“REESTRUCTURACIÓN Y POTENCIACIÓN DEL PROCESO DE DESARROLLO DE SOFTWARE EN COLORQUÍMICA S.A.”

SANDRA CAROLINA VELEZ HOLGUIN

UNIVERSIDAD EAFIT

ESCUELA DE INGENIERÍA

DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS

Medellín

2010

“REESTRUCTURACIÓN Y POTENCIACIÓN DEL PROCESO DE DESARROLLO DE SOFTWARE EN COLORQUÍMICA S.A.”

Trabajo de grado presentado como requisito para optar al título de Ingeniero de Sistemas.

Asesor:

Rafael David Rincón Bermúdez

UNIVERSIDAD EAFIT

ESCUELA DE INGENIERÍA

DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS

Medellín

2010

NOTA DE ACEPTACIÓN

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

PRESIDENTE DEL JURADO

_____

JURADO

_____

JURADO

_____

Medellín, noviembre de 2010

AGRADECIMIENTOS

Primero que todo quiero agradecer a Dios por darme la vida, la salud y la fuerza que necesito para lograr mis metas. Por las oportunidades que me ha brindado, porque no me ha dejado sola aún en los momentos más difíciles, porque me ha dado la fortuna de tener a mi alrededor personas maravillosas.

Segundo, quiero agradecer a mi asesor Rafael David Rincón Bermúdez, quien estuvo apoyándome constantemente en la realización de mi trabajo de grado y en gran parte de la carrera, quiero agradecerle por el tiempo y la dedicación que me ofreció para culminar exitosamente este trabajo.

En tercer lugar quiero agradecer a la empresa Colorquímica S.A, que depositó su confianza en mí para acceder a información confidencial para la realización de mi trabajo de grado. Gracias por permitirme brindar una solución a un problema interno, por escuchar mis propuestas, gracias por la posibilidad que me dieron de tener mi primera experiencia profesional, la cual fue enriquecedora.

De igual manera, quiero agradecer a mis profesores, quienes me brindaron gran parte del conocimiento que ahora poseo, a mis compañeros de estudio con quienes viví momentos tanto estresantes como relajados, con los cuales compartí, no solo intelectualmente sino también experiencias que marcaron mi vida.

Y por último, y más importante, quiero agradecer a mi familia, en especial a mis padres, por brindarme todo su apoyo durante estos años de universidad, porque se alegraron y vivieron conmigo cada triunfo, y, en vez de criticarme, me dieron aliento y apoyo en los fracasos y tal vez desilusiones que se presentaron. Gracias por estar conmigo en todos los momentos de mi vida. A ustedes dedico este triunfo, gracias a Dios y a ustedes por darme la oportunidad de tener una formación profesional que me apasiona.

CONTENIDO

LISTADO DE FIGURAS Y ANEXOS

INTRODUCCIÓN

Un proceso de desarrollo de software orientado bajo estándares de calidad, con definiciones técnicas y funcionales correctas puede contribuir a la generación de mejores resultados en las aplicaciones que se desarrollan, en los tiempos que se invierten, en costos, en la productividad, tanto del personal como de la empresa en general a través del control eficaz de todas las etapas que implica el desarrollo de software, además de centrarse en las necesidades del cliente y mejorar la gestión de la infraestructura, del personal, de los recursos, etc., también contribuye a ofrecer soluciones que permitan dar al usuario una respuesta inmediata a sus necesidades y mejorar continuamente la efectividad (eficiencia + eficacia) y productividad de la empresa.

Para el control de un proyecto en ejecución se debe que tener un proceso de software definido que permita predecir tiempos, riesgos, costos, y sobre todo, asegurar la calidad del producto. No contar con un proceso de desarrollo definido, no implica que los productos sean de mala calidad, sin embargo casi siempre, cuando el área de TI (Tecnologías de Información) no cuenta con un adecuado control sobre sus proyectos, se presentan retrasos en su plan de trabajo, las probabilidades de error se tienden a aumentar, si no hay orden se consume más tiempo de lo planeado, se incurre en gastos inesperados aumentando los costes, hay más gasto operativo, más esfuerzo y desgaste físico por parte del equipo de desarrollo, el cliente termina insatisfecho por retrasos en las fechas de entrega del producto, no hay suficiente documentación técnica ni funcional de la elaboración de cada producto, se sobrecargan los recursos con responsabilidades y tareas, etc. “Con un proceso predecible, el staff de TI, sabe razonablemente bien qué es lo que necesitan hacer y cuáles serán los entregables, sea que estén manejando una simple corrección de programación, un conjunto de cambios, o la creación de una aplicación completamente nueva. Algo que es importante recalcar es que los usuarios también ven resultados consistentes por parte de los proyectos de TI cuando estos se adhieren a estas reglas de juego”[1]

[1] Tomado de: CONTE, Paul. Guía de supervivencia para el desarrollo de software. Pág. 5 SoftLanding Systems, Inc., Peterborough. SISRED S.A.C. , Lima – Perú 2003

“Es agotable y totalmente ineficiente tratar de hacer un mejor trabajo de desarrollo de software a largo plazo, sin contar con un proceso bien definido. Sin tal proceso, las buenas ideas no podrán ser integradas efectivamente dentro de las prácticas en ejecución al interior de la organización de desarrolladores. Es más, el tratar de “apagar incendios” (resolver múltiples problemas) desperdicia demasiado tiempo y atención que deberían estar enfocados a la mejora en el desarrollo mismo del software”.

En este proyecto de grado se aportan conceptos que pueden ser soluciones para una reestructuración del proceso de desarrollo de software acorde con las necesidades del área de TI de Colorquímica, partiendo de los estándares establecidos por la empresa para el desarrollo de software, las definiciones según la norma ISO 9001, procesos enfocados en establecer orden y control para optimizar y mejorar la calidad de las aplicaciones que se desarrollan, tomando como base las prácticas establecidas por las metodologías PSP (Personal Software Process), el Proceso de Software Personal y TSP (Team Software Process), el Proceso Software para Equipos de Desarrollo.

“PSP es un proceso de auto mejora, diseñado para ayudar a controlar, administrar y mejorar la forma en que se trabaja. Es un framework estructurado de formas, directrices y procedimientos para desarrollar software”[2.], el Proceso de Software para Equipos de desarrollo tiene como fin formar y guiar equipos de ingenieros que desarrollan software. Por su parte, TSP “es un conjunto de procesos definidos y estructurados que enfatizan el balance entre procesos, producto y trabajo en equipo. Indica qué hacer en cada fase del ciclo de desarrollo del proyecto y muestra cómo aplicar prácticas reconocidas de Ingeniería de Software en un ambiente de trabajo en equipo”.[3]

Con esta propuesta se podrán minimizar los factores que bajan la productividad de la empresa y la de los desarrolladores, minimizar el nivel de estrés del área de TI, los reclamos por parte del usuario, maximizar la eficiencia de todas las partes involucradas tanto en el desarrollo de software como en los productos, resultado del mismo.

---

[2] RINCON BERMUDEZ, Rafael. 2008. “Calidad del Proceso de Software” Presentación en pdf. Pág. 11

[3] RINCON BERMUDEZ, Rafael. 2008. “Calidad del Proceso de Software” Presentación en pdf. Pág. 45.

1. ANTECEDENTES

En la actualidad y desde 1987, el área de TI de Colorquímica S.A desarrolla sus propias aplicaciones software según las necesidades y requerimientos cotidianos que surgen en la empresa, por lo que la compañía para la que trabaja es su "cliente". TI como proveedor de software definió en el año 2007 un proceso muy general de desarrollo de software que, aunque está definido, no se lleva del todo a cabo.

A continuación se presenta el diagrama de proceso de software tal y como se encuentra en Colorquímica.

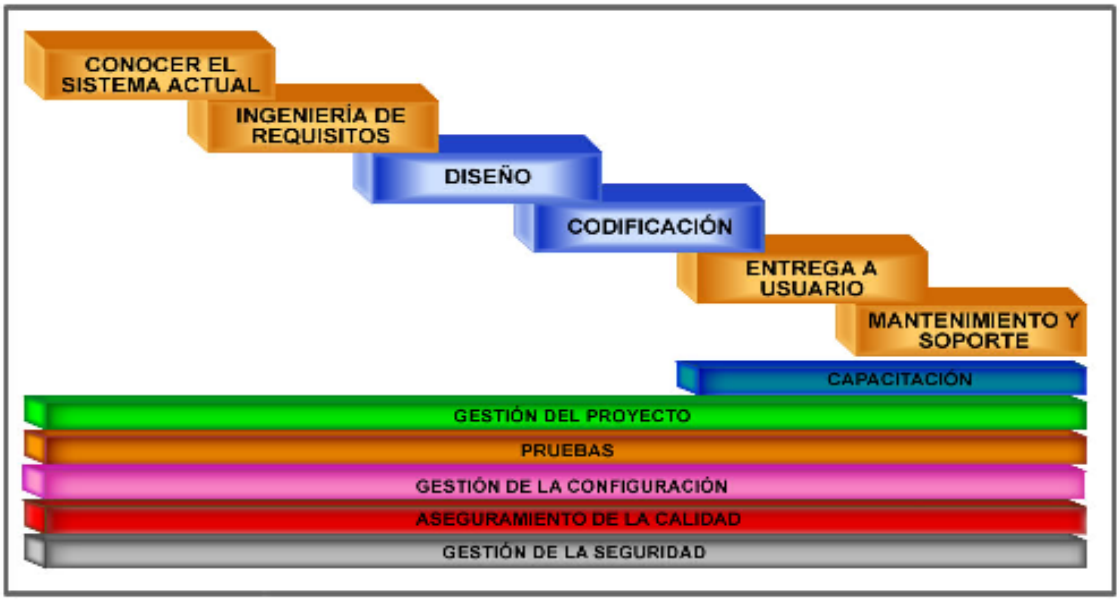


Figura 1. Proceso desarrollo SW CQ
Fuente: TI Colorquímica S.A

La documentación de este proceso es muy general e incompleta, muchas herramientas o procedimientos ya no se utilizan. No todos los subprocesos están definidos y los que están documentados tienen especificaciones que están desactualizadas.

De forma también muy general, partiendo de la norma ISO 9001, el área de TI definió un P-H-V-A para este proceso de desarrollo de software:

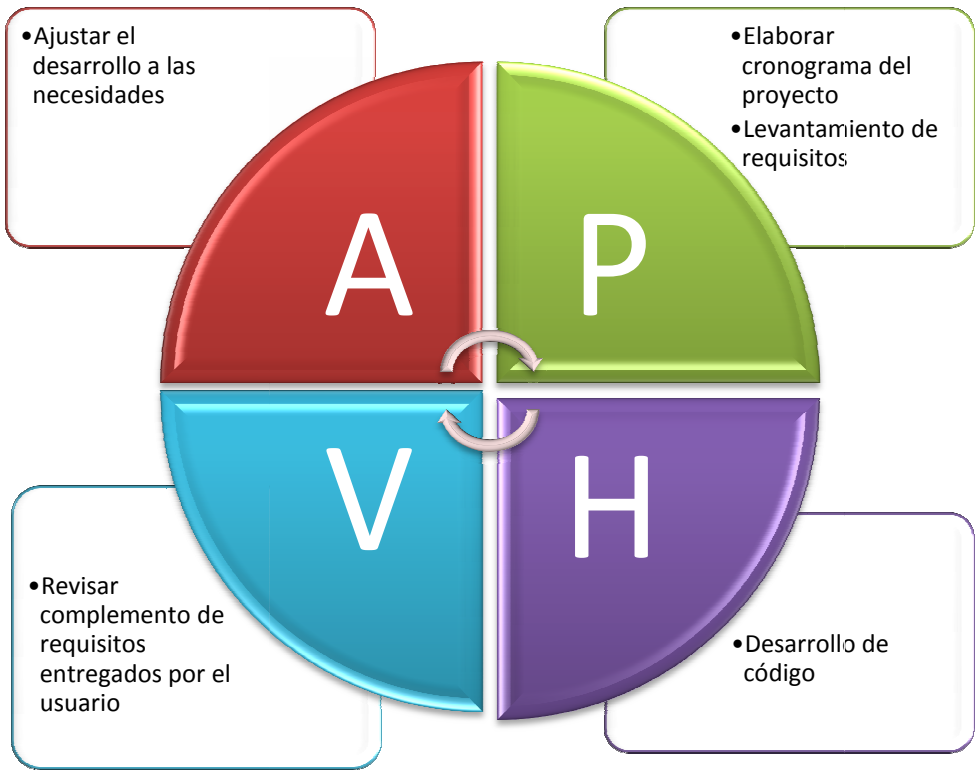


Figura 2. PHVA desarrollo de SW CQ
Fuente: Contenido, TI Colorquímica S.A. Diagrama, elaboración propia.

Definido de la siguiente manera:

ENTRADAS								SALIDAS				
INTERACCIONES			COMUNICACIONES					COMUNICACIONES		INTERACCIONES		
PROCESO	RESPONSABLE	INSUMOS	INFORMACIÓN	MEDIO		ACTIVIDAD	RESPONSABLE	INFORMACIÓN	MEDIO	PRODUCTO	RECEPTOR	PROCESO
					DESARROLLO DE SOFTWARE							
Proceso del Negocio	Usuario	-	Necesidades de software.	Escrito	P	Elaborar cronograma del proyecto	Líder del proyecto		Electrónico	Cronograma del proyecto	Usuario y Gerente de área	Proceso del Negocio
					P	Levantamiento de requisitos	Líder del proyecto y desarrolladores	Acta de requisitos	Electrónico		Usuario y Gerente de área	Proceso del Negocio
					H	Desarrollo de código	Líder del proyecto y desarrolladores					
					V	Revisar cumplimiento de requisitos entregados por el usuario	Líder del proyecto	Capacitación	Electrónico	Software de aplicación y manual de la aplicación	Usuario y Gerente de área	Proceso del Negocio
					A	Ajustar el desarrollo a las necesidades	Líder del proyecto y desarrolladores	Capacitación	Electrónico	Software de aplicación mejorado y manual de la aplicación	Usuario y Gerente de área	Proceso del Negocio

Figura 3. Definición PHVA desarrollo SW

Fuente: TI Colorquímica S.A

2. Definición del problema

Actualmente, el área de TI de Colorquímica S.A desarrolla su propio software teniendo como base la definición de un proceso de desarrollo cuya última actualización fue realizada en el año 2005, y según el trabajo diario, dicho proceso no se lleva a cabo a plenitud hoy en día. Es un proceso de desarrollo que no está completamente definido y documentado, que recarga tareas sobre un solo responsable, trayendo como consecuencia retardos en el tiempo de entrega, errores frecuentes en las aplicaciones, reprocesamiento de tareas, entre otras características que afectan el desempeño y la calidad, tanto de los productos como del personal de TI, y como consecuencia, a la compañía, ya que afecta el tiempo de respuesta a los clientes, baja la productividad de los empleados aumentando el tiempo de ocio, dado el largo tiempo de respuesta del área de TI a las necesidades y problemas que presentan las aplicaciones, entre otros. Estos problemas impactan el proceso de negocio, ya que Colorquímica integra sus operaciones apoyándose en las tecnologías de información.

Surge entonces la necesidad de redefinir el proceso de desarrollo de software implementando actividades y procedimientos que permitan mejorar la calidad de los productos, servicios y la productividad del personal involucrado en el desarrollo de software, partiendo de la norma que rige la calidad de la empresa, "ISO 9001, la cual especifica los requisitos para un buen sistema de gestión de la calidad que puede utilizarse para su aplicación interna por las organizaciones, para certificación o con fines contractuales"[4]. Y potenciando dicho proceso con las prácticas de las metodologías PSP (Personal Software Process) y TSP (Team Software Process).

[4] Tomado de: http://www.docstoc.com/docs/5455974/CERTIFICADOS-DE-CALIDAD

3. Justificación

La empresa Colorquímica S.A. obtuvo la certificación por ICONTEC en ISO 9001/2000 en el año 2002, por lo que todo proceso que allí se desarrolle debe estar debidamente definido, documentado, llevado a cabo tal y como se definió. De nada sirve tener procesos si no se siguen.

Sin embargo, el área de TI tiene un proceso definido desde hace tres años, el cual se sigue más por intuición que por convicción. El proceso actual no está completamente documentado ni actualizado. Por otra parte, no se hace una completa gestión de proyectos donde se siga y evalúe el proceso de desarrollo de una aplicación, entre otras características que no se tienen en cuenta a la hora de ejecutar un proyecto.

A partir de las reuniones de evaluación de área se encontraron falencias referentes a documentación y actualización, tanto del proceso de desarrollo de software como de los productos que se han realizado al día de hoy. Es aquí donde surge la idea y la necesidad de dar el primer paso para el mejoramiento de la calidad de los productos y servicios que presta el área de TI en Colorquímica, empezando con la reestructuración del proceso de desarrollo, documentación y actualización del mismo según las necesidades, herramientas, recursos que se tienen actualmente y las normas que rigen el área.

Además, el personal debe ser consciente que no es solo el proceso lo que debe estar certificado y llevado a cabo tal y como se define, los desarrolladores también deben tener criterio de cómo hacer las cosas de la mejor manera, con las mejores prácticas para llevar a cabo dicho proceso. Para esto se presenta como alternativa la adopción de las prácticas PSP y TSP, con el fin de potenciar el proceso y la forma de trabajo de los desarrolladores, ya que con la adopción de estas prácticas se puede garantizar un producto con calidad, una estimación acertada en tiempo y costos, un mejor aprovechamiento del tiempo, entre otros beneficios que se describirán en el proyecto.

4. OBJETIVOS

4.1 GENERAL

Redefinir el proceso de desarrollo de software de Colorquímica S.A. partiendo de la norma ISO 9001/2000 /2008 y de los requerimientos del área de TI, con el fin de satisfacer las necesidades de usuario final y sus clientes, potenciar el proceso de desarrollo de software y la productividad de los desarrolladores, adoptando las prácticas propuestas por PSP y TSP.

4.2 ESPECÍFICOS

- Redefinir el proceso de Ingeniería de Software de forma que sea coherente según las normas ISO/IEC 15504, ISO 9001:2000, entre otras que se definirán en el proyecto, y las que rigen tanto al Departamento de Sistemas como a la empresa actualmente, que cumplan con los requerimientos mínimos como lo son, control de la calidad, gestión de proyectos, documentación, etc. para llevar a cabo un producto con calidad.
- Documentar el proceso con información actual, estándares de programación vigentes, herramientas y recursos disponibles, entre otras características que hagan el proceso adecuado para implementarlo en TI Colorquímica S.A.
- Potenciar el proceso de desarrollo de software propuesto, con la adopción de las prácticas propuestas por PSP y TSP.
- Elaborar un documento base para con los pasos e información referente a la adopción de PSP y TSP.
- Realizar entrevistas con expertos en el tema de PSP y TSP.
- Validar el proceso propuesto con el área de TI de Colorquímica.
- Realizar los ajustes pertinentes a la propuesta.

5. ALCANCE

La realización y aplicación de este proyecto tiene como limitante físico el área de TI de la empresa Colorquímica S.A ubicada en el municipio La Estrella de la ciudad de Medellín. Sin embargo, la propuesta aquí presentada podrá ser de utilidad para áreas de TI de diferentes empresas e incluso, empresas dedicadas al desarrollo de software.

6. MARCO TEÓRICO

6.1 PROCESO:

Según Gabriel A. Pall, 1987: *Un Proceso puede ser definido como la organización lógica de personas, materiales, energía, equipos y procedimientos dentro de unas actividades de trabajo diseñadas para producir un resultado final específico*.

Una de las definiciones más generales de Proceso es: *Conjunto de actividades interrelacionadas o que interactúan, que transforman entradas en salidas* [Ing. Santiago Gámez, ICONTEC]

Un proceso consta de una Entrada, una Transformación y una Salida, donde la transformación depende de una serie de actividades que generan un producto "Salida" que es insumo para un proceso posterior. Para validar que el proceso siguiente trabajará con un subproducto confiable, se debe hacer una retroalimentación después de cada fase.

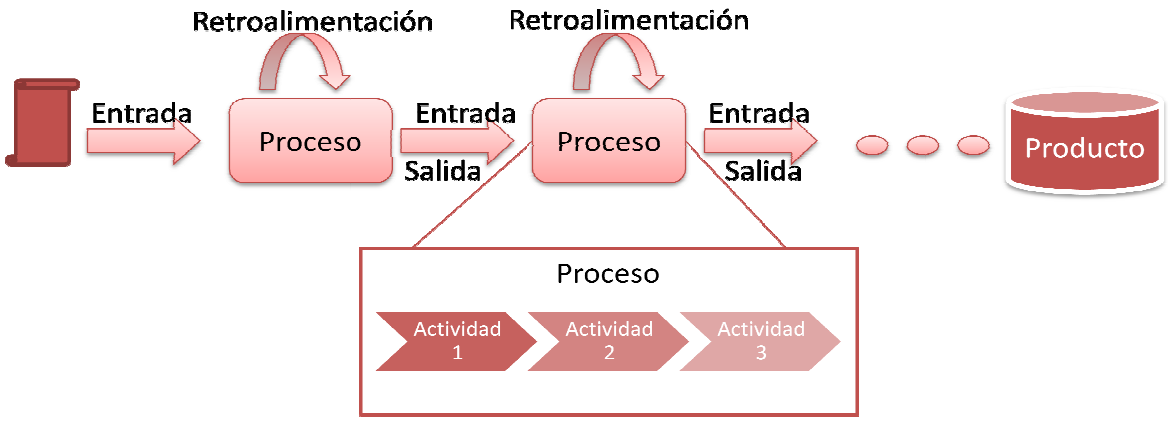


Figura 3. Proceso
Fuente: Elaboración propia

Donde las entradas pueden ser tangibles o intangibles:

- Materiales
- Equipos
- Componentes
- Energía

- Información (requisitos del producto, características y estado del producto, etc)
- Recursos financieros
- Otros

Las salidas pueden ser tangibles o intangibles:

- Materiales
- Productos
- Subproductos
- Componentes
- Energía
- Información (características y estado del producto, monitoreo del producto, datos de medición del mismo, etc.)
- Otros

El proceso como tal, visto como una caja negra, son un conjunto de una o más actividades, procedimientos o tareas que se realizan para transformar la entrada en la salida; y la retroalimentación es un paso que sirve para dar mejora al producto resultado, estudiándolo y midiéndolo para optimizar de alguna manera las tareas o actividades que se llevan a cabo para generar salidas de mayor calidad.

Trabajar con un enfoque por procesos genera valor agregado, ya que puede garantizar la buena calidad de los resultados que se obtienen a partir de estos.

Para que una organización funcione de manera eficaz, tiene que identificar y gestionar numerosas actividades relacionadas entre sí. Una actividad que utiliza recursos, y que se gestiona con el fin de permitir que los elementos de entrada se transformen en resultados, se puede considerar como un proceso.

Frecuentemente el resultado de un proceso constituye directamente el elemento de entrada del siguiente.

6.1.1 PROCESOS DE DIRECCIÓN

En una empresa, “los procesos de dirección corresponden a la definición de la política y estrategia de una organización y al control de las acciones realizadas para lograr los objetivos de la organización.”[5] En desarrollo de software, los procesos de dirección son los que definen cómo se realizará el proceso, controlan y coordinan el mismo. En estos procesos se estudia el tiempo, el alcance, el presupuesto, la calidad, entre otros factores que afectan la realización del producto final; son los que direccionan hacia el buen funcionamiento del proceso, asegurando que se siga como debe ser, garantizando total calidad, no solo del producto sino también del proceso.

6.1.2 PROCESOS DE EJECUCIÓN

“Corresponde a la realización del producto o servicio y por lo tanto a la actividad empresarial de la compañía”[6] En el desarrollo de software son todas las fases, procedimientos y actividades que se siguen para elaborar una aplicación de software.

6.1.3 PROCESOS DE APOYO

En una organización, “representa una actividad interna, generalmente horizontal, que asegura el buen funcionamiento de la empresa. Los procesos de soporte generalmente son invisibles para el cliente (beneficiario). Estos procesos incluyen administración financiera, administración de RR.HH., capacitación, etc.”[7] Ahora bien, viéndolo desde el punto de vista de desarrollo de software, los procesos de apoyo, brindan un soporte al proceso de

[5] http://es.kioskea.net/contents/qualite/processus.php3
[6] http://es.kioskea.net/contents/qualite/processus.php3
[7] http://es.kioskea.net/contents/qualite/processus.php3

desarrollo con el fin de mejorarlo y hacerlo proactivo. Algunos procesos de apoyo para el desarrollo de software pueden ser la gestión de seguimiento, control y verificación del mismo durante todo su ciclo de vida, entre otros.

6.2 PROCESO DE INGENIERÍA DE SOFTWARE

El proceso de ingeniería de software se define como "*un conjunto de etapas parcialmente ordenadas con la intención de logra un objetivo, en este caso, la obtención de un producto de software de calidad*" [Jacobson 1998].

El proceso de desarrollo de software "*es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo*" [Jacobson 1998].

Un conjunto coherente de políticas, estructuras organizacionales, tecnologías, procedimientos y artefactos qué son necesarios para concebir, desarrollar, instalar y mantener un producto software. [Fugetta, 2000]

"Conjunto de personas, estructuras de organización, reglas, políticas, actividades con procedimientos, componentes de software, metodologías y herramientas utilizadas o creadas específicamente para conceptualizar, desarrollar, ofrecer un servicio, innovar y extender un producto de software."[8]

[8] Diapositivas Calidad de Software. "2.3 Proceso Software" diapositiva 10

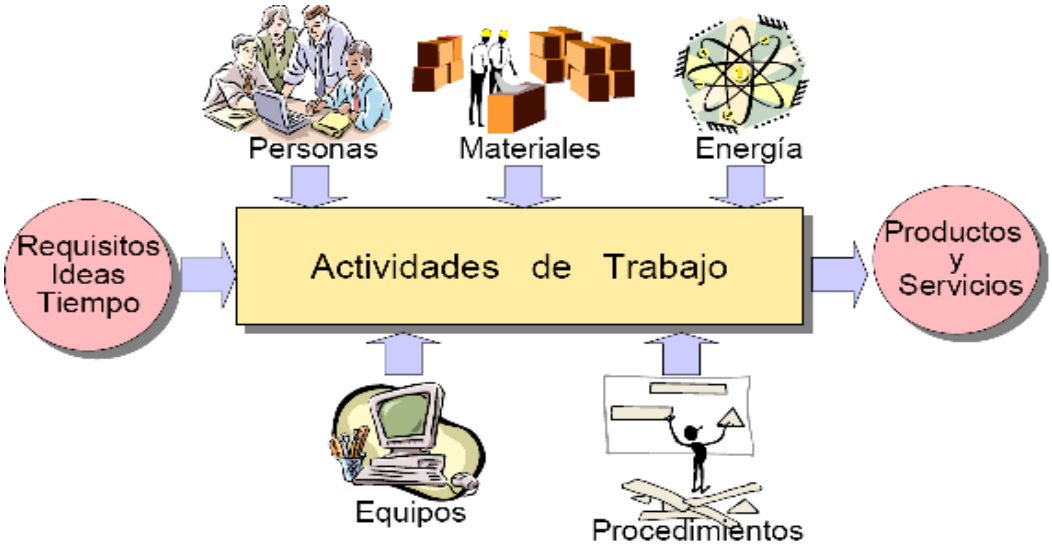


Figura 4. Actividades de trabajo
Fuente: Diapositivas Calidad de Software 2008-2

6.3 GESTIÓN DE PROYECTOS

"Un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único. La naturaleza temporal de los proyectos indica un principio y un final definidos. El final se alcanza cuando se logran los objetivos del proyecto o cuando se termina el proyecto porque sus objetivos no se cumplirán o no pueden ser cumplidos, o cuando ya no existe la necesidad que dio origen al proyecto."[9]

La gestión de proyectos tiene como objetivo orientar, coordinar, administrar recursos para dar culminación al proyecto dentro de un alcance, tiempo y costes definidos, poner metas, mantener a un equipo de trabajo orientado hacia esas metas, asignar tareas, garantizar el éxito del proyecto con seguimientos contantes a su evolución.

6.4 INGENIERÍA DE REQUISITOS

La ingeniería de requisitos consta de varias actividades cuyo fin es identificar, definir y documentar las necesidades del cliente y los interesados para darle cumplimiento a los objetivos del proyecto.

[9] Guía del PMBOK 4° edición

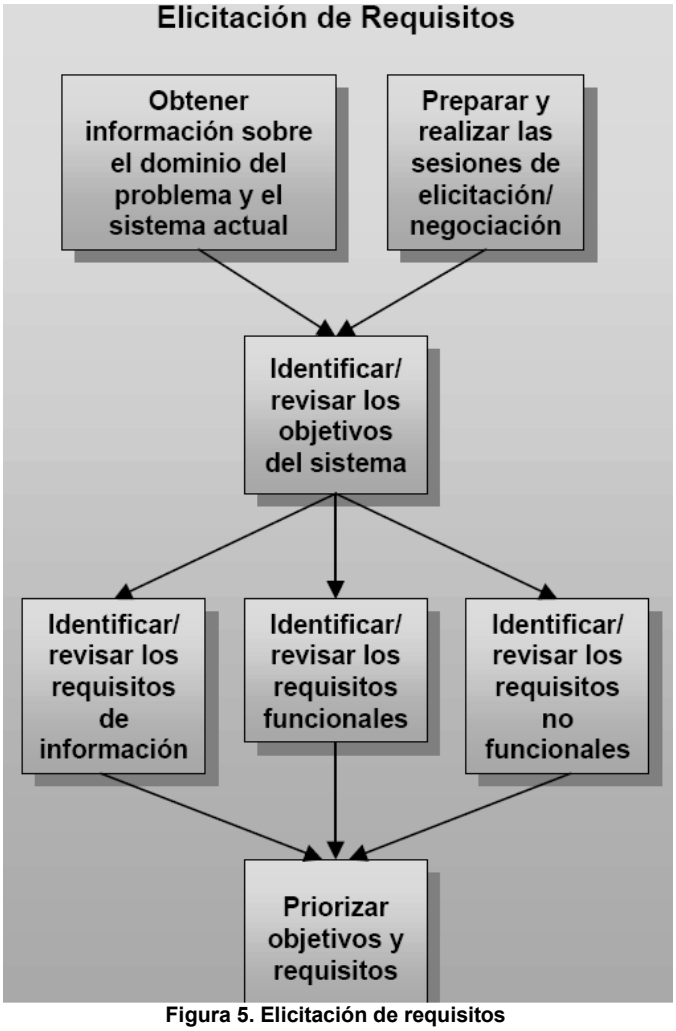


Figura 5. Elicitación de requisitos
Fuente: http://www.csae.map.es/csi/metrica3/gespro.pdf

Actividades:

- Elicitación de requisitos
- Análisis de requisitos
- Validación de requisitos
- Verificación de requisitos
- Formalización del documento
- Aprobación de requisitos
- Gestión y control de cambios a los requisitos

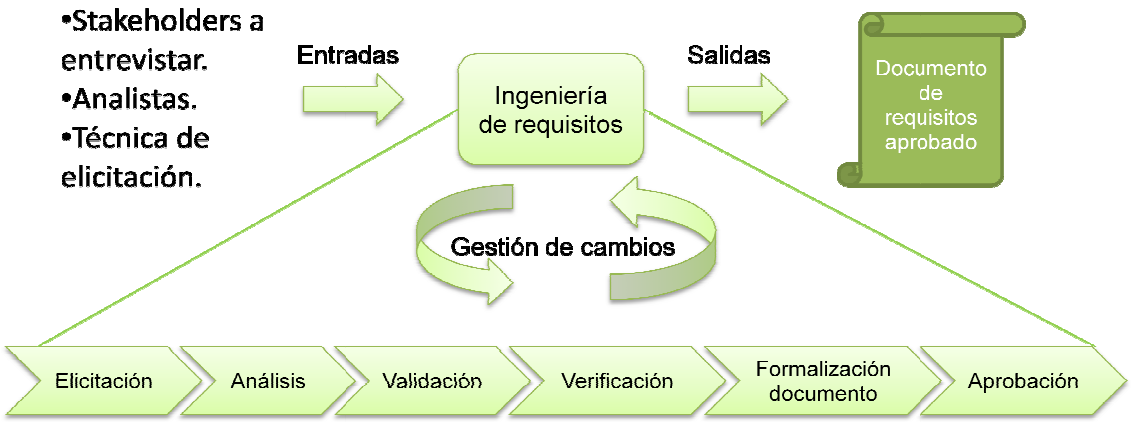


Figura 6. Proceso de elicitación de requisitos
Fuente: Elaboración propia

6.4.1 Elicitación de requisitos

Técnicas:

- Entrevistas,
- lluvia de ideas
- casos de uso
- prototipos

Ver Anexo 1.

6.4.2 ANÁLISIS DE REQUISITOS

Es el proceso mediante el cual se comprenden las necesidades del cliente. Se identifican y clasifican esas necesidades por requisitos del negocio, requisitos de usuario, funcionales, no funcionales, restrictivos para un mayor entendimiento del problema.

En el análisis de requisitos se debe hacer una abstracción de ellos, se deben priorizar y definir en lenguaje natural para que el usuario pueda entenderlos, y en forma técnica para que los desarrolladores puedan abstraerlos.

Se debe eliminar la ambigüedad y la inconsistencia, determinar los factores internos y externos que pueden afectar los requisitos y analizarlos de tal forma que se puedan mitigar los cambios.

A continuación se presenta un modelo de análisis de requisitos:

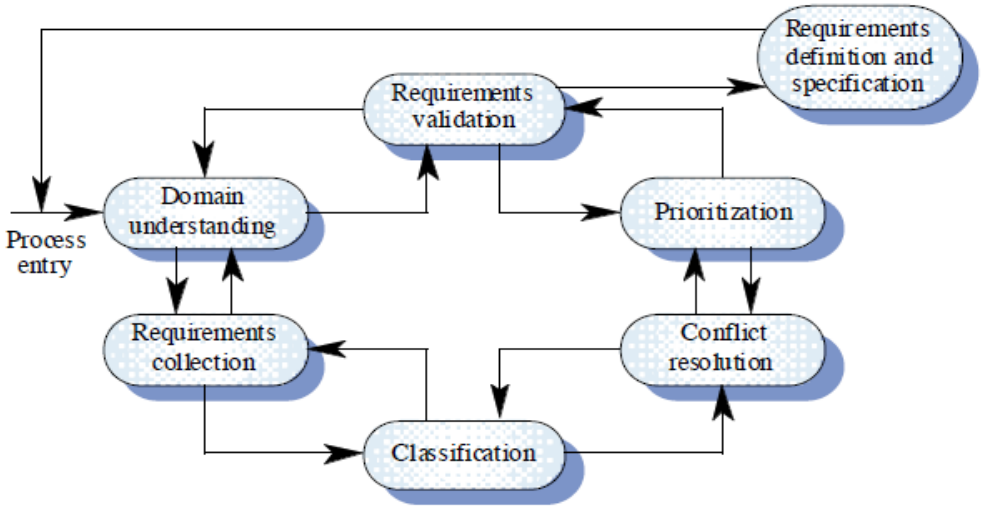


Figura 7. Proceso de análisis de requisitos.
Fuente: http://dis.unal.edu.co/~fgonza/courses/2003/ingSoft1/CAP5.pdf

6.4.3 Validación de requisitos

VALIDACIÓN (IEEE): el proceso de evaluar un sistema o componente durante o al final del proceso de desarrollo para determinar si satisface los requisitos especificados.

VALIDACIÓN (BOHEM): ¿se está construyendo el producto correcto?

VALIDACIÓN (POHL): la tarea de certificar que los requisitos especificados son consistentes con las intenciones de clientes y usuarios.

VALIDACIÓN (DURÁN): el conjunto de actividades encaminadas a llegar a un acuerdo entre todos los participantes en el que se ratifique que los requisitos elicitados y analizados representan realmente las necesidades de clientes y usuarios y que, por lo tanto, deberían llevar a la construcción de software útil.

Hay que validar que todos los requisitos (de negocio, de usuario, funcionales, no funcionales) sean reales, estén completos, sean coherentes. Hay que validar que los requisitos sí representen las necesidades del cliente.

6.4.4 Verificación de requisitos

VERIFICACIÓN (BOHEM): ¿se está construyendo correctamente el producto?
VERIFICACIÓN (POHL): comprobar que la especificación cumple los criterios de calidad oportunos.

6.4.5 Formalización del documento

Realizar un documento formal que incluya todos los requisitos y su clasificación para validar con el usuario, escrito en su lenguaje, sin palabras técnicas.

6.4.6 Aprobación de requisitos

El usuario debe aprobar los requisitos y constar esta aprobación con una aprobación al documento formal de requisitos.

6.4.7 Gestión y control de cambios a los requisitos

Se debe definir: una línea base para los requisitos, criterios de aceptación de requisitos y la negociación de los cambios de requisitos.

Los cambios en los requisitos se pueden dar por varios factores como, por ejemplo:

- Cambios tecnológicos
- Cambios en leyes o regulaciones
- Cambio en el alcance, costo y/o tiempo
- Caprichos de stakeholders
- Cambios en el mercado
- Cambios en el negocio

- Fallas o errores en la elicitación de requisitos,

Gestión de cambios de requisitos:

Línea base de requisitos:

"Es el conjunto de requisitos funcionales y no-funcionales que el equipo del proyecto se ha comprometido a implementar en una release específica. Es una versión aprobada de la especificación de requisitos del software."[10]

Se deben evaluar los cambios en términos de tiempo, costo y alcance para ver si son viables y necesarios. Para realizar un cambio en los requisitos se debe tener en cuenta el origen de este cambio. Este nuevo requisito o modificación de alguno debe pasar por los procesos de análisis, validación, verificación, formalización y aprobación que se mencionó anteriormente.

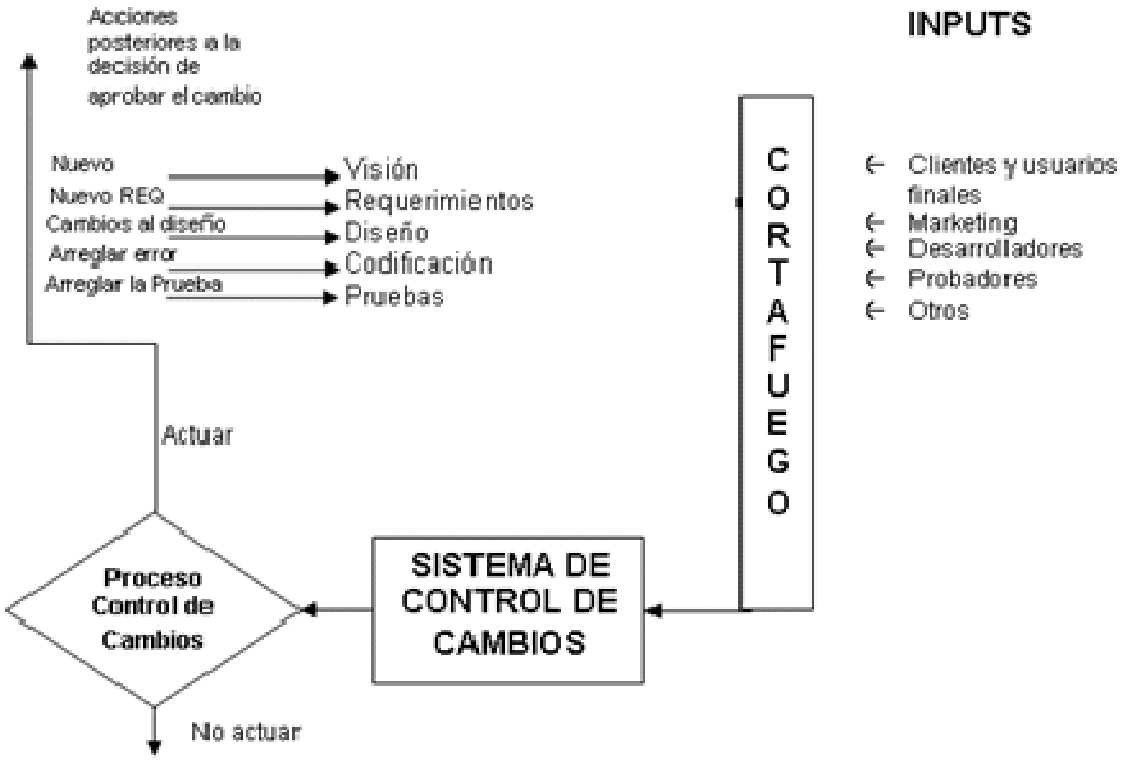


Figura 8. Canal de cambios
Fuente: http://www.monografias.com/trabajos41/requisitos-software/requisitos-software2.shtml

[10] http://www.monografias.com/trabajos41/requisitos-software/requisitos-software2.shtml

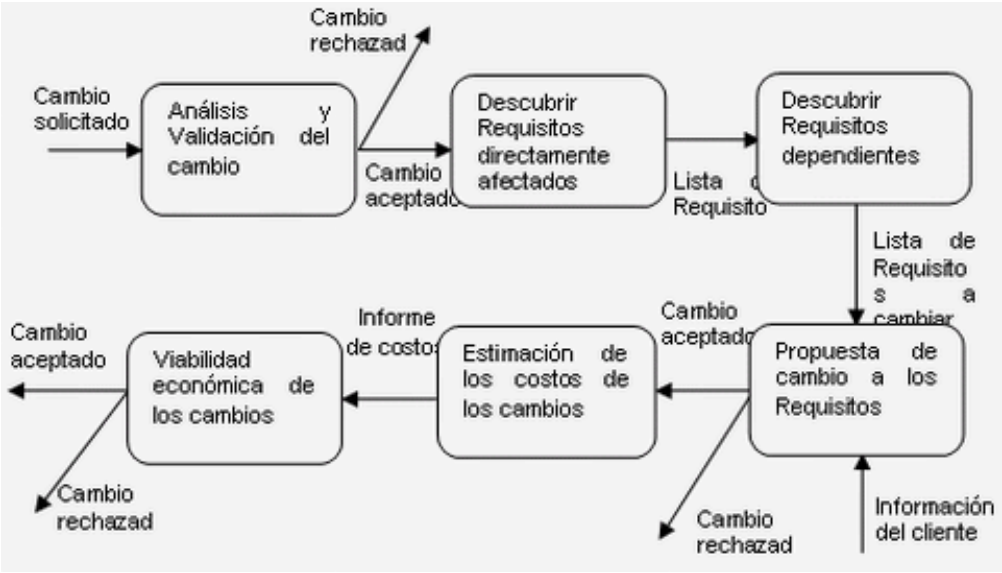


Figura 9. Proceso de control de cambios
Fuente: http://www.monografias.com/trabajos41/requisitos-software/requisitos-software2.shtml

6.5 ANÁLISIS Y DISEÑO

El análisis del sistema se lleva a cabo teniendo en cuenta ciertos principios:

- Representar el comportamiento del software.
- Dividir los modelos que representan la información, las funciones y el comportamiento

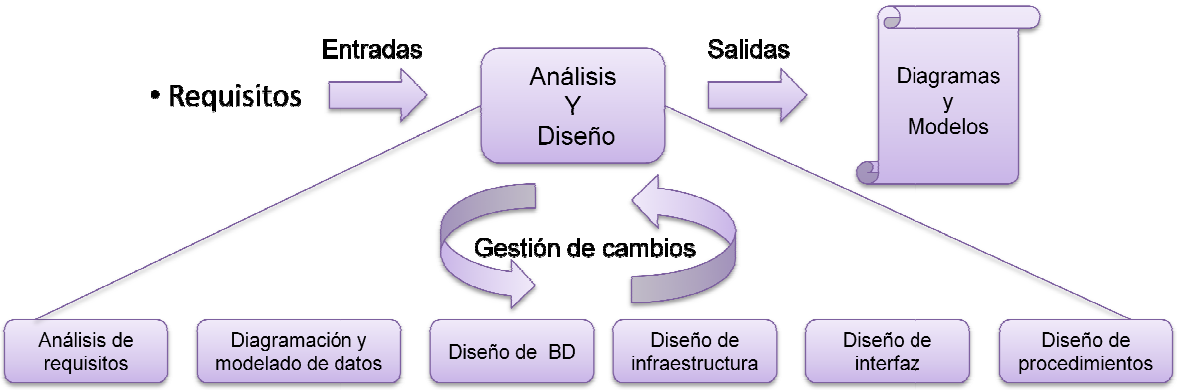


Figura 10. Proceso de Análisis y Diseño
Fuente: Elaboración propia

El Diseño de Sistemas se define como el proceso de aplicar ciertas técnicas y principios con el fin de definir un producto, un proceso o

un servicio, con suficientes detalles como para permitir su interpretación y realización.

"La etapa del Diseño del Sistema encierra cuatro etapas:

1. El diseño de los datos.
   Define la relación entre cada uno de los elementos estructurales del programa.
2. El Diseño Arquitectónico.
   Describe cómo se comunica el Software consigo mismo, con los sistemas que operan junto con él y con los operadores y usuarios que lo emplean.
3. El Diseño de la Interfaz.
4. El Diseño de procedimientos.
   Transforma elementos estructurales de la arquitectura del programa. La importancia del Diseño del Software se puede definir en una sola palabra Calidad, dentro del diseño es donde se fomenta la calidad del Proyecto. El Diseño es la única manera de materializar con precisión los requerimientos del cliente.

El diseño trasforma el modelo de dominio de la información creado durante el análisis en las estructuras de datos necesarios para implementar el Software.

El Diseño del Software es un proceso y un modelado a la vez. El proceso de Diseño es un conjunto de pasos repetitivos que permiten al diseñador describir todos los aspectos del Sistema a construir."[11]

6.6 CONSTRUCCIÓN Y CODIFICACIÓN

En construcción y codificación se crea la solución a nivel lógico (código fuente) que cumpla con los requisitos basándose en el análisis y diseño elaborados previamente.

[11] http://www.monografias.com/trabajos/anaydisesis/anaydisesis.shtml

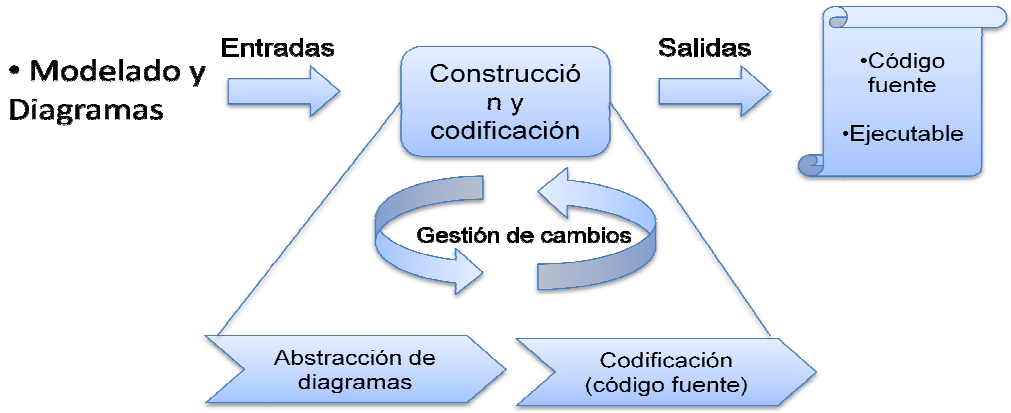


Figura 11. Proceso de construcción y codificación
Fuente: Elaboración propia

6.7 PRUEBAS

La idea de probar un sistema es realizar una serie de actividades donde se pretende encontrar fallas y vulnerabilidades a fin de corregirlas. Las pruebas son técnicas y funcionales.

Las pruebas técnicas consisten en hacer revisión del código, y las funcionales en revisar el cumplimiento de los requisitos y las necesidades del cliente. Si se encuentran errores en cualquiera de las dos pruebas, se debe enviar a codificación para que sea modificado.

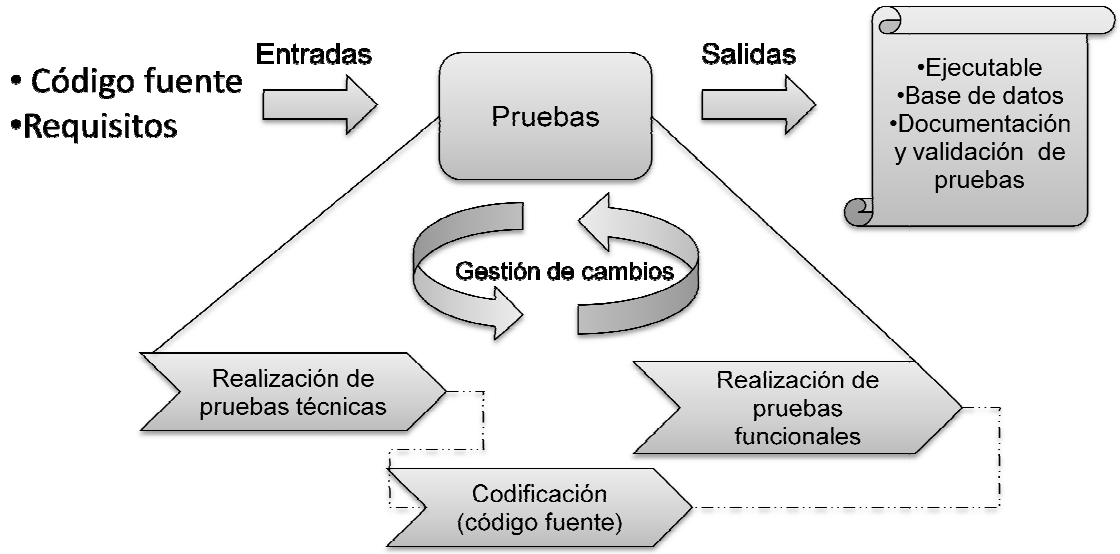


Figura 12. Proceso de Pruebas
Fuente: Realización propia

Entre las diversas pruebas que se le efectúan al software se pueden distinguir principalmente:

- “Prueba unitarias: Consisten en probar o testear piezas de software pequeñas; a nivel de secciones, procedimientos, funciones y módulos; aquellas que tengan funcionalidades específicas. Dichas pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código, mucho más reducidas que el conjunto, y que tienen funciones concretas con cierto grado de independencia, estas pruebas las debe realizar el desarrollador.
- Pruebas de integración: Se realizan una vez que las pruebas unitarias fueron concluidas exitosamente; con éstas se intenta asegurar que el sistema completo, incluso los subsistemas que componen las piezas individuales grandes del software, funcionen correctamente al operar e interoperar en conjunto.”[12]
- “Caja blanca (pruebas estructurales) En estas pruebas se observa siempre el código, y las pruebas se dedican a ejecutar con ánimo de "probarlo todo". Esta noción de prueba total se formaliza en lo que se llama "cobertura" y no es sino una medida porcentual de ¿cuánto código hemos cubierto?
- Caja negra (pruebas funcionales) Las pruebas de caja negra se centran en lo que se espera de un módulo, es decir, intentan encontrar casos en que el módulo no se atiene a su especificación. Por ello se denominan pruebas funcionales, y el probador se limita a suministrarle datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el módulo por dentro. Las pruebas de caja negra están especialmente indicadas en aquellos módulos que van a ser interfaz con el usuario (en sentido general: teclado, pantalla, ficheros, canales de comunicaciones, etc) Este comentario no obsta para que sean útiles en cualquier módulo del sistema.
- Pruebas de aceptación. Estas pruebas las realiza el cliente. Son básicamente pruebas funcionales, sobre el sistema completo, y buscan una cobertura de la especificación de requisitos y del

[12] http://es.wikipedia.org/wiki/Software

manual del usuario. Estas pruebas no se realizan durante el desarrollo, pues sería impresentable de cara al cliente, sino una vez pasadas todas las pruebas de integración por parte del desarrollador "[13]

6.8 IMPLEMENTACIÓN

Se hace entrega del producto y se realizan las capacitaciones pertinentes.

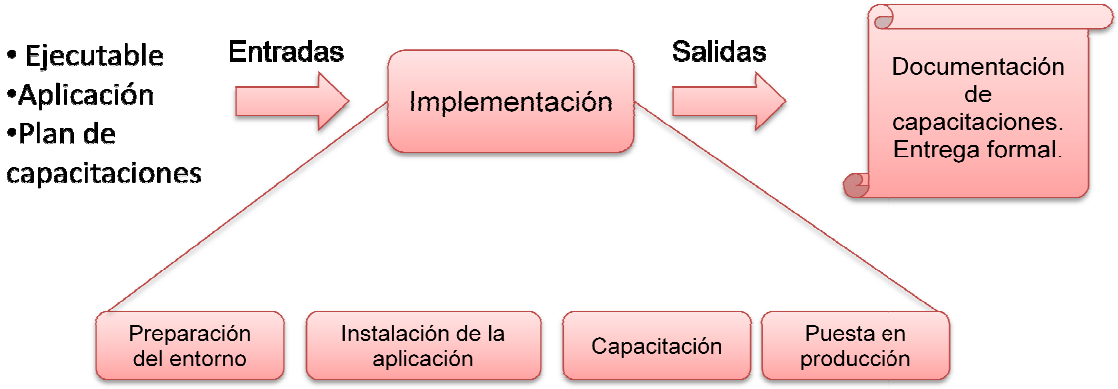


Figura 13. Proceso de Implementación
Fuente: Elaboración propia.

Para la implementación del software se deben tener en cuenta las siguientes actividades:

- Preparar el entorno operativo, es decir, proveer el software y hardware apto para el nuevo producto.
- Instalación de la versión del producto
- Capacitaciones a usuarios finales
- Puesta en producción

6.9 MANTENIMIENTO Y SOPORTE

Es el proceso post-implementación donde se da control, soporte, mejoras, optimizaciones, etc. al producto desarrollado. El tiempo y el costo del mantenimiento dependen de la realización del producto. Si tuvo

[13] http://www.lab.dit.upm.es/~lprg/material/apuntes/pruebas/testing.htm#s1

un buen desarrollo a lo largo del ciclo de vida, el tiempo y costo de mantenimiento será menor.

6.10 GESTIÓN DE CAMBIOS

Los cambios en el desarrollo de software pueden ocurrir en cualquier momento del ciclo de vida. La gestión del cambio se dedica entonces a identificar y controlar los cambios que se presenten. Debe definir políticas para manejar los cambios y para negociarlos con los stakeholders.

La gestión de cambios debe realizar actividades de seguimiento y control a todo el ciclo de vida de desarrollo del producto para poder controlar los cambios que se puedan presentar. Debe evaluar qué tan conveniente es aplicar un cambio en cuestión de tiempo, costo y alcance según la línea base que se tiene.

6.11 GESTIÓN DE LA DOCUMENTACIÓN

Esta etapa identifica, organiza, controla, verifica, todos los entregables que pueden surgir en la elaboración de un producto software, como son las actas, las aprobaciones, los documentos de entrada y salida de cada fase del ciclo de desarrollo, los manuales de usuario, técnico, etc. de modo que sean coherentes, completos, correctos, para poder controlar mejor la elaboración del producto y sus entregables.

6.12 GESTIÓN DE RIESGOS

La gestión de riesgos permite adelantarse a los posibles eventos que pueden cambiar el objetivo del desarrollo o afectar el costo, el tiempo y/o el alcance. Con un plan de gestión de riesgos se busca realizar una estrategia que permita identificar, mitigar, supervisar e incluso evitar que ocurra un riesgo.

6.13 GESTIÓN DE LA CONFIGURACIÓN

Esta etapa se encarga del versionamiento y cambios al producto final, es decir, actualizaciones necesarias, optimizaciones, etc. con el fin de tener un seguimiento de los cambios generales y necesarios al producto software.

6.14 METODOLOGÍAS PSP Y TSP

La información que se presenta a continuación fue complementada con entrevistas realizadas a mexicanos expertos en los temas PSP y TSP llevadas a cabo por el profesor Rafael Rincón Bermúdez entre septiembre y diciembre del año 2009.

Estas metodologías fueron propuestas en el año 1995 por Watts Humphrey, también llamado “el padre de la calidad del software”. El ideal de estas metodologías es complementar las prácticas de CMMI, ya que CMMI apunta al ¿QUÉ? Y el PSP y el TSP, apuntan al ¿CÓMO?

6.14.1 PSP (Personal Software Process)

> “El Proceso Personal de Software es un conjunto de prácticas disciplinadas para la gestión del tiempo y mejora de la productividad personal de los programadores o ingenieros de software en tareas de desarrollo y mantenimiento de sistemas.”[14]
>
> Esta metodología permite a los ingenieros de sistemas mejorar la forma como construyen software, partiendo de la optimización de la planeación, la calidad, una mejor estimación de tiempo y costo y buen manejo de estos para así poder alcanzar una buena productividad y alta calidad en los productos partiendo del contexto de un trabajo individual. “Con el PSP los desarrolladores

[14] http://es.wikipedia.org/wiki/Personal_Software_Process

utilizan procesos definidos y medibles. Se toma información de tamaño, tiempo y defectos al momento de realizar el trabajo."[15]

6.14.1.1 OBJETIVOS DEL PSP[16]

- Ayudar a los profesionales de software a utilizar de forma consistente una buena práctica de ingeniería.
- Mostrar a los ingenieros cómo planear y realizar un seguimiento de su trabajo, utiliza un proceso definido y medido.
- Establecer metas mensurables y hacer seguimiento en contra de estos objetivos.
- Mostrar cómo gestionar la calidad desde el inicio del trabajo, cómo analizar los resultados de cada trabajo, y cómo utilizar los resultados para mejorar el proceso para el próximo proyecto.

6.14.1.2 CARACTERISTICAS DEL PSP

"En PSP todas las tareas y actividades que el ingeniero de software debe realizar durante el proceso de desarrollo de un producto de software están puntualmente definidas en un conjunto de documentos conocidos como scripts. Los scripts son el punto medular de PSP, por lo que se hace mucho énfasis en que deben ser seguidos en forma disciplinada, ya que de ello dependerá el éxito de la mejora que se busca. Gran parte de las tareas y actividades definidas en los scripts generará en su realización un conjunto de datos, fundamentalmente de carácter estadístico. La aplicación de PSP en varios procesos de desarrollo y el análisis de la información estadística generada en cada uno de éstos

[15] http://www.kernel.com.mx/documentos/psp_tsp.pdf
[16] Watts Humphrey. The Personal Software Process. pág. 15
http://www.sei.cmu.edu/reports/00tr022.pdf

permitirán al ingeniero de software identificar, tanto sus fortalezas como sus debilidades, y crecer a través de un proceso de autoaprendizaje y auto mejora."[17]

6.14.1.3 [18]PRINCIPIOS DE CALIDAD DEL PSP

- Cada ingeniero es diferente, para ser más eficaz, los ingenieros deben planear su trabajo y deben basar sus planes en sus propios datos personales.
- Para mejorar consistentemente el desempeño, los ingenieros deben usar procesos personales medibles y bien definidos.
- Para obtener productos de calidad, el ingeniero debe asumir la responsabilidad personal de la calidad de sus productos. Los buenos productos no se obtienen por azar, sino como consecuencia de un esfuerzo positivo para hacer un trabajo de calidad.
- Es menos costoso detectar y corregir errores al principio.
- Es más efectivo prevenir defectos que detectarlos y corregirlos.
- Trabajar bien es siempre la forma más rápida y barata de trabajar.

6.14.1.4 [19]Pasos o niveles del PSP

El PSP propone scripts que se organizan en cuatro niveles, identificados del 0 al 3, donde cada nivel tiene un conjunto de aspectos a mejorar del proceso de desarrollo de software**.** Los scripts no pueden utilizarse en forma separada o desordenada.

[17] http://html.rincondelvago.com/personal-software-process-psp.html

[18]

19 http://ingsw.ccbas.uaa.mx/sitio/images/material/psp.htm

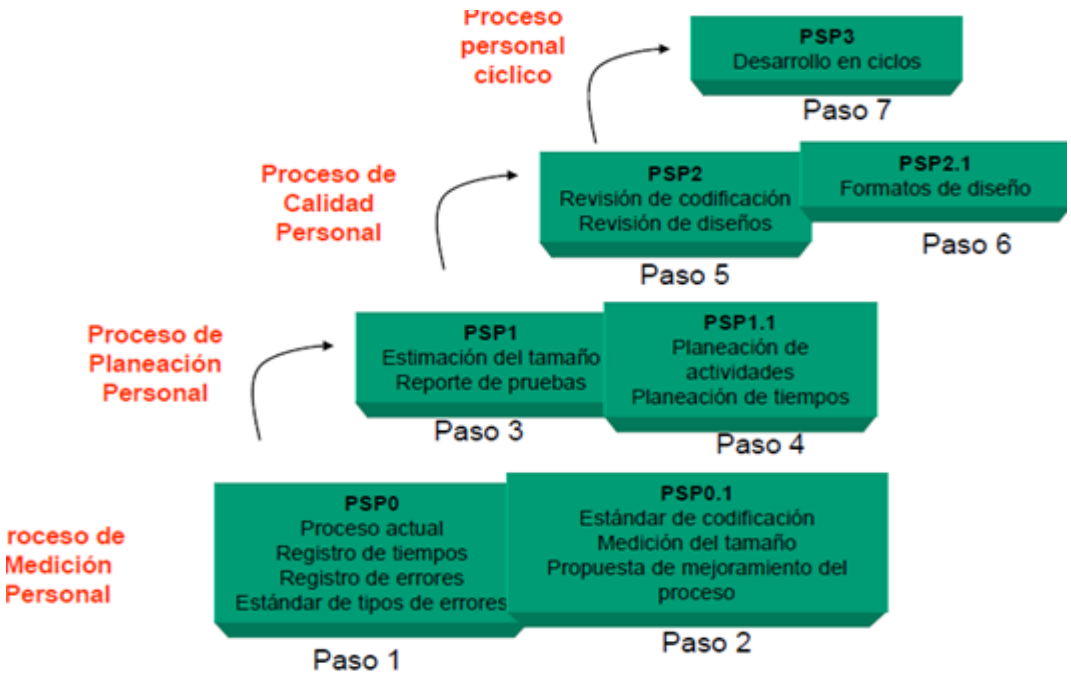


Figura 14. Niveles de PSP
Fuente: Diapositivas Calidad del PS, diapositiva # 39.

PSP 0. Línea base o punto de inicio del proceso.

- Identificar actividades: definición, secuencia
- Bases mejoras: planeación, evaluación, resultados
- Documentar proceso:
  - Formas de:

    Actividades, Tiempos, Defectos, Resumir planes, resultados.

PSP 0.1

Registrar tamaño del producto y hacer un histórico:

- Estándares de código
- Mediciones de tamaño
- Propuesta de mejora del proceso

Registrar problemas y mejoras de propuestas

PSP 1. Proceso personal de planeación

Mejora la planeación:

- Con la estimación tamaño del producto (histórico)
- Decidir en base a reportes de pruebas

PSP 1.1

Mejora la planeación:

- Con la estimación de recursos
- Introducción de calendarizar, plasmar el plan con números, un presupuesto.

Objetivos:

- Relacionar el tamaño de los programas con el tiempo de desarrollo
- Aprender a trazar metas alcanzables
- Planear un orden de trabajo
- Establecer bases para hacer seguimiento

PSP 2. Gestión de la calidad

Mejora la ejecución:

- Detección temprana de defectos, con base en la predicción de estos.
- Revisiones de diseño
- Revisiones de código
- Uso de checklists (Listas de verificación)

PSP 2.1

Criterios de diseño

- Se establecen técnicas y criterios para verificar la completitud y consistencia del diseño.

Objetivos:

- Conocer los errores que se están cometiendo
- Entender causas y consecuencias de los errores.

PSP 3. Proceso personal cíclico**.**

Escalar PSP2 a proyectos más grandes.

Estrategia:

- Dividir programas grandes en partes solucionables usando PSP2.
- Los programas son diseñados para ser desarrollados de forma incremental.
- En la primera iteración desarrolla un módulo base.
- En cada iteración se aplica un ciclo completo PSP2 (diseño, codificación, compilación, pruebas)
- Cada iteración cuenta con la calidad de la iteración anterior para hacer el programa escalable.

6.14.1.5 [20]Ciclo de vida PSP

Figura 15. Ciclo de vida PSP
Fuente: http://ingsw.ccbas.uaa.mx/sitio/images/material/psp.htm

20 http://ingsw.ccbas.uaa.mx/sitio/images/material/psp.htm

REQUISITOS

- Descripción del problema
- Especificación de componentes
- Formas de proceso
- Estimadores del tamaño del producto y tiempos en base a históricos

FASE DE PLANEACION

Entradas: Tipificación requerimientos, diseño conceptual, patrones de estimaciones de tamaño, resumen parte cíclico, seguimiento.

Actividades: Especificaciones externas, diseño modular, prototipos, estrategia de desarrollo y documentación, seguimiento.

Salidas: Diseño de programa, escenarios operacionales, especificación de funciones y lógica, resumen cíclico, seguimiento y estrategias de pruebas y ciclo.

FASE DE REVISIÓN Y VALIDACIÓN DEL DISEÑO

Entradas: Programa de diseño, escenarios operacionales, especificación de funciones y lógica, resumen cíclico, seguimiento y estrategia de pruebas y ciclo.

Actividades: Diseño de apariencia, verificación de máquinas y lógica, consistencia del diseño, reúso, estrategia de verificación, detectar errores.

Salidas: Diseño de alto nivel, registro de seguimiento, tiempos y defectos.

FASE DE DESARROLLO

Entradas: Diseño de alto nivel, registro de seguimiento, tiempos y defectos, ciclo de desarrollo, estrategia de pruebas, patrones de operación y función.

Actividades: Diseño de módulos, revisión de diseño, código, revisión de código, compilación, pruebas, aseguramiento de calidad y del ciclo.

Salidas: Módulos de software, patrón de diseño, lista de verificación de código y diseño, resumen del ciclo, patrón de reporte de pruebas, registro de tiempo, defectos y seguimiento.

FASE DE POSTMORTEM

Entradas: Definición de problema y requerimientos, plan de proyecto y de ciclo, producto de software, patrón de diseño, lista de verificación de código y diseño, resumen del ciclo, patrón de reporte de pruebas, registro de tiempo, defectos y seguimiento.

Actividades: Defectos previstos, removidos, tamaño, tiempo del producto.

Salidas: Producto, listas de verificación, plan de proyecto y ciclo, patrón de reporte de pruebas y diseño, forma con propuesta de mejora, registro seguimiento pruebas y tiempo.

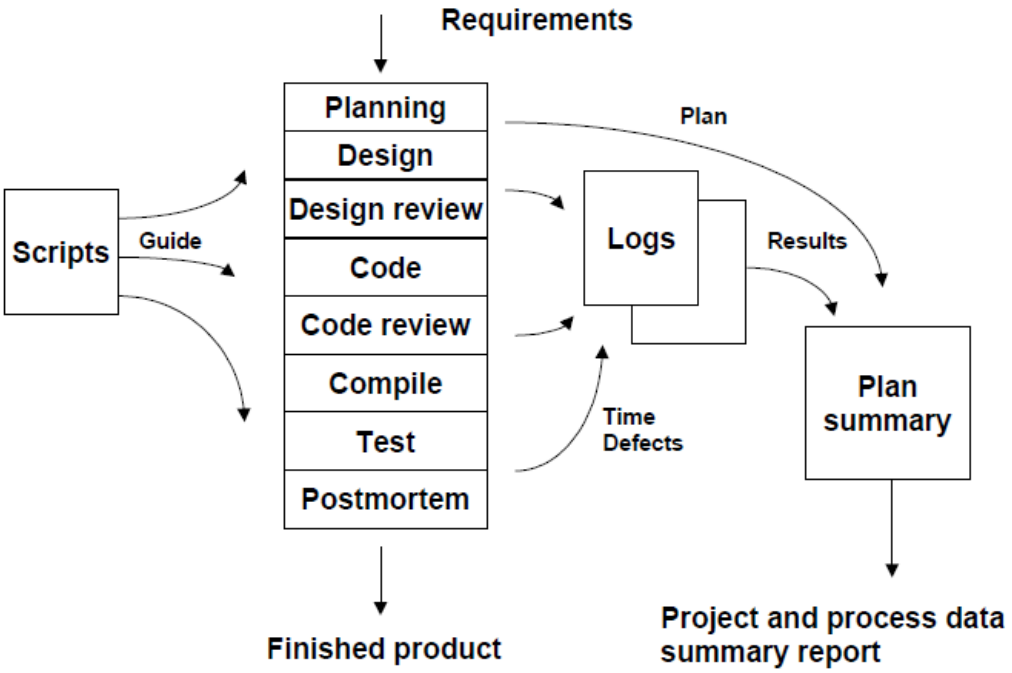


Figura 16.Flujo del proceso PSP

Fuente http://www.sei.cmu.edu/reports/00tr022.pdf

Todas las revisiones que se hagan en cada fase se deben registrar, esto con el fin de formar una base histórica para futuros desarrollos. Teniendo estos históricos personales organizados se puede dar una mejor estimación de tiempos y costos en la realización de futuros productos software partiendo de los propios datos.

6.14.1.6 COMPONENTES DEL PSP

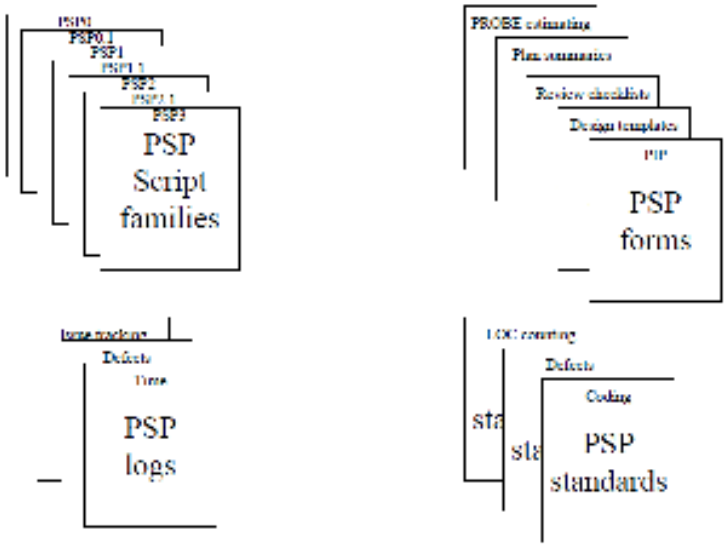


Figura 17.Componentes de PSP
Fuente: http://www.sei.cmu.edu/reports/00tr022.pdf

[21]Definiciones:

Scripts: los scripts del proceso definen los pasos para cada parte del proceso. “Contienen los criterios de entrada de cada etapa del proceso, las etapas/pesos y los criterios de salida. Su propósito es guiar al ingeniero para usar el proceso.”

Formas: Proveen un medio consistente y conveniente para recolectar y almacenar datos.

Medidas: Miden el proceso y el producto. Proveen la visibilidad de cómo está trabajando el proceso y del estado actual del trabajo.

[21] Diapositivas Calidad del PS 2008-2. Diap. 37

Estándares: Proveen definiciones consistentes que guían el trabajo y la recolección de datos.

6.14.1.7 [22]RECOLECCION DE DATOS

En PSP los ingenieros utilizan datos para supervisar su trabajo y ayudarles a hacer mejores planes. Para ello, se reúnen datos sobre el tiempo que pasan en cada fase del proceso, el tamaño de los productos que producen y la calidad de estos productos. Estos temas se discuten a continuación.

- Medidas de Tiempo

  En PSP los ingenieros utilizan el registro de tiempo para medir el tiempo empleado en cada proceso fase. En este registro se tiene en cuenta el momento en que se comenzó a trabajar en una tarea, el momento en que se dejó la tarea, y cualquier tiempo de interrupción. Por ejemplo, una interrupción podría ser una llamada telefónica, un breve descanso, o alguien que interrumpió para hacer una pregunta.

  Para el seguimiento del tiempo, los ingenieros realizan el seguimiento del esfuerzo que se gastó en las tareas del proyecto.

- Medidas de Tamaño.

  El tiempo que se tarda en desarrollar un producto es determinado en gran medida por el tamaño de dicho producto. Cuando se utiliza PSP, los ingenieros hacen una primera estimación de los tamaños de los productos que planean desarrollar. Luego, cuando se llevan a cabo, miden el tamaño de los productos que produjeron. Esto proporciona a los ingenieros los datos de talla que

[22] Fuente: traducción y adaptación de http://www.sei.cmu.edu/reports/00tr022.pdf

necesitan para hacer estimaciones precisas sobre el tamaño. Sin embargo, para que estos datos sean útiles, la medida de tamaño debe correlacionarse con el tiempo de desarrollo para el producto. Mientras que las líneas de código (LOC) son la principal medida de tamaño en PSP, cualquier medida de tamaño que se utilice debe proporcionar una correlación razonable entre el tiempo de desarrollo y tamaño del producto. Asimismo debe permitir la medición automática del tamaño real del producto.

- Medidas de calidad.
  El foco principal de la calidad de PSP está en los defectos. Para gestionar los defectos, los ingenieros necesitan datos sobre los defectos que se inyectan, las fases en que se inyectan, las fases en las que se encuentran y corrigen, y cuánto tiempo tomó para corregirlos. Con el PSP, los ingenieros deben llevar un registro de todos los defectos que encuentran en todas las fases, incluidas las revisiones, inspecciones, compilación y pruebas.

6.14.1.8 [23]INSTRUCCIONES PARA LA UTILIZACIÓN DEL PSP

A) Utilización de un Cuaderno de registro de tiempos

El objetivo de registrar el tiempo es el de obtener datos de cómo se trabaja realmente, la forma y el procedimiento utilizados para reunir los datos no es tan importante mientras los datos sean exactos y complejos. Es recomendable controlar el tiempo en unidades más pequeñas, mayormente se lo controla mediante horas, es más fácil controlar el tiempo en minutos.

---

23 http://www.postgradoinformatica.edu.bo/enlaces/investigacion/pdf/INGSW3_55.pdf?PHPSESSID=27279432238a87eb281829d637739354

El formato del cuaderno de registros de tiempo es el siguiente:

Fecha	Hora		Tiempo de Interrupción	Tiempo	Descripción Actividad	Comentarios	C.	U
	Inicio	Fin						

Figura 18. Formato del cuaderno de registro de tiempos
Fuente:
http://www.postgradoinformatica.edu.bo/enlaces/investigacion/pdf/INGSW3_55.pdf?PHPSESSID=27279432238a87eb281829d637739354

Donde:

Fecha: Se anota en esta comuna la fecha de realización de alguna actividad

Comienzo: Se anota en esta comuna la hora de inicio de la actividad

Fin: Se anota en esta comuna la hora de finalización de la actividad

Interrupción: pérdida de tiempo debida a una interrupción

Tiempo: tiempo dedicado a una actividad en minutos

Actividad: descripción de la actividad

Comentarios: descripción completa de la actividad

C: Completado

U: Numero de unidades de la tarea dedicada

B) Gestión de la Interrupciones

La forma de gestionar las interrupciones en el Cuaderno de Registro de Tiempos consiste en anotarlas en la columna Tiempo de Interrupción. Puesto que el tiempo de las interrupciones no es tiempo de trabajo productivo, se deben controlar las interrupciones. Si la cantidad de este

tiempo fuese constante no habría que hacer mucho para gestionarlo. Sin embargo, el tiempo de las interrupciones es muy variable. Si no se mide, habría que añadir un número aleatorio en todos los datos de tiempos, lo que haría más difícil utilizar estos datos para planificar o gestionar el tiempo. Esos datos registrados pueden utilizarse para comprender con qué frecuencia se interrumpe el trabajo. Las interrupciones no son solamente un despilfarro de tiempo, sino que rompen el ritmo de pensamiento, llevando a la ineficiencia y al error. Comprender cómo se es interrumpido ayuda a mejorar la calidad y eficiencia del trabajo.

C) Control de las tareas Finalizadas

Para controlar cómo se gasta el tiempo se necesita controlar los resultados producidos. Para la asistencia a clases o reuniones, por ejemplo, un registro del tiempo sería adecuado. Cuando se desarrollan programas, se leen los capítulos de un libro o se documenta un trabajo, se necesita saber cuánto trabajo se ha realizado, y así se podrá calcular la productividad de la tarea. Con este conocimiento se puede mejorar la planificación de futuros trabajos. · Las columnas C y U de la derecha del Cuaderno de Registro de tiempos significan Completado (C) y Unidades (U). Estas columnas ayudan a identificar rápidamente el tiempo dedicado a las distintas tareas y lo que se ha hecho. Por unidad se entiende unidad de trabajo. Cuando se ha leído un capítulo se ha completado una unidad de trabajo. Un diagrama acabado es otra unidad de trabajo. Para rellenar la columna C hay que comprobar cuando se ha terminado una tarea.

Para tener unos registros de tiempo exactos, es importante completar las columnas C y U cada vez que se finalice una tarea que tenga resultados medibles.

D) Ideas para registrar el tiempo

El control del tiempo es sencillo. Unos pocos trucos, sin embargo, pueden ayudar a hacerlo de forma más consistente y precisa:

- Llevar siempre el cuaderno de notas.
- Cuando ocasionalmente se olvide registrar la hora de comienzo, la hora de fin o la duración de la interrupción, realizar una estimación tan pronto como se recuerde.
- Se puede utilizar un cronómetro para controlar las interrupciones. Puede parecer excesivamente preciso pero es más sencillo registrar el tiempo de inicio y finalización de cada interrupción.
- Resumir el tiempo puntualmente. Utilizar el Resumen Semanal de actividades para resumir semanalmente el tiempo en este curso.
- Los registros personales se pueden llevar a mano para más comodidad, pero los resúmenes periódicos se realizarán en ordenador.

E) Resumen periódico de las actividades

Para hacer una planificación correcta es importante conocer cómo se gasta el tiempo. El primer paso es registrar el tiempo utilizando el Cuaderno de Registro de Tiempos. Después de reunir los datos de tiempo una o dos semanas, se empezará a ver cómo se emplea el tiempo. Puesto que los registros de tiempos son muy detallados para el propósito de la planificación, se necesita resumir los datos de una forma más útil. En la siguiente tabla se muestra el formato del resumen semanal que deberá hacer cada miembro del grupo:

Los datos de esta tabla se resumirán cada tres semanas en una tabla similar que englobará los datos de todos los miembros del equipo. Todas las tablas (los cuadernos de

registro de tiempos y los resúmenes semanales de cada miembro del grupo, así como el resumen de todo el grupo) deberán ser firmados a efectos de control por el profesor de prácticas correspondiente cada tres semanas, y se entregarán al final del cuatrimestre con la segunda entrega del trabajo de la asignatura. Se podrán complementar, de forma voluntaria, con gráficas comparativas, de evolución del trabajo, etc.

F) Actividades Generales

De forma general, se utilizarán las siguientes actividades para controlar el proceso:

- Gestión del proyecto (tareas relacionadas con la planificación, la elaboración de las tablas de registros de tiempo, reparto de tareas en el grupo)
- Actividades por artefactos (diagramas, descripciones) la elaboración de cada uno de los distintos tipos de artefactos se considerará una actividad (es decir, "Diagramas de Interacción" será una única actividad, aunque esté compuesta de la elaboración de diferentes diagramas).
- Documentación: recopilación de diagramas, formateado del trabajo, elaboración de los apéndices (si es que los hay)

6.14.2 TSP (Team Software Process)

Proceso de desarrollo para equipos de ingenieros, basado en CMMI.

"Es una metodología para dirigir el trabajo de mejora y desarrollo de software, además de establecer un entorno donde el trabajo efectivo del equipo sea normal y natural."[24]

[24] http://www.slideshare.net/dlpoma/team-software-process-tsp

"En combinación con el Personal Software Process (PSP), el llamado Team Software Process (TSP) proporciona un marco de trabajo de procesos definidos que está diseñado para ayudarle a equipos de gerentes e ingenieros a organizar y producir proyectos de software de gran escala, que tengan tamaños mayores a varios miles de líneas de código."[25]

6.14.2.1 ¿QUÉ ES UN EQUIPO?

"Al menos dos personas que están trabajando juntos por una meta/objetivo/misión común, donde a cada persona se le ha asignado roles o funciones específicas a desarrollar, y el cumplimiento de la misión requiere de algún tipo de dependencia entre los miembros del grupo."[26]

"Un equipo es más que un grupo de personas que trabajan juntos. El trabajo en equipo requiere práctica y habilidades especiales. Los equipos requieren procesos comunes, necesitan objetivos previamente convenidos, y requieren orientación y liderazgo efectivo. Los métodos para orientar y dirigir equipos son bien conocidos, pero no son obvios. El Software Engineering Institute (SEI) está apoyando el TSP como una forma de guiar a los ingenieros y sus gerentes en el uso de métodos eficaces para el trabajo en equipo."[27]

6.14.2.2 [28]OBJETIVOS DE TSP

- Minimizar costos
- Maximizar la calidad del software

[25] http://es.wikipedia.org/wiki/Team_Software_Process
[26] http://www.slideshare.net/dlpoma/team-software-process-tsp
[27] Fuente: Traducción y adaptación de http://www.sei.cmu.edu/reports/00tr023.pdf
[28] Diapositivas Calidad del PS. 2008-2. Diap. 46

- Integrar equipos independientes de alto rendimiento que planeen y registren su trabajo, establezcan métricas y sean dueños de sus procesos y planes.
- Mostrar a los gerentes cómo monitorear y motivar a sus equipos de trabajo y cómo ayudarlos a alcanzar su máxima productividad.
- Acelerar la mejora continua de los procesos.
- Proveer una guía para el mejoramiento en las organizaciones maduras.

6.14.2.3 [29]EQUIPOS EFECTIVOS

Para ser eficaces, los equipos deben estar debidamente capacitados y ser capaces de trabajar como unidades cohesivas. Los equipos eficaces tienen ciertas características comunes:

- Los miembros deben ser expertos.
- El objetivo del equipo es importante, definido, visible y realista.
- Los recursos del equipo deben ser adecuados para el trabajo.
- Los miembros se deben sentir motivados y comprometidos a cumplir la meta del equipo.
- Los miembros deben cooperar y apoyarse mutuamente.
- Los miembros deben ser disciplinados en su trabajo.

Otra característica de los equipos eficaces es su capacidad para innovar. La innovación es algo más que pensar en ideas brillantes. La innovación hace parte de casi todas las tareas de ingeniería, los equipos innovadores deben tener personal cualificado y capaz, que estén motivados. Tienen que ser creativos, flexibles y disciplinados. Deben esforzarse por

[29] Fuente: traducción y adaptación de http://www.sei.cmu.edu/reports/00tr023.pdf

cumplir con la exigencia de las necesidades cambiantes de los programas. Deben controlar los costos y horarios para medir su progreso.

6.14.2.4 ROLES DE UN EQUIPO TSP

Los equipos TSP están compuestos por un líder, un gestor de desarrollo, uno de calidad, uno de planeación y otro de configuración.

La conformación de los equipos puede implicar que una sola persona maneje uno o varios roles para poder distribuir la carga entre todos.

[30]**Roles**:

Líder:

Objetivo:

Dirigir al equipo y asegurar que todos reporten sus datos de las actividades realizadas y completen su trabajo

Responsabilidades principales:

- Motivar al equipo, dirigir las reuniones semanales.
- Generar los informes semanales del avance del equipo
- Verificar la entrega puntual y completa de tareas.
- Identificar tareas semanales
- Mantener el documento del proyecto
- Mantener informado al instructor sobre el avance del proyecto
- Resolver conflictos

Gestor de desarrollo:

Objetivo:

[30] http://www.slideshare.net/guestb3b081/roles-para-el-tsp-3810337 y http://www.uv.mx/rrojano/desarrollo_dos/Roles%20para%20TSP.pdf

Guiar al equipo en diseño y desarrollo del producto

Responsabilidades principales:

- Dirigir las fases de desarrollo siguiendo los estándares propuestos y generando los productos de cada fase
- Integrar el trabajo de todos
- Dirigir en análisis de alto nivel, requerimientos, diseño, desarrollo y pruebas.
- Auxiliar en las estimaciones de tamaño y tiempo

Gestor de planificación:

Objetivo:

Guiar al equipo en la planificación y seguimiento del trabajo.

Responsabilidades principales:

- Apoyar y guiar la planeación y el seguimiento
- Efectuar la planificación de común acuerdo con el equipo y asegurarse de que se cumpla el calendario.
- Resolver los riesgos que se presenten
- Producir un plan completo, preciso y exacto del plan del equipo y de cada uno de sus miembros.
- Reportar con exactitud el estado del proyecto cada semana.
- Dirigir la planeación para cada ciclo

Gestor de calidad:

Objetivo:

Proponer un plan de calidad tanto para el proceso como para el producto.

Responsabilidades principales:

- Determinar necesidades en el proceso de calidad.

- Apoyar al equipo en la definición del proceso, gestionar el plan de calidad, generar estándares para obtener un trabajo uniforme.
- Dirigir las inspecciones y revisiones de los productos generados

Administrador de requerimientos/soporte:

Objetivo:

Ayudar al equipo a conseguir las herramientas para que pueda realizar su trabajo y gestiona la configuración.

Responsabilidades principales:

- Determina, obtiene y mantiene las herramientas para realizar tareas administrativas.
- Consigue lo necesario para el desarrollo del proyecto, genera un plan de configuración para realizar su gestión.

6.14.2.5 [31]PRINCIPIOS DEL TSP

Uso de un equipo autodirigido para gestionar el conocimiento.

- El trabajo del conocimiento debe ser manejado por el equipo y las personas que realmente hacen el trabajo.
- El proceso de lanzamiento TSP crea un equipo autodirigido.
- Un plan detallado se desarrolla antes de comprometer los objetivos de la gestión y el cliente.
- La gerencia del equipo se realiza a través de reuniones semanales de TSP y presentación de informes de gestión.

[31] http://www.sei.cmu.edu/tspsymposium/past-proceedings/2009/DAY%203%20115%20PM%20Over%20Akiyama.pdf

6.14.2.6 [32]LANZAMIENTO DEL PROCESO

El TSP propone los siguientes pasos para realizar un lanzamiento:

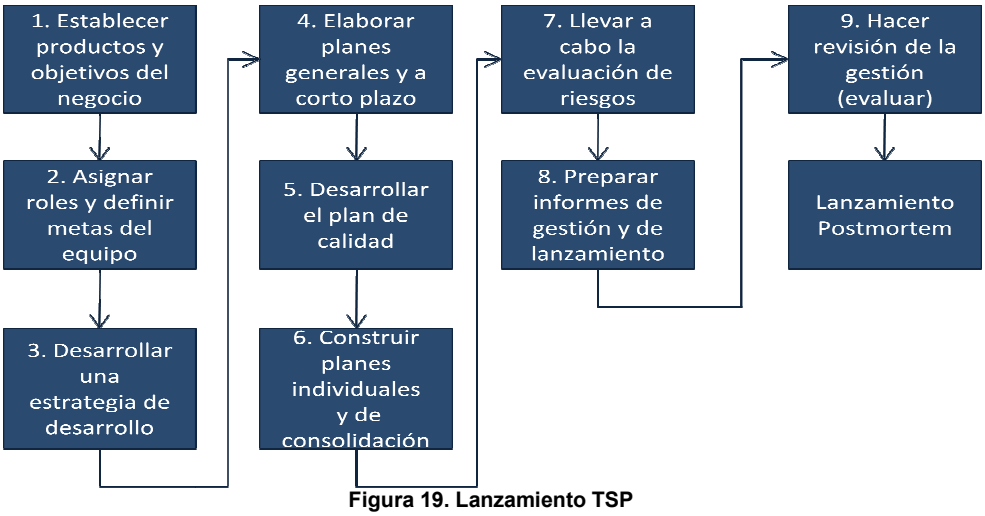


Figura 19. Lanzamiento TSP

Fuente: traducido de http://www.sei.cmu.edu/tspsymposium/past-proceedings/2009/DAY%203%20115%20PM%20Over%20Akiyama.pdf

6.14.2.7 [33]FASES DEL TSP

El TSP propone el siguiente modelo de ciclo de vida:

- Launch (lanzamiento)
- Strategy (estrategia)
- Plan
- Requirements (requerimientos)
- Design (diseño)
- Implement (implementación)
- Test (pruebas)
- Postmortem (lecciones aprendidas)

[32]Fuente: traducido de http://www.sei.cmu.edu/tspsymposium/past-proceedings/2009/DAY%203%20115%20PM%20Over%20Akiyama.pdf

[33] http://eisc.univalle.edu.co/materias/Material_Desarrollo_Software/PSP-TSP.pdf

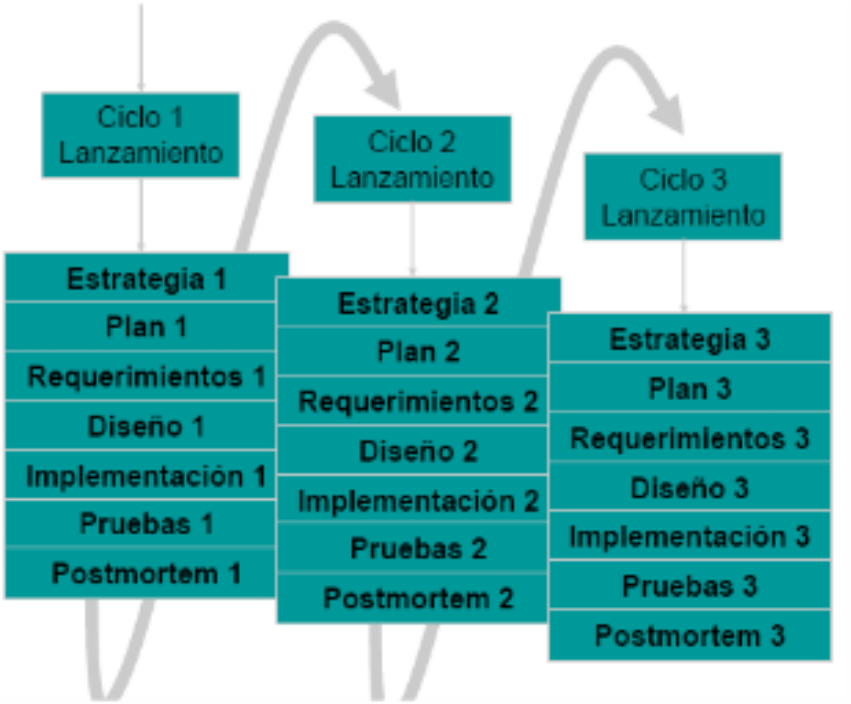


Figura 20. Fases del TSP
Fuente: http://www.slideshare.net/dlpoma/team-software-process-tsp

Lanzamiento:

- Revisión de objetivos a perseguir
- Asignación de equipos y roles al personal
- Se describen las necesidades del cliente.
- Se establece las metas individuales y del equipo.

Estrategia:

- Crear un diseño conceptual para el producto.
- Se establece la estrategia de desarrollo: se decide qué será producido en cada ciclo.
- Se hacen estimaciones iniciales de esfuerzos y tamaño.
- Se establece un plan de administración de la configuración.
- Se reutiliza el plan anterior.
- Se establecen riesgos de administración

Planeación:

- Estima el tamaño de cada artefacto a ser desarrollado.

- Se identifican las tareas: se estima el tiempo para completar cada tarea; se asignan tareas a los miembros del equipo.
- Hacer un cronograma semanal para tareas terminadas.
- Hacer un plan de calidad

Requerimientos

- Se analizan las necesidades del cliente y se entrevistan
- Se especifican los requerimientos.
- Se hace inspección de los requerimientos.
- Se diseña un plan de pruebas del sistema.

Diseño:

- Se crea un diseño de alto nivel.
- Se especifica el diseño.
- Se inspecciona el diseño.
- Se desarrolla un plan de pruebas de integración.

Implementación:

- Se usa PSP para implementar módulos y unidades.
- Se crea el diseño detallado de los módulos y unidades.
- Se revisa el diseño.
- Se convierte el diseño al código .
- Se inspecciona el código
- Se compilan y prueban los módulos y unidades.
- Se analiza la calidad de los módulos/unidades.

Pruebas:

- Se construye e integra el sistema.
- Se llevan a cabo las pruebas del sistema.
- Se produce la documentación de usuario.

Postmortem:

- Análisis de resultados.
- Se escribe el reporte del ciclo.
- Se produce producen evaluaciones de pares y equipo.

6.14.2.8 [34]ESTRUCTURA DEL TSP

Antes de que los miembros puedan participar en un equipo de TSP deben saber cómo hacer el trabajo disciplinado. Como se muestra en la figura 21, la formación en el Proceso de Software Personal (PSP) es necesaria para proporcionar a los ingenieros los conocimientos y habilidades para utilizar el TSP. La formación PSP incluye el aprendizaje de cómo hacer planes detallados, la recolección de los datos y el uso de ellos en el proceso, el desarrollo de planes de valor ganado, con valor acumulado para el seguimiento de un proyecto, medición y gestión de la calidad del producto, y la definición y el uso de los procesos operativos. Los ingenieros deben ser entrenados en estas habilidades antes de que puedan participar en la construcción del equipo de TSP o seguir el proceso definido TSP.

[34]Fuente: traducido y adaptado de http://www.sei.cmu.edu/reports/00tr023.pdf

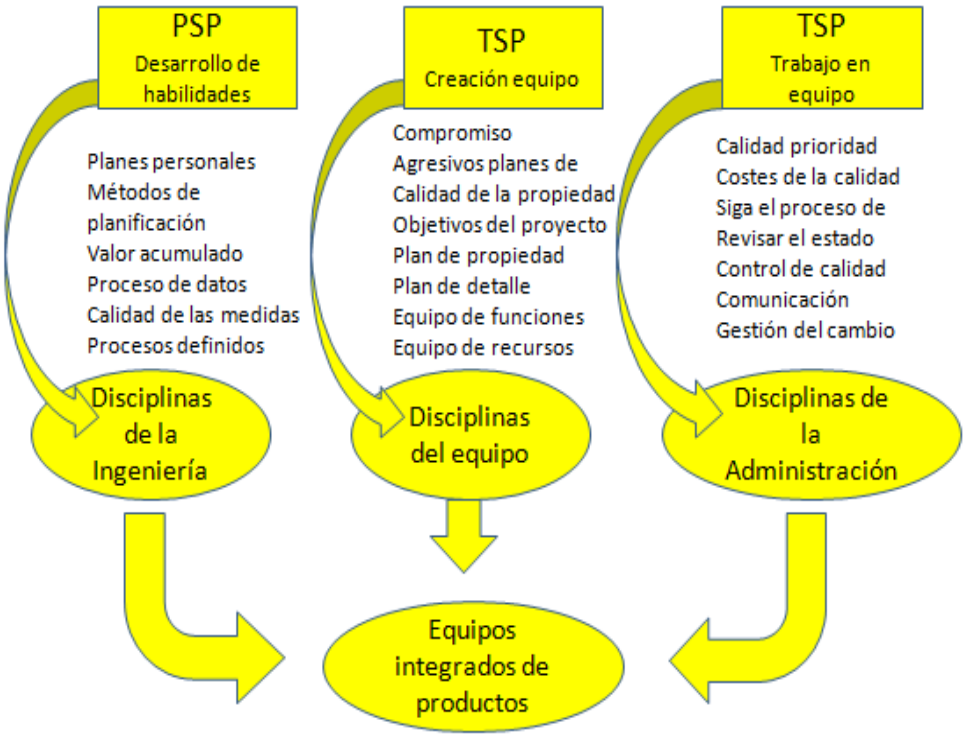


Figura 21. Estructura del TSP
Fuente: traducido de http://www.sei.cmu.edu/reports/00tr023.pdf

6.14.3 [35]PSP Y TSP COMPLEMENTO A LA ORGANIZACIÓN

TSP y PSP surgieron como metodologías que facilitan la implantación de CMMI en las organizaciones; sin embargo, estas metodologías, como se mencionó anteriormente, van dirigidas a los ingenieros de sistemas, por lo que cualquier organización que no tenga CMMI las puede adoptar.

CMMI se aplica a nivel organización, mientras que TSP y PSP son aplicados a nivel de equipos y personas. Cuando CMMI establece el qué se debería hacer, PSP y TSP dicen el cómo llegar allá, así que PSP y TSP están inmersos en el entorno organizacional

[35] Entrevistas a mexicanos expertos en PSP y TSP. Por Rafael Rincón Bermúdez. 2009. México

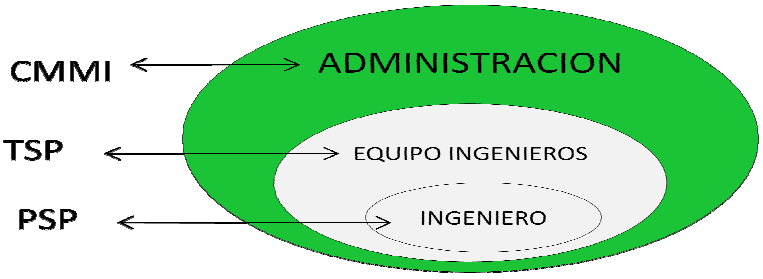


Figura 22. Entornos
Fuente: http://www.slideshare.net/dlpoma/team-software-process-tsp

PSP y TSP tienen como base algunos esquemas de software como la pirámide y el triangulo, donde su objetivo es tener sinergia para así poder realizar productos de calidad, partiendo del trabajo individual.

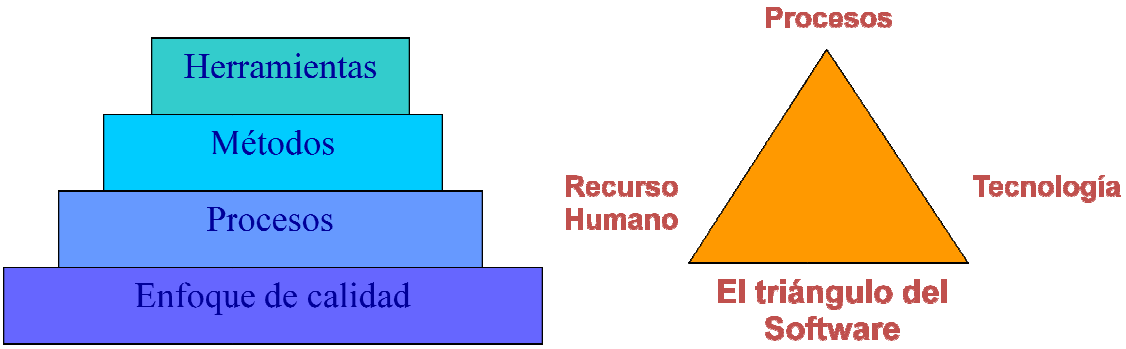


Figura 23. Esquemas del software
Fuente: Karina Cedillo Cázares. Quarksoft

Estas metodologías no van ligadas a algún tipo de tecnología, se puede aplicar con cualquier lenguaje de programación, incluso a proyectos de desarrollo con BPM cuya metodología es muy diferente a la de proyectos de programación normal. Tampoco van ligadas a algún modelo de ciclo de desarrollo, se pueden implementar por fases, o en ciclos de vida en cascada o en el que mejor se acomode a las necesidades de la organización.

PSP es la base para formar equipos TSP, puesto que cada miembro del equipo debe hacerse responsable de su rol y mostrar excelentes resultados, lo que permite el PSP, ya que una persona organizada puede trabajar bien, tanto individual como grupalmente.

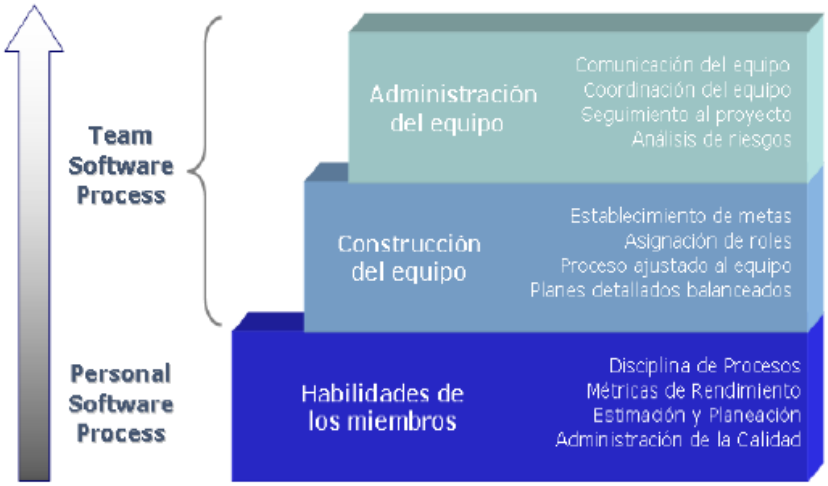


Figura 24. Escalones TSP y PSP
Fuente: http://www.kernel.com.mx/documentos/psp_tsp.pdf

6.14.4 [36]VENTAJAS

- Con PSP los desarrolladores utilizan procesos definidos y medibles. Se toma información de tamaño, tiempo y defectos al momento de realizar el trabajo. Se utilizan los datos para: planear y monitorear el trabajo, administrar la calidad de los productos que se producen y medir y mejorar el desempeño.
- TSP ha permitido resolver problemas típicos de negocio: predecibilidad de costo y tiempo, mejora de productividad y ciclos de desarrollo, mejora de calidad de productos.
- PSP/TSP mejora el desempeño tanto de equipos como individuos; es disciplinado y ágil; provee beneficios inmediatos y medibles; acelera las iniciativas de mejora de procesos organizacionales.
- Con TSP los equipos encuentran y reparan defectos en etapas tempranas del proceso de desarrollo.
- Esto reduce de manera importante el tiempo de pruebas.
- Con un testing más corto, el ciclo completo se reproduce.

[36] http://www.kernel.com.mx/documentos/psp_tsp.pdf

- Otra ventaja de estas metodologías PSP y TSP no van ligados a cierto modelo o ciclo de desarrollo, por lo que se pueden adaptar a las necesidades de la empresa.

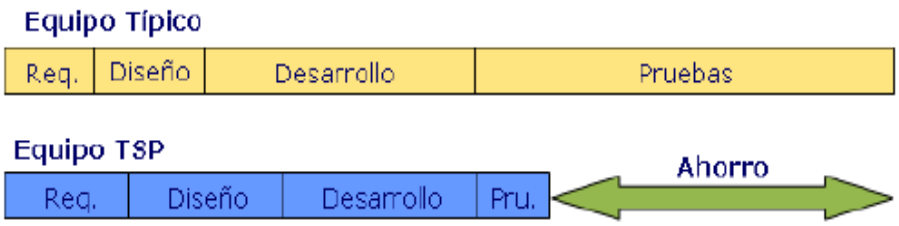


Figura 25. Comparación equipos TSP
Fuente: http://www.kernel.com.mx/documentos/psp_tsp.pdf

6.14.5 DESVENTAJAS

Las desventajas de este modelo es que es necesario que cada uno de los miembros tenga el compromiso y la disciplina de seguir el plan. Debe llenar toda la documentación requerida que incluye sus registros, planificación, las plantillas o formularios. Se debe contar con un buen conjunto de métricas y parámetros de calidad, lo cual, para algunas organizaciones, puede ser difícil de definir. Cada miembro debe estar entrenado en el PSP, si algún miembro se va, es necesario entrenar a los nuevos miembros. Algo que puede resultar una desventaja importante es que la Gerencia debe dejar trabajar a los equipos de trabajo autodirigidos de acuerdo con sus planes, algo que no muchos resisten.

Como todo modelo, para obtener los beneficios que se plantean es necesario seguirlo, y con base en los resultados que se obtengan, hacer los ajustes para la organización sin perder la esencia.

2. ENTREVISTAS Y CHARLAS CON EXPERTOS

CHARLA PSP y TSP.

Expositor: Pedro Javier Beltrán, Director Ejecutivo SILAC, Ingeniería de Software, S.A. de C.V. Guadalupe, Zacatecas, México

Lugar y fecha: Universidad Eafit, 8 al 9 de Marzo de 2010

El objetivo del evento fue la socialización de las buenas prácticas identificadas en México, para la implementación de metodologías que garantizan la calidad del software, TSP - PSP generando un impacto altamente positivo en la productividad y competitividad del clúster software en este país a niveles comparables con la India, un referente mundial en el desarrollo de esta industria.

Por tal motivo las reuniones y el panel que se llevaron a cabo en el marco del evento son fundamentales para fortalecer el trabajo en la región y evaluar cómo se pueden replicar estas buenas prácticas en Medellín.

ENTREVISTAS A EXPERTOS

Realizado por: Rafael Rincón Bermúdez, Ingeniero de sistemas

Lugar y fecha: México, septiembre - diciembre 2009

Las metodologías PSP y TSP, creadas por Watts Humphrey, son fuertemente impulsadas por el gobierno mexicano para promover los productos y servicios de tecnologías de la información, apuntando a una buena calidad tanto de los productos como de los procesos de desarrollo de software y de las personas involucradas en el mismo, con el fin de ser más competitivas a nivel internacional y sacar el mayor provecho de este mercado.

Estas metodologías no se aprenden de la noche a la mañana, requieren cursos de capacitación y se debe obtener certificación en PSP y TSP. Dichos cursos y certificaciones están abalados por el SEI (Software Engineer Intitut). La inversión para capacitarse en PSP y TSP es relativamente alta, pero los beneficios son tantos, que el retorno de la inversión se ve muy rápido. Normalmente el 50% del tiempo en un proyecto, se invierte casi siempre en

corregir defectos; con implementaciones de calidad como TSP y PSP se puede bajar ese 50% a más o menos un 20%, se reducen los tiempos, se reducen los defectos, se aumenta la calidad y la productividad, se cumplen con las fechas estimadas, entre otros beneficios que conllevan a generar ganancias. El generar productos de alta calidad y mantener al cliente satisfecho, generan una mayor rentabilidad.

“PSP ha demostrado que genera muy altos niveles de calidad. La industria de información, la industria del software es una de las pocas industrias modernas donde cualquiera hoy en día puede competir. Las herramientas que tienen para desarrollo de software cualquier compañía de la India, cualquier compañía americana, cualquier compañía en Colombia, o en México son las mismas. Todas las herramientas pueden estar a un costo accesible para construir software; por ejemplo, si se decide desarrollar procesadores de computador, eso va a significar una inversión de millones de dólares, años de inversión, años de estudio, años de tecnología y tal vez no se llegue nunca a nada; en cambio, con el software se puede llegar rápidamente a cualquier nivel que tiene cualquier otra compañía grande del mercado. Entonces PSP y TSP va a garantizar un crecimiento, garantiza que las cosas se hacen con calidad. En el software lo que se necesita son las herramientas de codificación, pero existen, y son las mismas para todo el mundo; en realidad el diferenciador es la calidad. La industria del software generalmente se ha distinguido por su mala calidad, y no solo la industria en Latinoamérica, es la industria a nivel mundial que se ha distinguido por su mala calidad. Hay un reporte que se llama el reporte CHAOS donde se ve muy claramente los resultados de calidad de la industria del software, y han cambiado desde el año 84 hoy; hay cierta mejoría pero sigue siendo una industria que genera muy malos resultados, una industria que se ha ganado merecidamente el titulo de que no entrega con calidad, no entrega a tiempo. Para cualquier compañía que contrata desarrollo de software es un dolor de cabeza. Muchas compañías quieren desarrollar software internamente, y para ellos es un dolor de cabeza, entonces deciden subcontratar, pero siguen teniendo el mismo dolor de cabeza. Hay una oportunidad muy grande ahí porque el área en general no tiene los mejores resultados y estas prácticas de PSP y TSP son prácticas que sí garantizan resultados de calidad, siempre y

cuando estén bien implementadas. TSP tiene un proceso formal, tiene una mecánica formal, tiene sugerencias de cómo se implementan las prácticas y si se hace bien la tarea, se garantiza altos niveles de calidad."[37]

Los ingenieros de sistemas de hoy en día, salen de la universidad a enfrentarse con un mundo competitivo, y las metodologías de aprendizaje deben ir cambiando para preparar gente que sea capaz de enfrentarse a ese mundo. "La academia en general, por sus raíces y la manera como tradicionalmente se ha formado, no está enfocada en la producción de calidad, normalmente tiene el enfoque del que programe más rápido y siempre el más rápido en la etapa de programación es el mejor, y normalmente eso es válido en el nivel de academia porque los proyectos que hacen son de determinado tamaño, no son proyectos que requieren equipos de trabajo, y eso como que desvía la atención."[38] Es por eso que el SEI propone empezar a implementar estas prácticas, desde las universidades lanzando una versión de PSP y TSP académica, cuya documentación se encuentra en la página del SEI http://www.sei.cmu.edu/tsp/tools/academic/.

ENTREVISTA A EXPERTO

Entrevistado: Yuri Ontibón, Director y consultor Sénior, SEONTI SEI Partner, Querétaro México.

Realizado por. Sandra Vélez

Fecha: Septiembre de 2010

El TSP y el PSP son metodologías que van orientadas a los ingenieros involucrados en el proceso de desarrollo de software, por lo que se puede implementar sin que su fin sea una certificación en CMMI.

En un equipo TSP cada integrante debe ser certificado en PSP. Un equipo puede estar integrado a partir de 3 personas, donde no hay un rol de "jefe", sino que cada uno asume su propio rol, siendo responsable de sus actividades

[37] Entrevista a Yuri Ontibón

[38] Entrevista a Yuri Ontibón

y de la comunicación con su equipo. Es un grupo autodirigido que requiere de observación más que de vigilancia.

En la empresa Colorquímica son 7 integrantes del área de desarrollo, por lo que sería perfecto poder empezar a aplicar algunas prácticas de estas metodologías y quizá en un futuro, obtener la certificación. En la página del SEI se encuentran formatos que se pueden analizar y estudiar sólo con fines académicos; no se pueden implementar a nivel empresarial sin el correspondiente permiso del SEI, pero son un buen comienzo para tener contacto con las prácticas PSP y TSP.

TERTULIA TSP

Panelista: Yuri Ontibón, Director y consultor Sénior, SEONTI SEI Partner, Querétaro México.

Lugar y fecha: Universidad Eafit, 9 de Septiembre de 2010

La administración ha venido evolucionando comenzando con la teoría de Taylor, donde la administración se enfocaba en la industrial, lo más importante era la producción y no el ser humano y sus necesidades; después aparecieron las teorías de Maslow, Mayo y Fayol, fomentaron la teoría administrativa donde las personas y sus necesidades son más importantes; de ahí surge el concepto de motivación del personal. Ahora, según Peter Drucker, quien definió el concepto de los “Trabajadores del Conocimiento”, propone que se debe administrar el conocimiento. Partiendo de este último concepto, Watts Humphrey dice: “La regla clave para administrar al trabajo del conocimiento es esta: Los gerentes no pueden administrarlo, los trabajadores deben administrarse a sí mismos”. Los conceptos de PSP y TSP están basados en los principios de Drucker y en el concepto de equipos autodirigidos.

3. PROPUESTA

Con este panorama, se propone al área de TI de la empresa Colorquímica el siguiente diagrama de proceso de desarrollo de software con su respectiva documentación.

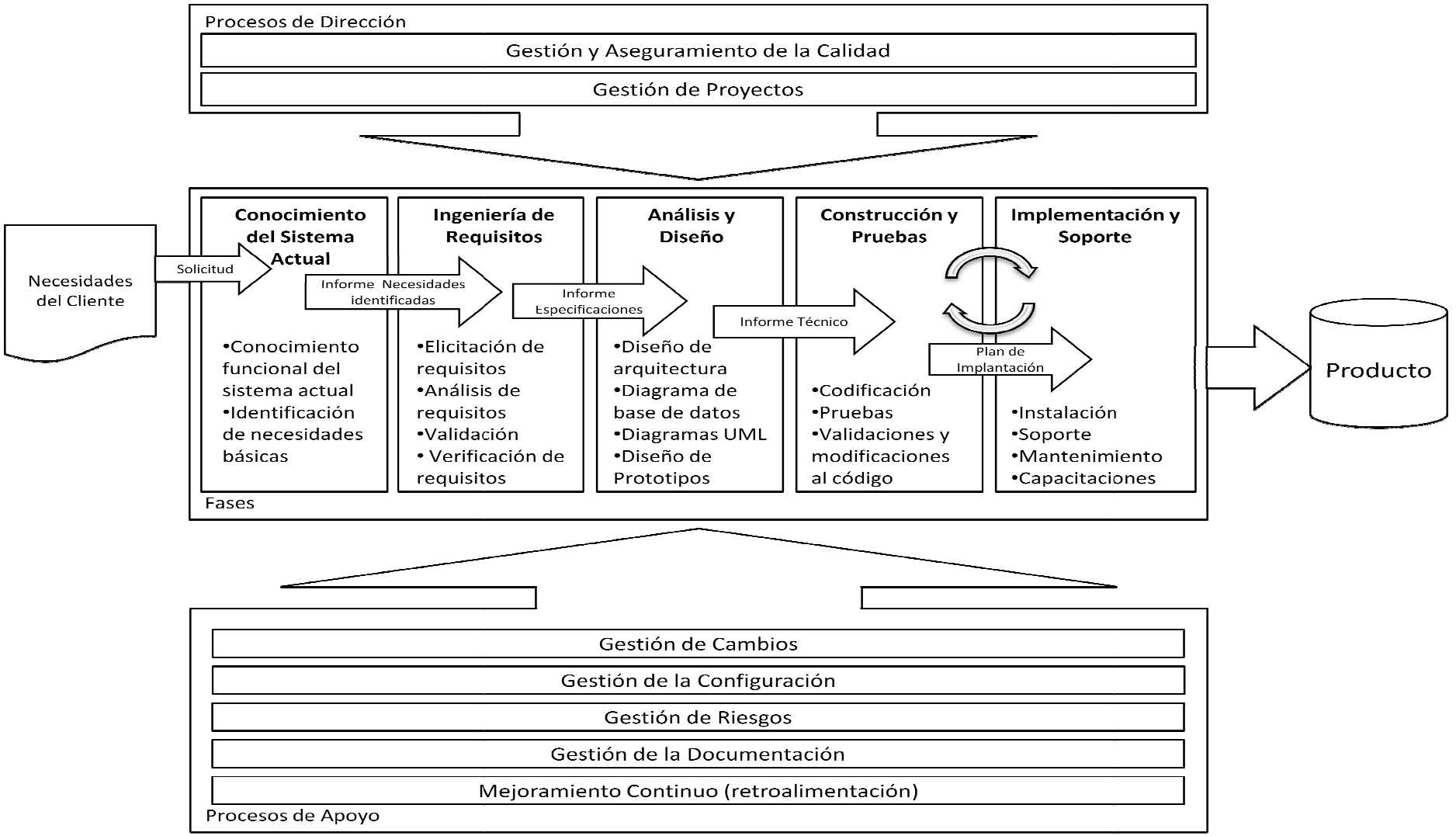


Figura 26. Proceso de desarrollo de software detallado
Fuente: Elaboración propia

PROCESO DE DESARROLLO DE SOFTWARE 2010

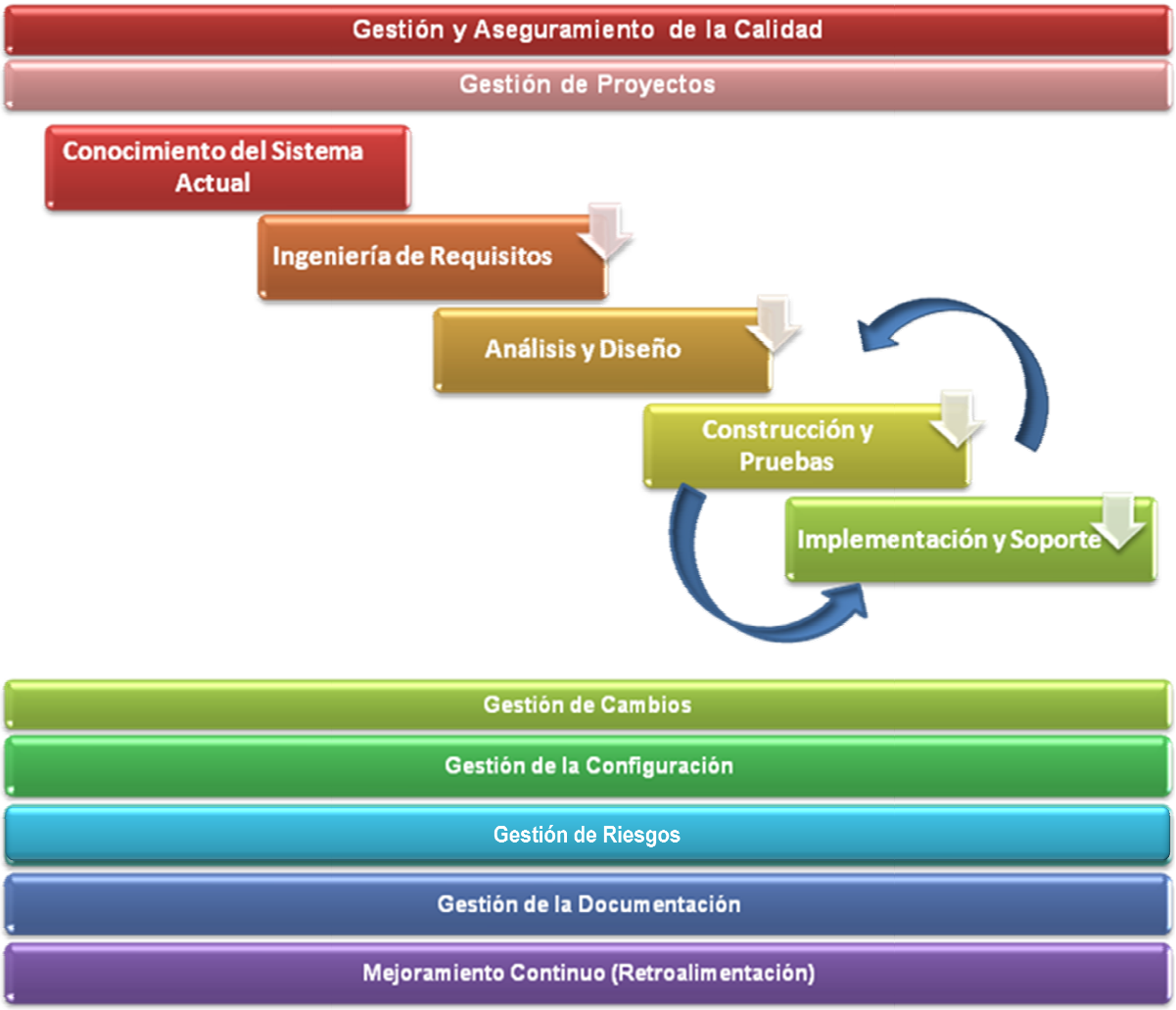


Figura 27. Proceso de desarrollo de software general
Fuente: Elaboración propia

PSP Y TSP EN EL PROCESO

Como PSP y TSP no van ligados a determinado ciclo de desarrollo, sino que tienen la ventaja de ser acomodados según las necesidades de cada empresa; se propone que, a este nuevo proceso de desarrollo se le implementen algunas prácticas de estas metodologías.

La idea para Colorquímica es poder precisar un poco más las estimaciones y disminuir los defectos. Para hacer una buena implantación de las metodologías PSP y TSP se requiere de cursos y capacitaciones, por lo que en esta propuesta se mencionarán solo algunas prácticas y se mostrarán algunos formatos para diligenciar, ya que la mayoría no son de uso público.

Las prácticas que se proponen implementar en el proceso de desarrollo de Colorquímica son, el registro de tiempos y el registro de defectos, con el fin de que cada analista sepa cuánto tiempo es productivo en realidad y cuánto tiempo de ocio utiliza, cuántos defectos inyecta al proyecto y cuántos remueve, para así, con estos históricos que se van recopilando, poder realizar estimaciones más precisas en futuros desarrollos, y según la experiencia, minimizar los defectos que se le inyectan al proyecto.

Con estas prácticas no se pretende hacer más tedioso el proceso, sino por el contrario, realizar un trabajo más productivo y con buena calidad, por lo que se necesita compromiso por parte de cada integrante del equipo de desarrollo.

Para el registro del tiempo:
Se recomienda utilizar la tabla presentada en el anexo 2. *"Plantilla 1. Registro de tiempos"*, donde se podrá llevar un registro del tiempo que se invierte produciendo y el tiempo de ocio; con la cual estimar el tiempo en un proyecto futuro. Se podrá basar en estos datos para realizar las estimaciones.

Para el registro de defectos:
Para medir y controlar los defectos, el SEI propone llevar registro de los defectos que cada ingeniero inyecta y de los defectos corregidos, para así poder saber qué tan productivo es su trabajo. "Los ingenieros necesitan datos

sobre los defectos que se inyectan, las fases en que se les inyecta, las fases en que se encuentren y arreglan, y cuánto tiempo tomó arreglarlos."[39]

"Para realizar este registro primero hay que tener claro qué es un defecto. El defecto se refiere a algo que está mal en un programa. Podría ser un error ortográfico, un error de puntuación, o una declaración del programa incorrecto. Los defectos pueden ser en los programas, en diseños, incluso en los requisitos, especificaciones, u otra documentación. Los defectos pueden encontrarse en cualquier fase del proceso y pueden ser declaraciones adicionales, declaraciones incorrectas, u omisiones de secciones del programa. Un defecto, de hecho, es todo lo que resta valor a la capacidad del programa de manera completa y efectiva para satisfacer las necesidades del usuario. Así, un defecto es una cosa objetiva. Es algo que los ingenieros pueden identificar, describir y contar."[40]

Se recomienda utilizar la tabla que se muestra en el anexo 3. *"Plantilla 2.Registro de defectos*"

DEFINICIÓN DEL PROCEDIMIENTO

PROCESO SOFTWARE COLORQUIMICA S.A.

PSC

Para la definición de este proceso, se tomó como base la definición que tiene el área de TI Colorquímica actualmente. Se sugiere esta actualización, complemento, y adaptación.

ENTRADA:

NECESIDADES DEL CLEINTE
Descripción
Un proyecto de desarrollo parte de una serie de necesidades específicas por parte del cliente o usuario con el fin de automatizar u optimizar un proceso.
Actividades
1. Hacer petición al Gerente de TI para llevar a cabo una solicitud o requerimiento de una aplicación, quien es el encargado de aprobar dicha

[39] Fuente: traducido de http://www.sei.cmu.edu/reports/00tr022.pdf
[40] http://www.sei.cmu.edu/reports/00tr022.pdf

solicitud.
2. Separar reunión con líder de proyectos para materializar dicha solicitud.
3. Comenzar el proyecto con el quipo de trabajo.

Herramientas

Word

Excel

Outlook

Entregables

Acta de solicitud de desarrollo de software firmada por el usuario, el gerente (quien aprueba) y líder de proyecto.

Información Complementaria

PROCESOS DE DIRECCIÓN:

GESTIÓN DEL PROYECTO

Propósito

El proceso de Gestión de Proyectos tiene como fin orientar, coordinar, administrar recursos para dar culminación al proyecto dentro de un alcance, tiempo y coste definidos

Actividades

1. Establecer los objetivos del proyecto
2. Planificar y establecer cronograma asignando roles a los recursos, hitos o tareas, entregables.
3. Supervisión, seguimiento y revisión del avance del proyecto.
4. Evaluación del personal.
5. Realizar plan de riesgos y su contingencia.
6. Dar la cara al cliente sobre el avance del proyecto.
7. Revisión de la documentación resultante de cada etapa del proceso de desarrollo.
8. Revisar los registros de tiempos y defectos a lo largo del proyecto basados en las plantillas 1 y 2 propuestas por PSP y TSP

Roles y Responsabilidades

Rol	Responsabilidad
Líder del proyecto	Acompañar al equipo de trabajo. Realizar las actividades mencionadas

	anteriormente.

Herramientas

Project.

Word

Excel

Outlook

Entregables

Documento con objetivos, planeación, cronograma, responsabilidades, plan de riesgos y contingencia, seguimiento, observaciones, evaluaciones al personal.

Información Complementaria

GESTIÓN Y ASEGURAMIENTO DE LA CALIDAD

Propósito

Asegurar la calidad del producto final partiendo del estudio de cumplimento de estándares y características establecidas por normas, tanto externas como internas, de cata etapa del proceso de desarrollo de software.

Actividades

1. Evaluar y verificar los entregables de cada etapa, partiendo de estándares y características definidas en cada.
2. Realizar retroalimentación en cada etapa
3. Medir la calidad en términos de tiempo y defectos utilizando el registro de datos propuesto por PSP y TSP

Roles y Responsabilidades

Rol	Responsabilidad
Líder del proyecto	Supervisar y hacer seguimiento constante del proyecto. Evaluar estándares e indicadores

Herramientas

Excel

Word

Entregables

Documento de análisis de indicadores de cada etapa del proceso de desarrollo Conclusiones y observaciones de cada etapa**.**
Información Complementaria

FASES Y PROCEDIMIENTOS

CONOCER EL SISTEMA ACTUAL
Propósito
Conocer el ambiente y procesos detallados de las actividades normales del sistema a evaluar, con el fin de entender la metodología, aplicabilidad y funcionalidad de cada procedimiento, con ello poder brindar soluciones efectivas para la optimización de los procesos. Se entiende por sistema cualquier metodología ó método utilizado recurrentemente para soportar el proceso.
Actividades
1. Identificar el formato de “Descripción del procedimiento” por parte del usuario, que contiene la información relacionada con el proceso y estudiarlo. 2. Citar reuniones con los usuarios para conocer el sistema actual. 3. Identificar y evaluar las necesidades y solicitudes presentadas por los usuarios.
Roles y Responsabilidades

Rol	Responsabilidad
Analista desarrollador	Hacer acompañamiento al usuario en su actividad para conocer la funcionalidad básica del sistema actual. Para ello se llevan bitácoras.
Usuario	Mostrar detalladamente sus funciones y actividades en el uso del sistema actual.

Herramientas
Word Excel
Entregables
Documento de Word y Excel con el contexto del problema
Información Complementaria

INGENIERÍA DE REQUISITOS
Propósito
Identificar, definir y entender las necesidades del cliente partiendo de la toma de requisitos, análisis, validación, verificación, de los mismos. Se apoyan en 8 fases nombradas a continuación
Actividades

Fase 1: Elicitación de requisitos:

Según el tipo de proyecto la toma de requisitos puede hacerse de varias maneras.

Ver anexo "Técnicas de elicitación"

Fase 2: Análisis de requisitos:

En esta fase se hace un estudio sobre los requisitos suministrados en la fase anterior para detectar conflictos y/o ambigüedades. Ver cuáles sobrepasan las necesidades del cliente y cuáles verdaderamente la suplen, esto con el fin de integrar en el nuevo sistema, una solución viable, factible y adecuada, que sea realista más que idealista.

Pasos:

a. Clasificar los requisitos
b. Analizar los requisitos que tienen conflictos, ambigüedades, requisitos que se repiten, etc.
c. Documentar los requisitos en REM

Fase 3: Validación y Verificación

En esta fase se confirma con el usuario si los requisitos recolectados y analizados son correctos, suplen la verdadera necesidad y están completos. Se responden las preguntas ¿se está construyendo el producto correcto? ¿Se está construyendo correctamente el producto?

Pasos:

a. Listar cada requisito y validarlo con el líder del proceso para su aprobación

Fase 4: Formalización de los requisitos

Pasos:

a. Realizar un documento en Word con la especificación de requisitos en lenguaje natural.
b. Entregar el documento al usuario final.

Fase 5: Aprobación de los requisitos

Pasos:

a. Firmar el documento formal de requisitos tanto por el analista como por el usuario final y el gerente de TI.

Fase 6: Gestión de cambios de requisitos

Se deben definir criterios de aceptación de requisitos, definir la negociación de los cambios de requisitos con el usuario y ser aprobados por el gerente de TI.

Fase 7. Diligenciar el registro de tiempo

Hacer uso de la Plantilla 1 “Registro de tiempos” propuesta por la metodología PSP para recopilar históricos.

Ver Anexo Plantilla 1.

Fase 8. Diligenciar el registro de defectos

Describir en la Plantilla 2. “Registro de defectos” propuesta por el PSP y TSP los defectos que se encontraron y los que se removieron en esta fase de Ingeniería de Requisitos.

Ver Anexo Plantilla 2.

Roles y Responsabilidades

Rol	Responsabilidad
Ingeniero de requisitos (Analista de desarrollo)	Aplicar todas las técnicas conocidas y estudiadas para elicitar los requisitos adecuadamente y analizarlos. Realizar el documento en REM y el documento para el usuario.
Usuario Líder	Conocer las necesidades y requisitos de las funcionalidades del proceso que se quieren automatizar.

	Entregar está información lo más detallada y completamente posible al Ingeniero de requisitos.

Herramientas

Outlook, REM, Word.

Entregables

Informe de especificaciones: REM impreso, cronograma en Project actualizado, documento de requisitos aprobado.

Información Complementaria

Nota: *En la etapa inicial del proyecto debe informarse al líder del proceso que él es el responsable de firmar el documento de requisitos al finalizar esta fase.*

ANÁLISIS Y DISEÑO

Propósito

Analiza y diseñar una solución a las necesidades del cliente.

Actividades

1. Análisis de requisitos por medio de diagramas UML
2. Diagramar y modelar los datos
3. Diseñar la BD
4. Diseñar de la arquitectura
5. Realizar diseño de interfaz y prototipos
6. Diseñar procedimientos.
7. Diligenciar el registro de tiempo: Hacer uso de la Plantilla 1 “Registro de tiempos” propuesta por la metodología PSP para recopilar históricos.
   Ver Anexo Plantilla 1.

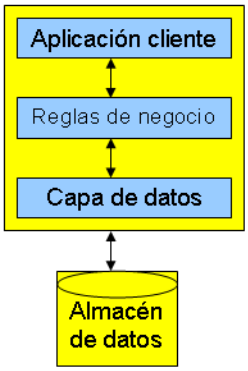
8. Diligenciar el registro de defectos: Describir en la Plantilla 2. “Registro de defectos” propuesta por el PS y TSP los defectos que se encontraron y los se removieron en esta fase de Análisis y diseño.
   Ver Anexo Plantilla 2.

Estándares Nombres Bases de Datos (Notación Pascal)	
Base de datos	*Información confidencial.*
Script versionado base de datos	*Información confidencial.*
Backup de la base de datos	*Información confidencial.*

(Ver Referencia Estándares de Programación)

Arquitectura de la aplicación

Implementaremos el modelo de 3 capas:



Estas capas estarán reflejadas en 3 carpetas dentro del proyecto con los siguientes nombres:

Información confidencial.

Roles y Responsabilidades

Rol	Responsabilidad
Analista de desarrollo	Diseñar la arquitectura que soportará el desarrollo del software, respetando los estándares de diseño definidos.

Herramientas

SQL 2005

Oracle.

Herramientas UML

Entregables

Informe técnico:

Diagrama general de la BD

Definición de la arquitectura de la aplicación

Ficha técnica de la aplicación

Prototipos

Información Complementaria

CONSTRUCCIÓN

Propósito

Definir las pautas para la construcción del software y desarrollarlo

Actividades

1. Hacer la codificación de los requisitos y diagramas siguiendo los estándares de programación vigentes.
2. Respetar la documentación definida al codificar.
3. Seguir instrucciones de gestión de la configuración para el código generado:
   La ruta donde quedarán las versiones del software que se está produciendo son:

   Información confidencial.

4. Diligenciar el registro de tiempo: Hacer uso de la Plantilla 1 “Registro de tiempos” propuesta por la metodología PSP para recopilar históricos.
   Ver Anexo Plantilla 1.

5. Diligenciar el registro de defectos: Describir en la Plantilla 2. “Registro de defectos” propuesta por el PSP y TSP los defectos que se encontraron y los que se removieron en esta fase. Los defectos en esta fase pueden ser expresados con relación a líneas de código.
   Ver Anexo Plantilla 2.

Roles y Responsabilidades

Rol	Responsabilidad
Programador (Analista de desarrollo)	Hacer la codificación del programa, siguiendo todos los estándares establecidos.

Herramientas

Visual Studio 2005,

SQL Server

PL/SQL

Documento de Requisitos de REM

Diagramas de diseño

Entregables
Plan de implantación Base de datos Instalador de la aplicación Código fuente
Información Complementaria
Este procedimiento es cíclico y se debe realizar el número de iteraciones necesarias para que el producto cumpla con una calidad aceptable, según el resultado de la etapa de pruebas

IMPLEMENTACIÓN
Propósito
Hacer de forma oficial la entrega del producto software y su documentación al usuario final. También, de esta etapa en adelante y según cronograma, se hacen capacitaciones necesarias.
Actividades
1. Adaptar la infraestructura necesaria para la instalación del producto 2. Instalación de la aplicación. 3. Difusión de la aplicación a las áreas interesadas, y en los casos específicos a toda la compañía. 4. Capacitación a los usuarios de la aplicación. 5. Entrega de Manual de usuario final, Manual de usuario administrador, Manual técnico. Todos los manuales serán compuestos por los siguiente ítems :

Manual	Descripción	Ítems
Manual de Usuario Final	Contiene la descripción paso a paso del funcionamiento de la aplicación	- Portada - Tabla de Contenido - Funciones - Glosario (En caso necesario).
Manual Técnico	Contiene la descripción de las especificaciones técnicas, requisitos y guía de instalación de la aplicación.	- Portada - Tabla de Contenido - Requisitos - Instalación - FAQ´s - Glosario (En caso necesario).

6. Diligenciar el registro de tiempo: Hacer uso de la Plantilla 1 “Registro de tiempos” propuesta por la metodología PSP para recopilar históricos.
   Ver Anexo Plantilla 1.

7. Diligenciar el registro de defectos: Describir en la Plantilla 2. “Registro de defectos” propuesta por el PSP y TSP los defectos que se encontraron y los que se removieron en esta fase. Los defectos pueden ser de instalación, o de no tener el equipo necesario para la implementación, etc.
   Ver Anexo Plantilla 2.

Roles y Responsabilidades

Rol	Responsabilidades
Líder de proyecto	Aprobación del software y la documentación. Entrega al usuario final
Usuario	Estudiar los manuales y utilizarlos como guía para operar el software.

Herramientas

Word

PDF

UPK (Software para hacer ayudas visuales, videos.)

Aplicación desarrollada.

Entregables
Manual de Usuario Final Manual Técnico Aplicación desarrollada
Información Complementaria

PROCESOS DE APOYO:

GESTIÓN DE CAMBIOS	
Propósito	
Administrar los cambios pedidos por el cliente o usuario según los requisitos elicitados, este proceso solo se hace hasta la parte de análisis y diseño.	
Actividades	
1. Estar pendiente de cambios en los requisitos por parte de los usuarios 2. Analizar viabilidad de los cambios en tiempo costo y alcance 3. Validar y Verificar cambios en requisitos 4. Formalizar los nuevos cambios 5. Pedir aprobación 6. Actualizar el documento entregable al usuario y el documento generado en la herramienta REM 7. Diligenciar el registro de tiempo: Hacer uso de la Plantilla 1 “Registro de tiempos” propuesta por la metodología PSP para recopilar históricos. Registrar cuanto tiempo lleva analizar, aprobar o rechazar un cambio. *Ver Anexo Plantilla 1.*	
Roles y Responsabilidades	
Rol	**Responsabilidades**
Analista de desarrollo	Estar pendiente de cambios en los requisitos por parte de los usuarios Analizar viabilidad de los cambios en tiempo costo y alcance Validar y Verificar cambios en requisitos Formalizar los nuevos cambios Pedir aprobación

Usuario	Brindar información necesaria para los cambios pertinentes

Herramientas

Word

REM

Entregables

Documento de cambios en requisitos en REM y en documento de usuario

Información Complementaria

Si en la etapa de construcción y pruebas se hacen cambios, no deben ser por requisitos sino por modificación al código.

Si en la etapa de Implementación y Soporte se hacen cambios en la aplicación debe ser por actualización o mantenimiento, no por requisitos iniciales.

GESTIÓN DE LA CONFIGURACIÓN

Propósito

Realizar el control de las versiones de los diferentes artefactos del proyecto. Esto con el fin de en cualquier momento conocer la línea base de un producto software y hacer control de cambios y versiones.

Actividades

El estándar para el versionamiento de los artefactos de software son:

Codificación de los Ítems del proyecto

Ítem	Nombre
Requisitos	*Información confidencial.*
Análisis	*Información confidencial.*
Código	*Información confidencial.*
Diseño	*Información confidencial.*
Entrega a Usuario	*Información confidencial.*
Entrega a Usuario	*Información confidencial.*

Cambio de Versión en el ítem de configuración Requisitos

Para versionar los requisitos se guardarán una copia en la carpeta llamada REM ubicada en cada proyecto en los siguientes casos:

Cada que se cambie el primer o segundo dígito del ítem de configuración código.

Cuando haya entrevistas con los usuarios y surjan requisitos de las mismas.

Cuando el total requisitos modificados sea igual o superior al 20% del los requisitos hechos. Cuando el analista encargado así lo crea pertinente

Cambio de Versión en el ítem de configuración Código

Para la versión en el ítem de configuración código se va a manejar 3 dígitos:

El primer dígito de derecha a izquierda va a cambiar cuando se corrijan errores en la aplicación.

El dígito del medio va a cambiar cuando se hagan cambios en la aplicación por mejoras o sugerencias.

El último dígito de derecha a izquierda va a cambiar cada que se hagan cambios grandes en la aplicación como adiciones de módulos, cambios en la arquitectura y/o cambios en la BD etc.

Cada que cambie un dígito de la versión inmediatamente quedan en ceros los dígitos que se encuentran a la derecha del mismo.

Las versiones de los ítems requisitos, análisis y diseño se va a utilizar un solo dígito.

Una de las actividades más importantes de esta área es establecer la línea base de la aplicación.

Roles y Responsabilidades

Rol	Responsabilidades
Analista de desarrollo	Controlar el cambio de versiones de los diferentes ítems. Establecer y actualizar la línea base de los proyectos que tenga a cargo.

Herramientas

Plantilla de línea base

Entregables

Ítems versionados

Nueva línea base de la aplicación
Información Complementaria

GESTION DE RIESGOS	
Propósito	
Administrar los riesgos, vulnerabilidades, amenazas que puedan afectar tanto al desarrollo del proyecto como al producto	
Actividades	
1. Identificar riesgos 2. Elaborar un plan de contingencia 3. Evaluar riesgos y hacerles seguimiento a lo largo del proyecto	
Roles y Responsabilidades	
Rol	**Responsabilidades**
Líder del proyecto Analista de desarrollo	Identificar riesgos. Sacar indicadores y porcentajes. Analizar riesgos. Hacer plan de contingencia.
Herramientas	
Word Excel	
Entregables	
Informe de riesgos identificados Informe de análisis de riesgos Plan de contingencia.	
Información Complementaria	

GESTION DE LA DOCUMENTACIÓN
Propósito
Con el avance del proyecto se va ir documentando cada fase, para que el líder

haga un adecuado seguimiento del mismo.

Actividades

1. Elaborar documentación por etapas
2. Elaborar entregables para el proceso siguiente
3. Elaborar Manuales de Usuario
4. Verificar la completitud de los documentos
5. Elaborar actas
6. Diligenciar el registro de tiempo: Hacer uso de la Plantilla 1 “Registro de tiempos” propuesta por la metodología PSP para recopilar históricos. Registrar cuánto tiempo toma realizar cada uno de los entregables.
   Ver Anexo Plantilla 1.
7. Diligenciar el registro de defectos: Describir en la Plantilla 2. “Registro de defectos” propuesta por el PSP y TSP los defectos que se encontraron y los que se removieron en cada uno de los entregables.
   Ver Anexo Plantilla 2.

Roles y Responsabilidades

Rol	Responsabilidades
Analista de desarrollo	Documentar cada fase que va realizando. Elaborar entregables para el proceso siguiente. Elaborar Manuales de Usuario.
Líder de proyecto	Revisar y colaborar con la realización de la documentación

Herramientas

Word

Entregables

Informes para cada etapa siguiente

Informes al líder de proyecto

Actas de cada proceso

Documentos aprobados

Información Complementaria

MEJORAMIENTO CONTINUO (Lecciones aprendidas)

Propósito

Evaluar el proceso en cada una de sus etapas para generar y aplicar acciones

correctivas y preventivas.

Actividades

1. Evaluar por medio de indicadores, la evolución del software en cada etapa.
2. Evaluar cada etapa y procedimiento del proceso de desarrollo de software para generar acciones correctivas o preventivas.
3. Analizar los datos recopilados en la Plantilla 1. “Registro de tiempos” para mejorar la productividad de cada ingeniero.
4. Analizar los datos recopilados en la Plantilla 2. “Registro de defectos” para mejorar la calidad de cada ingeniero.
5. Formar históricos con los datos recopilados.

Roles y Responsabilidades

Rol	Responsabilidades
Analista de desarrollo	Seguir el proceso tal y como está definido.
Líder de proyecto	Hacer seguimiento al producto desde su fase inicial Evaluar indicadores del proceso en cada etapa y procedimiento

Herramientas

Word

Excel

Entregables

Informe de mejoramiento y retroalimentación del proceso (mejoras)

Información Complementaria

Salida:

PRODUCTO
Descripción
Aplicación solución a las necesidades del cliente.

DEFINICIÓN SEGÚN PHVA:

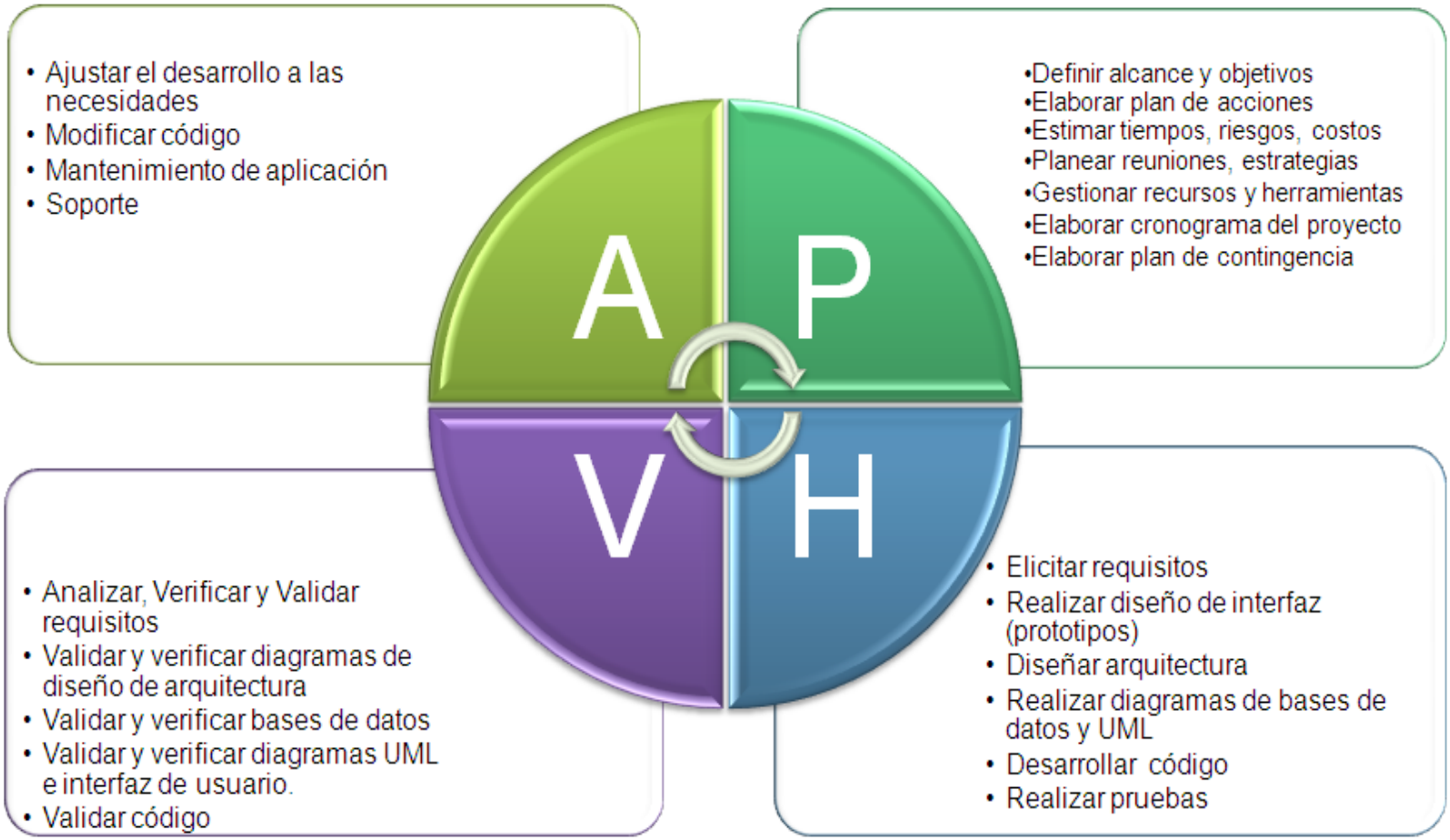


Figura 28. P-H-V-A Colorquímica.
Fuente: Elaboración propia

Definido de la siguiente manera:

ENTRADAS								SALIDAS				
INTERACCIONES			COMUNICACIONES					COMUNICACIONES		INTERACCIONES		
PROCESO	RESPONSABLE	INSUMOS	INFORMACIÓN	MEDIO		ACTIVIDAD	RESPONSABLE	INFORMACIÓN	MEDIO	PRODUCTO	RECEPTOR	PROCESO
DESARROLLO DE SOFTWARE												
Proceso del Negocio	Usuario	-	Necesidades de software.	Escrito		Elaborar cronograma del proyecto	Líder del proyecto	Necesidades del cliente, disponibilidad del equipo	Electrónico y escrito	Cronograma del proyecto	Usuario y Gerente de área	Gestión de Proyectos
						Definir alcance y objetivos del proyecto	Líder del proyecto y desarrolladores	necesidades del cliente	Electrónico y escrito	Documento formal de metas	Usuario, Gerente de área, equipo de desarrollo	Gestión de Proyectos
						Elaborar plan de acciones	Líder del proyecto y desarrolladores		Electrónico y escrito	tareas, hitos y responsabilidades	Grupo de desarrollo	Gestión de Proyectos
					P	Estimar tiempos, riesgos y costos	Líder del proyecto	disponibilidad de cada uno de ellos	Electrónico y escrito	panorama del proyecto	Grupo de desarrollo	Gestión de Proyectos
						Planear reuniones y estrategas	Líder del proyecto y desarrolladores	disponibilidad del equipo	Electrónico	Citas acordadas		Gestión de Proyectos
						Gestionar recursos y herramientas	Líder del proyecto	disponibilidad de los mismos	Electrónico			Gestión de Proyectos
						Levantamiento de requisitos	Líder del proyecto y desarrolladores	Acta de requisitos	Electrónico		Usuario y Gerente de área	Proceso del Negocio

						Levantamiento de requisitos	Líder del proyecto y desarrolladores	Acta de requisitos	Electrónico		Usuario y Gerente de área	Proceso del Negocio
					H	Desarrollo de código	desarrolladores	informe técnico y de especificaciones		Aplicación	usuario, Tester	Construcción y pruebas
						Elicitar requisitos	Ingeniero de requisitos	necesidades del cliente	Escrito	Informe de especificaciones	Usuario y analista desarrolladora	Ingeniería de requisitos
						Realizar diseño de interfaz (prototipos)	Analista desarrollador	informe de requisitos, estándar de la empresa	Escrito	Informe técnico	Analista desarrollador	Análisis y diseño
						Diseñar arquitectura de la aplicación	Analista desarrollador	informe de requisitos, estándar de la empresa	Escrito	Informe técnico	Analista desarrollador	Análisis y diseño
						Realizar diagramas de bases de datos y UML	Analista desarrollador	informe de requisitos, estándar de la empresa	Escrito	Informe técnico	Analista desarrollador	Análisis y diseño
						Realizar pruebas	Tester	Codificación		Informe de correcciones	Analista desarrollador	Construcción y pruebas
					V	Analizar, Verificar y Validar requisitos	Ingeniero de requisitos	informe de especificaciones, necesidades del usuario	Electrónico y escrito	Informe de especificaciones	Analista desarrollador	Ingeniería de requisitos
						Validar y verificar diseño de arquitectura	Analista desarrollador	informe de requisitos, estándar de la empresa	Escrito	Informe técnico	Analista desarrollador	Análisis y diseño
						Validar y verificar diagrama de bases de datos	Analista desarrollador	informe de requisitos, estándar de la empresa	Escrito	Informe técnico	Analista desarrollador	Análisis y diseño
						Validar y verificar diagramas UML y prototipos	Analista desarrollador	informe de requisitos, estándar de la empresa	Escrito	Informe técnico	Analista desarrollador	Análisis y diseño

						Validar código	Tester y desarrolladores	Codificación		Informe de correcciones	Analista desarrollador	Construcción y pruebas
						Ajustar el desarrollo a las necesidades	Analista desarrollador	informe de requisitos, estándar de la empresa	Escrito	Informe de correcciones	Analista desarrollador	Implementación y soporte
						Modificar código	Analista desarrollador	Informe de correcciones	Escrito	codificación	Analista desarrollador	Construcción y pruebas
					A	Mantenimiento de la aplicación	Analista desarrollador	informe de requisitos, estándar de la empresa	Escrito	codificación	Analista desarrollador	Implementación y soporte
						Soporte	Equipo de TI	Necesidades del usuario		capacitaciones	Analista desarrollador	Implementación y soporte

CONCLUSIONES

- Un proceso de desarrollo de software bien definido y que se siga como se debe, es fundamental para realizar productos de calidad.

- El proceso por sí solo no presenta ningún beneficio, los ingenieros deben estar comprometidos con la realización del trabajo a través de la ejecución y mejora del proceso.

- PSP es una metodología basada en estimación. La estimación permite saber cuándo y cómo se desarrollan las tareas de un proceso, por lo que podría citarse como un aspecto importante de esta metodología el estar basada en métricas y estimaciones. [41]

- Tener datos históricos ayuda a realizar estimaciones más reales y a confrontar la productividad del ingeniero.

- Puede resultar difícil adaptarse al cambio, pero ver los resultados ayuda a tomar conciencia y hacer las disciplinas de TSP y PSP parte inherente del desarrollo de software en una organización.

- PSP y TSP no van ligados a algún modelo de ciclo de desarrollo ni a algún tipo de tecnología, se puede adaptar a cualquiera según las necesidades de cada empresa.

- PSP y TSP son metodologías que promueven la calidad en el trabajo individual y grupal y los grupos autodirigidos compuestos por personas responsables y organizadas.

[41] http://html.rincondelvago.com/personal-software-process-psp.html

- La propuesta construida a través de este proyecto fue presentada en el área de TI en Colorquímica, dicha propuesta fue aceptada y será detenidamente revisada con procesos de auditoría para evaluar la viabilidad de su implementación.

- Con la implementación de esta propuesta, el área de TI podrá tener una base de proceso bien definida y estructurada, de modo que permita realizar el trabajo de mejor manera, obteniendo productos de mayor calidad, sin tener que recurrir a reprocesos, realizando entregas acorde con el tiempo estimado, sin tener que preocuparse mucho por el mantenimiento a las aplicaciones post-implementación (cambios a la aplicación después de entregada).

- El impacto que puede tener la aplicación de las prácticas expuestas en la propuesta, es positivo ya que reduce el tiempo de ocio de los ingenieros haciendo más productiva su labor, recopila datos con los cuales se pueden medir varios índices como lo son, el tiempo, el esfuerzo, los defectos de cada integrante del área de desarrollo, y la calidad tanto de los ingenieros como del producto.

- El área de desarrollo puede realizar estimaciones más precisas para proyectos futuros si aplica las prácticas propuestas de modo constante y responsable, ya que le permite realizar un historial de datos (tiempo, costo, calidad, defectos) de cada una de las fases del ciclo de desarrollo, y poder predecir con más certeza lo que ocurrirá en un futuro.

ANEXOS.

TECNICAS DE ELICITACIÓN

1. Elicitación de requisitos

1.1 Entrevistas:

Trata de reunirse con todos o los principales stakeholders (interesados en el proyecto) y a través de preguntas previamente preparadas, identificar y definir las necesidades del cliente.

Tipos de entrevistas:

Individual

Un seleccionador, un entrevistado. Es el modelo más frecuente.

Múltiple

Un o varios entrevistadores seleccionan a varios stakeholders y en forma de debate, plantear las necesidades.

Dirigida

Ceñida a un cuestionario cerrado de preguntas directas, como una encuesta. No permite profundizar.

No dirigida

Abierta, puede comenzar con una pregunta general que le permite al entrevistado responder de forma abierta.

Semidirigida

Un término intermedio y el caso más frecuente. El entrevistador parte de un esquema, pero lo modifica a partir de las respuestas y reacciones del aspirante. Combina preguntas concretas con otras de respuesta amplia.

Para llevar a cabo una buena entrevista, se recomienda seguir los siguientes pasos:

a. Estudiar el dominio del problema
b. Seleccionar a las personas que se van a entrevistar.
c. Determinar el objetivo y el contenido de la entrevista:
   Realizar las preguntas y enviarlas a los entrevistados antes de la entrevista para que conozca los temas a tratar.
d. Planificar la entrevista:
   Definir fecha, lugar, hora, duración.
e. Realizar la entrevista
f. Analizar resultados-

1.2 Lluvia de ideas (brainstorming)

"Es una técnica de reuniones en grupo cuyo objetivo es la generación de ideas en un ambiente libre de críticas o juicios [Gause y Weinberg 1989, Raghavan et al. 1994].

Las sesiones de brainstorming suelen estar formadas por un número de cuatro a diez participantes, uno de los cuales es el jefe de la sesión, encargado más de comenzar la sesión que de controlarla. Como técnica de elicitación de requisitos, el brainstorming puede ayudar a generar una gran variedad de vistas del problema y a formularlo de diferentes formas, sobre todo al comienzo del proceso de elicitación, cuando los requisitos son todavía muy difusos."[42]

Para llevar a cabo una lluvia de ideas es recomendable seguir los siguientes pasos:

[42] http://www.dsi.uclm.es/asignaturas/42541/pdf/metodologia_elicitacion.pdf

a. Preparar la lluvia de ideas

   Definir participantes, fecha, lugar.

b. Generación:

   "El jefe abre la sesión exponiendo un enunciado general del problema a tratar, que hace de semilla para que se vayan generando ideas. Los participantes aportan libremente nuevas ideas sobre el problema semilla, bien por un orden establecido por el jefe de la sesión, bien aleatoriamente. El jefe es siempre el responsable de dar la palabra a un participante. Este proceso continúa hasta que el jefe decide parar, bien porque no se están generando suficientes ideas, en cuyo caso la reunión se pospone, bien porque el número de ideas sea"[43]

c. Consolidación:

   Se revisan las ideas más importantes, más influyentes, descartar algunas, priorizar las ideas.

d. Documentación.

1.3 Casos de uso

Un caso de uso es la descripción de una secuencia de interacciones entre el sistema y uno o más actores en la que se considera al sistema como una caja negra y en la que la que los actores obtienen resultados observables. Los actores son personas u otros sistemas que interactúan con el sistema cuyos requisitos se están describiendo [Scheneider y Winters 1998].

Los casos de uso presentan ciertas ventajas sobre la descripción meramente textual de los requisitos funcionales [Firesmith 1997], ya que facilitan la elicitación de requisitos y son fácilmente comprensibles por los clientes y usuarios. Además, pueden servir

[43] http://www.dsi.uclm.es/asignaturas/42541/pdf/metodologia_elicitacion.pdf

de base a las pruebas del sistema y a la documentación para los usuarios [Weidenhaput et al. 1998].

A pesar de ser una técnica ampliamente aceptada, existen múltiples propuestas para su utilización concreta [Cockburn 1997]. En esta metodología se propone la utilización de los casos de uso como técnica tanto de elicitación como de especificación de los requisitos funcionales del sistema.

Para la descripción concreta de los casos de uso se proponen plantillas, en las que las interacciones se numeran siguiendo las propuestas de [Cockburn 1997], [Scheneider y Winters 1998] y [Coleman 1998] y se describen usando lenguaje natural en forma de patrones lingüísticos.

1.4 Prototipos

Elaboración de gráficos que muestra cómo podría quedar visualmente el producto, con funcionalidad limitada o incluso ninguna. Se desarrolla en la herramienta en la que se desarrollará el sistema.

2. PLANTILLA 1. REGISTRO DE TIEMPOS

Fecha	Hora		Tiempo de Interrupción	Tiempo	Descripción Actividad	Comentarios	C	U
	Inicio	Fin						

3. PLANTILLA 2. REGISTRO DE DEFECTOS

FASE	DEFECTOS INYECTADOS	DEFECTOS REMOVIDOS
Elicitación de requisitos		
Análisis y diseño		
Revisión de diseño		
Codificación		
Pruebas		
Implementación		
Documentación		

BUBLIOGRAFÍA

- 2010 SlideShare Inc. All rights reserved. http://www.slideshare.net/diego_aacc/psp-personal-software-process-presentation

- ICONTEC. http://www.icontec.org/BancoConocimiento/C/compendio_de_tesis_y_otros_trabajos_de_grado/compendio_de_tesis_y_otros_trabajos_de_grado.asp?CodIdioma=ESP

- REGLAMENTO DE PROYECTOS DE GRADO DE LA ESCUELA DE INGENIERÍA. http://www.eafit.edu.co/institucional/reglamentos/Paginas/reglamentos-escuela-ingenieria.aspx

- Docstoc 2010. CERTIFICADOS DE CALIDAD. http://www.docstoc.com/docs/5455974/CERTIFICADOS-DE-CALIDAD

- CONTE, Paul. Libro: Guía de supervivencia para el desarrollo de software. Pág. 5 SoftLanding Systems, Inc., Peterborough. SISRED S.A.C. , Lima – Perú

- RINCON BERMUDEZ, Rafael. 2008. “Calidad del Proceso de Software” Presentación en pdf. Pág. 11

- 2010. SEI Software Engineering Istitute http://www.sei.cmu.edu/