**PLAN DE PROCESOS DE SOPORTE**

Ilustración : Plan de procesos de soporte

*Esta sección en particular es bastante extensa, Construx ofrece plantillas completas para cada uno de estos planes ya que en proyecto de gran alcance es necesario verificar y validar cada uno de los artefactos entregados por los miembros del equipo. En esta sección se deben proponer y especificar métricas para tener la posibilidad de calificar los artefactos entregados, dichas métricas deben permitir realizar una clasificación para cada artefacto al darle un peso valorativo y de esta forma aprobarlo o de lo contrario tomar medidas como arreglar dicho artefacto o reiniciar su elaboración.*

*Para el entendimiento de estos planes es necesario tener en claro los siguientes conceptos:*

Ilustración : Definiciones básica del plan de procesos de soporte

## PLAN DE ADMINISTRACIÓN DE LA CONFIGURACIÓN

*La IEEE 610 define el plan de administración de la configuración así:*

*“La administración de la configuración es una disciplina que aplica estrategias técnicas, administrativas y de seguimiento para documentar las características físicas y funcionales de un ítem de configuración, además sirve para controlar, almacenar y reportar todos los cambios realizados sobre estos ítems, por otra parte, este plan también sirve para verificar que todos estos ítems estén cumpliendo los requerimientos del proyecto.”*

*Para empezar a desarrollar este plan es fundamental definir cuáles serán los documentos tipo línea base y de porcentaje de avance además se deben especificar las formas para dar versiones a cada uno de estos documentos. El método que se use para definir las versiones en los documentos puede ser definido por el mismo equipo de trabajo, pero es importante que este formato especifique el avance del ítem que se esté realizando. A continuación se muestra un ejemplo de una posible numeración de versiones para los artefactos de configuración:*

Ilustración : Numeración de versiones

*En cuanto al almacenamiento de las versiones de estos artefactos de configuración Construx [15] aconseja definir claramente donde y quien podrá guardar estos documentos o de lo contrario utilizar un sistema para trabajo concurrente que permita a todo el equipo de trabajo modificar un mismo artefacto de configuración al mismo tiempo. Para este fin existen varias herramientas conocidas bajo el nombre de CVS (Concurrent Versions System por sus siglas en ingles), algunas de estas son Tortoise (disponible en [13] y el CVS que trae por defecto Eclipse en su última versión [14]*

*En este plan se debe definir cómo realizar el control de cambios. Para la estandarización del control de cambios se debe especificar formatos con el fin que los miembros del equipo de trabajo, realicen las peticiones de cambio, estas peticiones tienen como objetivo llevar un historial de cómo ha sido afectado un artefacto en particular [1]. A continuación se presenta una tabla que sirve como modelo para formalizar una petición de cambio sobre un artefacto de configuración:*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Fecha** | **Nombre del solicitante del cambio** | | | **Cambio solicitado** | | | **Razón de petición** | | |
|
| *Fecha en que se solicita el cambio* | *Nombre completo de la persona que solicita permiso para el cambio.* | | | *Descripción del cambio que se desea hacer. En este campo se especifica cuál será el artefacto afectado por el cambio.* | | | *El solicitante debe diligenciar porque desea realizar cambios sobre el artefacto, de esta forma el administrador de configuración puede decidir si aceptar o no los cambios.* | | |
| **Persona responsable de la petición** | | **Cantidad de cambio.** | | | | **Grado de impacto del cambio** | | | |
| **Alto** | **Medio** | | **Bajo** | **Alto** | | **Moderado** | **Insignificante** |
| *Persona encargada de la comunicación entre el solicitante del cambio y el administrador de configuración, en general este campo solo se diligencia si la estructura organizacional es demasiado estricta en sus líneas de comunicación.* | | *Para diligenciar las casillas de “Cantidad de cambio” y “Grado de impacto del cambio” es necesario que el equipo de trabajo liderado por el administrador de configuración definan cuales serán las métricas para calificar los cambios que se desean realizar sobre los documentos. Es importante recalcar que la clasificación realizada en “Alto Medio y Bajo” o “Alto Moderado Insignificante” es solo un ejemplo, el equipo puede definir la escala de sus métricas como guste. Esta sección de la tabla es diligenciada por la persona que desea realizar el cambio.*  *La sección de cantidad de cambio se diligencia de acuerdo a que tan grande o pequeña será la parte modificada del artefacto, y por otra parte, la casilla de Grado de impacto del cambio se diligencia de acuerdo a que tantos artefactos afectara la modificación que se desea realizar.* | | | | | | | |

Tabla : Peticiones de cambio

*Con el fin de realizar el seguimiento adecuado a cada uno de los artefactos de configuración, y que todos los miembros del equipo conozcan claramente cuál es la última versión y que contiene dicha versión, es necesario que en caso de aceptar una petición de cambio el administrador de configuración comunique a todos los miembros del equipo cual fue el artefacto afectado y una descripción de dicho cambio. A continuación se muestra un ejemplo de una posible tabla para realizar el seguimiento de cambios a los artefactos de configuración.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Documento de Cambio* | *Nuevo nombre del documento* | *Responsables* | *Fecha de cambio* | *Razones por las que cambia el documento* | *Archivos que implican modificación* |
| *Nombre del documento modificado* | *Nuevo nombre del documento que por lo general es un incremento en la versión* | *Personas que realizaron el cambio en el documento.* | *Fecha en la que se realizo el cambio* | *Razones por las cuales el administrador de configuración acepto el cambio en el documento.* | *Este campo se diligencia con todos los archivos que están sujetos a ser modificados cuando se realice el cambio.* |

Tabla : Seguimiento de cambios

## PLAN DE VERIFICACIÓN Y VALIDACIÓN

*Antes de empezar a desarrollar este plan es necesario tener en cuenta la diferencia entre estos dos términos, a continuación se muestra la definición de cada uno de estos según el diccionario enciclopédico VOX y RAE:*

* *Verificar: Comprobar o examinar la verdad de algo [RAE].*
* *Validar: Comprobación de un conjunto de datos para determinar si su valor se halla dentro de unos límites de fiabilidad [VOX].*

*Un ejemplo orientado a ingeniería de software my sencillo para entender la diferencia entre estos dos términos es:*

* *Verificar es revisar que un requerimiento haya sido implementado.*
* *Validar es revisar que dicho requerimiento cumpla con su función dado un conjunto de pruebas en donde se encuentren los datos que permite el requerimiento.*

*Según la NASA, las actividades del plan de verificación y validación deben evaluar que los elementos de software cumplan con los requerimientos establecidos con el cliente final. Nótese que la NASA hace referencia a los “elementos de software” como a todos los artefactos que se tienen que entregar para un proyecto de este tipo como por ejemplo los manuales, la documentación, las versiones Beta etc. En resumen, la guía del plan de verificación y validación de la NASA propone los siguientes cuatro consejos para estos tipos de planes:*

* *Verificar que cada elemento de software ayude a cumplir los requerimientos.*
* *Evaluar cada elemento de software antes que este sea usado como base (entrada) para otras actividades.*
* *Asegurar que las evaluaciones sobre los elementos de software sean realizadas por una persona diferente al autor del mismo.*
* *Asegurar que el esfuerzo y dedicación en la verificación y validación de cada elemento de software sea suficiente para poder aprobar dicho elemento como base para las siguientes actividades.*

*La NASA también aconseja realizar reportes sobre las pruebas realizadas al código, estas pruebas son las mismas mencionadas en el plan de aseguramiento de la calidad así que se puede realizar una referencia a la sección 7.4. En resumen, el plan de verificación y validación se ejecuta en paralelo en todo el ciclo de vida del proyecto para realizar pruebas sobre todos los documentos y entregables que se realicen, por otro lado es importante aclarar que el equipo de trabajo puede definir sus formas de verificación y validación con los métodos que gusten, es más, pueden solo hacerlo sobre el código por falta de tiempo en el cronograma.*

## PLAN DE DOCUMENTACIÓN

*Según la IEEE esta subclase del SPMP debe contener un plan de cómo se documentaran todos los entregables del proyecto de software, este plan incluye productos que se entregan a los clientes y a los mismos desarrolladores. Además se debe definir que sección de la organización se encargara de proveer información para la realización de los documentos y por supuesto dicha sección de la organización también se encargara de realizar las revisiones sobre cada documento. Por otra parte, los encargados de este plan deben generar lista de chequeos para verificar los ítems para cada documento, por ejemplo, una lista de chequeo para los documentos puede ser los mismos numerales utilizados para dicha plantilla. En resumen este plan debe contener: [3]*

Ilustración : Contenido plan de documentación

## PLAN DE ASEGURAMIENTO DE LA CALIDAD

*En general, el objetivo de este plan es establecer lineamientos para garantizar un producto final libre de defectos, incluyendo software y documentación. La lista de chequeo que se muestra a continuación es propuesta por Construx [15] y sirve para asegurar que los estándares de calidad establecidos para el proyecto se cumplan en cada fase del ciclo de vida del mismo:*

* + *Lista de chequeo para el plan de aseguramiento de la calidad [16].*

Ilustración : Lista de chequeo para el aseguramiento de la calidad

*Por lo general en un proyecto de software existen productos y entregables generales, a continuación se muestran algunos de estos dando un ejemplo de cómo una organización puede asegurar la calidad en este entregable o artefacto de configuración.*

* ***Código:*** *Se debe establecer como los desarrolladores tomaran decisiones de acuerdo a lo que se va a implementar, es decir, que algoritmo es el indicado para desarrollar una parte del programa. Una posible forma de especificar como tomar dichas decisiones e implementar el código, es realizar reuniones con el arquitecto de software para informarle como se está realizando cierta parte del programa, y así el arquitecto aprobara o no dicha decisión si esta cumple con los requerimientos y arquitectura establecida.*
* ***Documentación de código:*** *Para la documentación de código el equipo de trabajo puede realizar su propio formato de código, especificando como se documentará cada sección del mismo, pero por lo general los proyectos de software toman formatos previamente definidos como el que utiliza Java en su API; por lo general, las herramientas de desarrollo permiten documentación con el formato Javadoc [17] que es el mismo mencionado anteriormente.*
* ***Manuales:*** *Por lo general los productos de software contienen dos diferentes tipos de manuales, el manual de usuario, y el de instalación. Los manuales de usuario deben contener un escenario de ejemplo para cada funcionalidad que el software ofrezca, así que el equipo de trabajo debe especificar la forma en que se harán estos escenarios, por ejemplo, si se desean colocar imágenes de muestra, se debe definir previamente la dimensión de la imagen, así como el tamaño de las hojas y de los párrafos. Por otra parte para la realización de los manuales de instalación se deben especificar previamente para que plataformas operativas se realizara el software y así realizar pruebas para cada plataforma y documentar el proceso de instalación, en algunas ocasiones es necesario instalar aplicaciones extras para que el software funcione correctamente como por ejemplo la maquina virtual de java, librerías de C o frameworks para diferentes lenguajes, para este aplicaciones se debe detallar en el manual de instalación cual es la versión que se usara y donde conseguir la documentación asociada a este.*
* ***Documentación para el producto de software (SPMP, SRS, SDD):*** *Se debe especificar en qué forma se realizará cada uno de los diferentes documentos a entregar, por lo general esto se define haciendo referencia a alguna plantilla existente como las de la IEEE, Construx, ISTAR etc.*
* ***Plan de pruebas:*** *En este plan se deben especificar las métricas para calificar los resultados de las pruebas y tener formalizado que hacer en caso de que una prueba arroje un resultado no deseado. Por lo general siempre se aplican estos cuatro tipos de prueba sobre un producto de software [18]:* 
  + ***Pruebas de programa con datos de prueba:*** *En estas pruebas se validan y verifican secciones separadas del código, en otras palabras, cada clase por separado (si es programación orientada a objetos). Los datos con los que se desea realizar las pruebas deben ser ingresados con el fin de revisar que sucede cuando estos se encuentran en el intervalo aceptado por el método o función probada y que pasa cuando estos datos exceden el intervalo. El propósito principal es revisar en que secciones del código se debe realizar manejo de excepciones.*
  + ***Pruebas de vínculos con datos de prueba:*** *En este tipo de pruebas se empiezan a fusionar clases o módulos, su objetivo al igual que las pruebas anteriores es revisar si con la vinculación y datos dentro y fuera del intervalo la nueva fusión de clases funciona correctamente.*
  + ***Sistema completo con datos de prueba:*** *Esta prueba se realiza cuando el sistema está completo, al igual que las pruebas anteriores, los datos ingresados deben estar dentro y fuera del intervalo para revisar su comportamiento en estas situaciones. Por lo general, estas pruebas ayudan a modificar la interfaz gráfica para indicarle al usuario que ha ingresado mal un dato o que simplemente no lo ingreso.*
  + ***Pruebas de aseguramiento de requerimientos funcionales:*** *Esto es más una lista de chequeo donde se verifica que el software este cumpliendo con todos los requerimientos funcionales del software.*
  + ***Pruebas de aseguramiento de requerimientos no funcionales:*** *Las pruebas para los requerimientos no funcionales deben depender de los documentos de diseño del sistema, ya que en estas pruebas, se debe especificar por ejemplo tiempos de respuesta del sistema o el número de horas que el sistema estará activo, esto depende del tipo de requerimiento no funcional que se desee probar.*

Ilustración : Proceso de pruebas

## REVISIONES Y AUDITORIAS

*Este plan está muy ligado con el nivel de calidad que se le quiere dar al proyecto, ya que dependiendo de la exigencia de los niveles de calidad se deben contratar auditorias para diferentes secciones del producto como por ejemplo, una auditoria para documentos y otra distinta para código. Según la IEEE este plan especifica los siguientes elementos [3]:*

* *Empresas, entidades o personas encargadas de hacer las auditorias.*
* *Fechas en que se realizaran las auditorias.*
* *Auditorias para el aseguramiento de la calidad*
* *Número de personas que se encargaran de la revisión de cada entregable.*
* *Fechas en las que se revisará el avance del proyecto.*

## PLAN DE RESOLUCIÓN DE PROBLEMAS

*Especifica recursos (dinero, tiempo, personal), métodos, herramientas (software), técnicas, y procedimientos para ser usados en el reporte, análisis y priorización de los problemas que surjan a lo largo del proyecto. El plan de resolución de problemas debe definir que miembro del equipo será designado a los diferentes problemas que pueden surgir en un proyecto de software, a continuación se muestra un ejemplo de esta asignación [3]:*

* ***Director de desarrollo:*** *Será el encargado de gestionar los retrasos en la implementación mediante subcontrataciones y/o pago de horas extras.*
* ***Arquitecto:*** *Serán los encargados de verificar que la implementación realizada siga la arquitectura establecida, esto se realiza mediante pruebas que comprueben la no dependencia entre los módulos, y verificar que la comunicación entre los componentes del sistema se realice de acorto al protocolo y puertos especificados. Si las pruebas den resultados no deseados el arquitecto exigirá al equipo de trabajo las modificaciones correspondientes.*
* ***Administrador de configuración:*** *En caso de retrasos en los lanzamientos de las líneas bases, su labor, es realizar un consolidado de los artículos de avance para llegar a una posible aproximación y así no atrasar por completo al equipo de trabajo.*
* ***Gerente:*** *En el plan de revisiones se definen las fechas en que se hará el seguimiento al cronograma, por lo tanto, si se descubre un retraso el gerente deberá hacer uso de sus recursos (dinero, personal, tiempo) para evitar que dicho retraso represente una amenaza para el proyecto. Los recursos se manejan de diferente forma de acuerdo a las etapas del proyecto. A continuación se muestra un ejemplo de un plan de resolución de problemas en un retraso en la codificación, donde se clasifican las soluciones de los problemas de acuerdo a las etapas del proyecto:*
* *Etapa 1, retrasos en el inicio de la implementación: Si se evidencian retrasos en esta etapa del proyecto se integraran nuevos miembros al equipo de desarrollo ya que estos no retrasaran las labores debido a que estarían al mismo nivel de conocimiento que los demás.*
* *Etapa 2 y 3, retrasos en implementación y pruebas: En esta etapa no se contrataran nuevos desarrolladores, debido a que estos retrasaran al equipo debido a que se les tendrá que capacitar y explicar detalladamente el código implementado, así que la solución será solicitar al equipo horas extra de trabajo para recuperar el tiempo perdido.*
* ***Director de calidad:*** *En caso de encontrar defectos en los entregables del proyecto o en los procesos definidos del mismo, es obligación del director de calidad delegar tareas para satisfacer los requisitos de calidad no logrados por el equipo de trabajo. Si los defectos encontrados en los entregables del proyecto son causados por el no cumplimiento de los deberes de los miembros del equipo de trabajo, el director de calidad debe diseñar estrategias para que estos errores no se vuelvan a repetir, dichas estrategias pueden ser memorandos o sanciones laborales.*

## PLAN DE ADMINISTRACIÓN DE SUBCONTRATOS

*Esta subclase del SPMP contiene planes para administrar cualquier clase de subcontrato que contribuya al desarrollo de los productos del proyecto de software. Cuando los subcontratos no son pactados desde el principio se deben definir listas de chequeo para aprobar dichas subcontrataciones, por el contrario, si ya se pacto un subcontrato, se debe especificar un cronograma con la entidad encargada de la ejecución de este, dicho cronograma será objeto de un plan administración y seguimiento especificado en esta sección.*

*En ocasiones, las organizaciones cuentan con diferentes formatos para la celebración de los contratos, por ejemplo, diferentes términos para auditorias y para realización de programas, dichos contratos deben estar como anexos del presente documento [3].*

## PLAN DE MEJORAS DEL PROCESO

*Esta sección del SPMP debe determinar las áreas de la organización que pueden ser objeto de mejora, y por supuesto, es esta sección también se especifica cómo se realizarán estos planes de mejora. Este plan debe estar fuertemente ligado con el plan de resolución de problemas, por ejemplo, la fuente causante de un problema puede encaminar al equipo de trabajo a mejorar el proceso que genera dicho problema, la ejecución de este plan trae grandes beneficios como la reducción en el esfuerzo dedicado a la revisión de los documentos y la reutilización tanto de documentación cómo de código. Por lo general cuando el proyecto ya están en marcha, los procesos no pueden sufrir grandes cambios debido a que generarían retrasos en el cronograma, entonces, es deber del director de calidad, documentar los cambios que se le puede hacer al proceso que presenta problemas para que en la realización de un nuevo proyecto se realicen las respectivas mejoras sobre el proceso [3].*

Ilustración : Plan de mejoras de proceso