل فعالیتهای مربوط به توسعه نرم افزار را کنترل کند. معیارهای حساب های هزینه و بسته های کاری باید به مدیریت برنامه گزارش شده و در دسترس مشتری باشد.

### نظارت بر برنامه

این نظارت عموماً به صورت هفتگی یا اغلب اوقات اگر مدام در حال تغییر است در بازه های زمانی بیشتری بازبینی شوند. معیارهای برنامه ریزی (با استفاده از دستاوردهای هفتگی مهم ، از جمله داده های پیمانکار فرعی) باید همراه با وضعیت برنامه های اقدام اصلاحی / بازیابی گزارش شود. IMS و برنامه های دقیق نیز باید در سطوح پایین تر در SWIPT بررسی شوند.

### نظارت بر سرشماری

SWIPT ها باید به طور هفتگی میزان سرشماری را کنترل کرده و در شناسایی زود هنگام مشکلات احتمالی اقدام کنند. به روزرسانی های کامل، تعداد واقعی بودجه و پیش بینی شده باید ماهانه انجام شود. پیش بینی ها باید در گزارش های عملکرد هزینه داخلی که شامل واریانس هزینه / برنامه و تغییرات آخرین برآورد اصلاح شده است ، به روز شده و گزارش شوند. هزینه ها و برنامه ها باید با نظارت بر هزینه های کل ، ارزیابی پیشرفت کنترل شوند.

### نظارت بر فرآیند و محصول

ارزیابی محصول ، بررسی نرم افزار ، ممیزی فرآیند و ارزیابی توسط زیر سیستم SQA و CSWE, SWIPTS به عنوان ابزاری برای تعیین انطباق با استانداردهای تعیین شده توسط SDP استفاده می شود. عدم مطابقت محصولات پایه از طریق فرایند اقدام اصلاحی انجام می شود (بخش ۴,۶ را ببینید). برای تعیین انطباق با فرایندهای مشخص شده در SDP ، ممیزی های فرآیند باید توسط SQA و با پشتیبانی CSWE انجام شود. SQA باید مسئول مستندسازی و تأیید بسته شدن مسئله عدم انطباق باشد. در بخش ۶,۵,۶ روند ارزیابی محصول نرم افزار شرح داده شده است. شما یا مدیریت ارشد باید مکانیزم های ارتباط با مشتری و برقراری ارتباط با مشتری را پیاده سازی و حفظ کنید.

۵,۱,۶ از افزون کاری در پروژه خودداری کنید

دستیابی به هدف متعالی و کارآیی در مقایسه با میل به کمال خیلی پر هزینه نیست.

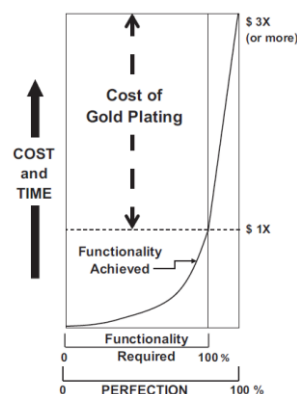
تمایل شدیدی در کلیه مهندسان برای موفقیت والا وجود دارد. که اغلب افزون کاری نامیده می شود. وظایفی را که در قرارداد به عنوان یک نیاز سیستم ذکر نشده، می توان افزون کاری دانست ، مگر اینکه این وظیفه یک نیاز مشتق شده برای عملکرد سیستم باشد. اصلاح کار برای "بهرتر کردن" آن و سپس بازسازی بیشتر برای "حتی بهتر کردن آن" ، و ادامه این روش "پرداخت کاری" همیشه تأثیر شدیدی بر هزینه و زمانبندی پروژه دارد. این قابل درک است که مهندسان معمولاً می خواهند کار خود را بهترین عملکرد ادامه دهند. به عنوان مدیر پروژه نرم افزار ، باید مراقب باشید تا از دام کمال گرایی دوری کنید.

به عنوان مثال ، شکل ۵,۳ یک منحنی تصویری است که هزینه نسبی کمال را در برابر عملکرد سیستم مورد نیاز نشان می دهد. هزینه و زمان رسیدن به کمال بسیار زیاد است. در مواردی که جان و امنیت انسان درگیر باشد ، کمال قابل توجیه است و مشتری باید هزینه آن را پرداخت کند. در غیر این صورت ،

کمالگرایی بسیار گران خواهد بود. وقتی به ۱۰۰٪ از عملکردهای مورد نیاز قرارداد رسیدید ، شرایط قرارداد را برآورده کرده اید. اگر زمان و هزینه در آن مرحله ۱X دلار باشد ، همانطور که در شکل ۵,۳ نشان داده شده ، کمالگرایی می تواند حداقل سه برابر گرانتر و وقت گیرتر باشد.

### درسی که آموختیم:

۵٪ آخر هر هزینه است. ممکن است ذات انسانی سبب شود تا عالی شدن پروژه تحویل آن را به تاخیر بیندازید ، اما مسئولیت مدیر نرم افزار این است که افزون کاری در پروژه رخ ندهد. من مشاهده کرده ام که ، اگر جان و ایمنی انسان ها درگیر نباشد ، ۹۵٪ از عملکرد برنامه ریزی شده اغلب ۹۵٪ از نیاز کاربران را تامین می کند؛ که ۵٪ گذشته یک قسمت بسیار شیب دار از منحنی است ، بنابراین ، اگر بتوانید از انجام این بخش جلوگیری کنید ، می توانید صرفه جویی زیادی در هزینه انجام دهید. این یک بازی حيله گرانه است. اکثر محصولات نرم افزاری "کوچک شده" احتمالاً وقتی محصول در ابتدا به بازار تحویل داده می شود ، تقریباً به ۹۵٪ از عملکرد کامل نزدیک نمی شوند.



شکل ۵,۳ هزینه نسبی افزون کاری در پروژه

## ۵,۱,۷ برنامه ریزی تست موردی نرم افزار

آزمایش موردی نرم افزاری در حال توسعه توسط زیر سیستم ها باید توسط مهندسين مربوطه انجام شود. آنها مسئول مستندسازی تست نرم افزار (STP)، توضیحات تست نرم افزار (STD) و گزارش تست (STR) هستند تا تأیید کنند که SI ها مورد نیاز زیر سیستم آنها را تأمین می کنند. نسخه اولیه STP معمولاً در طول فعالیت های طراحی نرم افزار تولید می شود، اما STP کامل نتیجه فعالیت برنامه ریزی آزمون SI است. تولید STD و STR در خلال تست موردی نرم افزار (SIQT) انجام می شود و در بخش ۱۱,۷ بررسی شده. فعالیت های آزمون برای کلاس های نرم افزاری مهم (SCS) و پشتیبانی (SS) نیز باید در پرونده های توسعه نرم افزار مانند بخش ۹,۴ مستند باشد.

برنامه تست نرم افزار. STP برنامه هایی را برای آزمایش صلاحیت SI توصیف می کند و یک سند مهم نرم افزاری است. این محیط آزمایش مورد استفاده را توصیف می کند، آزمایشات قابل انجام را مشخص می کند، برنامه هایی برای کارهای آزمایش فراهم می کند، منابع مورد نیاز را تعریف می کند و وظایف برنامه ریزی مورد نیاز برای انجام SIQT را بیان می کند. جدول ۵,۵ خلاصه ای از معیارهای آمادگی از نظر معیارهای ورود، خروج و تأیید برای اطمینان از کامل بودن STP است.

## ۵,۱,۸ برنامه ریزی تست سیستم

SDP باید حاوی روند ارائه پشتیبانی از برنامه ریزی تست سیستم است. اگر تأیید سیستمی و طرح تست وجود باشد، این کار باید توسط SEIT صورت گیرد. این کار ماهیت اصلی اسناد برنامه ریزی تست سیستم است. فعالیتهای سیستم یکپارچه سازی و تست سیستم در بخشهای ۱۰,۳ و ۱۰,۴ شرح داده شده اند. تیم آزمایش سیستم باید مسئول انجام آزمایش واقعی سیستم باشد. توسعه دهندگان نرم افزار و / یا

مهندسين آزمایش نرم افزار نقش پشتیبانی در برنامه ریزی آزمایش سیستم دارند که ممکن است شامل مرور آماده سازی آزمون و تهیه موارد پشتیبانی آزمایش نرم افزار ، مانند مستندات آزمایش نرم افزار قابل استفاده مجدد ، شبیه سازها ، درایورها و ابزارهای تجزیه و تحلیل باشد. مهندسان نرم افزار همچنین از تجزیه و تحلیل ناهنجاری برای تعیین اینکه آیا مشکل فقط به نرم افزار ، سخت افزار یا ترکیبی از هر دو مربوط است پشتیبانی می کنند. در صورت نیاز به تست رگرسیون بر روی ساخت نرم افزار ، SCM باید ساخت نرم افزاری را که بر اساس آن آزمایشات انجام می شود ، ارائه دهد. رگرسیون در بخش ۳،۹،۳ مورد بحث قرار گرفته است.

#### **۵،۱،۹ برنامه ریزی برای انتقال نرم افزار به مرحله عملیات و نگهداری**

برنامه ریزی برای انتقال نرم افزار به مرحله عملیات و نگهداری: SDP باید رویکرد انجام برنامه ریزی نصب نرم افزار را توصیف کند. این فعالیت شامل آماده سازی ، و نصب نرم افزار اجرایی در سایت کاربر است. برنامه ریزی و آماده سازی باید تقریباً در اوایل چرخه فعالیت شروع شود تا از انتقال روان به سایت کاربر مطمئن شوید. این باید شامل تهیه مستندات و محصولات نرم افزاری مورد نیاز کاربر برای انجام کارهای عملیاتی باشد. این شامل کد مربوط به هر SI و مستندات پشتیبانی از جمله تهیه کتابچه های راهنمای کاربر و مطالب آموزشی کاربر با در دسترس قرار گرفتن اطلاعات مربوطه است (بخش ۱۱،۹ را ببینید).

## جدول ۵,۵ معیارهای آمادگی: برنامه آزمایش نرم افزار

معیارهای ورود	معیارهای خروج
<p>موارد داده STP مناسب و سایر مواد مرجع بدست می آید.</p> <p>الزامات نرم افزار در این بخش شکل گرفته و الزامات مشخصات و مورد نیاز رابط (IRS) برای نیاز والدین قابل ردیابی است.</p> <p>معماری سطح بالا نرم افزار ایجاد شده است.</p> <p>ماتریس تأیید آزمایش مورد نیاز (RTVM) روش و سطح تأیید آزمون را برای هر نیاز در SRS و IRS مشخص می کند. (به ۱۱,۱,۲ مراجعه کنید).</p>	<p>کارهای زیر در STP تعریف و مستند شده است:</p> <p>- محیط آزمایش (سایت ها ، سخت افزار ، نرم افزار ، ابزار آزمایش ، امکانات آزمون ، داده های آزمایش و ...) برای انجام آزمایشات چرخه فعالیت مورد نیاز است.</p> <p>- سناریوهای آزمایشی که باید شامل برنامه انجام فعالیت های آزمایشی انجام شود.</p> <p>- قابلیت ردیابی بین الزامات SI و آزمایشات مربوطه و مراحل آزمون که در آن مورد تأیید قرار می گیرند.</p> <p>- پرسنل ، سازمان ها ، مسئولیت ها و فعالیت های مدیریتی مورد نیاز برای توسعه و اجرای آزمایش برنامه ریزی شده.</p> <p>اهداف هر آزمون شامل سطح آزمون ، نوع ، شرایط ، داده های ثبت شده ، روش (های) تاییدی ، تجزیه و تحلیل داده ها ، فرضیات و محدودیت ها ، ایمنی ، امنیت و حریم خصوصی.</p> <p>- روند موضوعات مرتبط مانند داده ها ، آموزش ، تست رگرسیون ، تاخیر در عملکرد و اسناد قابل تحویل.</p> <p>- معیارهای ارزیابی نتایج آزمون.</p>
معیارهای تأیید	
<p>آزمونهای مشخص شده کاملاً شرایط سیستم مورد آزمایش را تأیید می کنند.</p> <p>زمان بندی مراحل آزمون در چرخه فعالیت ، به علاوه ضوابط ورود و خروج برای هر مرحله آزمون ، شناسایی و مستند شده است.</p> <p>اصطلاحات و قالب بین SRS ، RTVM ، IRS و STP سازگار است.</p> <p>STP بررسی همتا را با موفقیت پشت سر گذاشت.</p>	
اندازه گیری ها	
آماري از STP Peer Review (نگاه كنيد به فصل ۱۵)	

