



TSP

(Team Software Process)

Integrantes

- Díaz Sánchez Dulce Yadira
- Maldonado Reyes Isai Michelle
- Reveles Pérez Osvaldo David
- Escamilla Camargo Alexis

Índice

Introducción.....	1
Antecedentes	2
TSP	2
Objetivos	3
Entornos.....	3
Fases del ciclo de vida	4
Estructura TSP	6
Relación PSP-TSP	6
Ventajas y desventajas	7
Ventajas.....	7
Desventajas	7
Roles en los equipos TSP	7
Conclusiones.....	9
Referencias	10

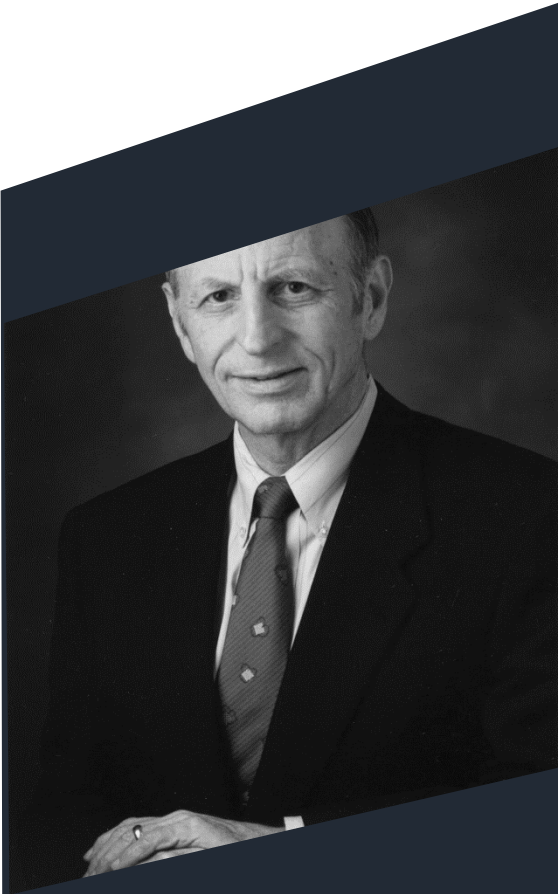
Introducción

En el mundo del desarrollo de software no es común que se adopten formas de trabajo particulares pues cada una de las personas que se desenvuelven en este ámbito adquieren conocimientos de estructuras de trabajo estandarizadas que permiten llevar un mejor control del proceso de desarrollo para crear software de calidad.

La calidad del software, se ha convertido en un requisito fundamental para los clientes y gracias a normas o estándares genéricos y procedimientos para desarrollar software esto es posible de alcanzar, cabe señalar que es importante que al utilizar estos procesos o métodos de trabajo se deben cumplir con cada una de sus etapas o de lo contrario no se origina software de calidad.

Los modelos de calidad, son herramientas que ayudan a gestionar los procesos de desarrollo para producir software de calidad, como el caso del modelo TSP (Team Software Process) que ayuda a predecir costos y tiempos de proyectos, mejorando la productividad y en general todo el ciclo de desarrollo proporcionando características de calidad.

En este documento se habla acerca de los antecedentes, la definición, los objetivos, entornos, fases del ciclo de vida, estructura, ventajas, desventajas y roles de TSP.



Watts Humphrey

(1927-2010)

Fue un conocido pensador estadounidense en el mundo de ingeniería de software, y a menudo fue llamado como el padre de la calidad de software.

Antecedentes

El proceso TSP (Team Software Process) fue desarrollado por Watt Humphrey en 1996 como una herramienta capaz de ayudar a equipos de desarrollo, así como a los ingenieros a organizar y producir proyectos de software a gran escala.

El desarrollo de esta metodología se originó debido a las limitaciones que presentaba PSP (Personal Software Process) a nivel industrial (McAndrews, 2001) ya que el PSP abarca solo las fases de desarrollo de software desde el diseño a las pruebas unitarias y permite tener control del personal mediante la mejora de las habilidades personales, en busca de la reducción de los efectos presentados en los productos y no proporciona la manera de como los ingenieros podrían aplicar estas habilidades en la práctica dentro de las organizaciones.

TSP

Esta novedosa herramienta es considerada como una metodología para administrar el trabajo de mejora y

desarrollo de los procesos de software, además de garantizar un entorno de trabajo agradable y natural para los equipos. El TSP brinda un conjunto de pasos bien estructurados que indican qué hacer en cada fase del desarrollo del proyecto y muestra cómo conectar cada fase para construir un producto completo, además brinda una ayuda acerca de cómo conformar equipos para el desarrollo de software de calidad.

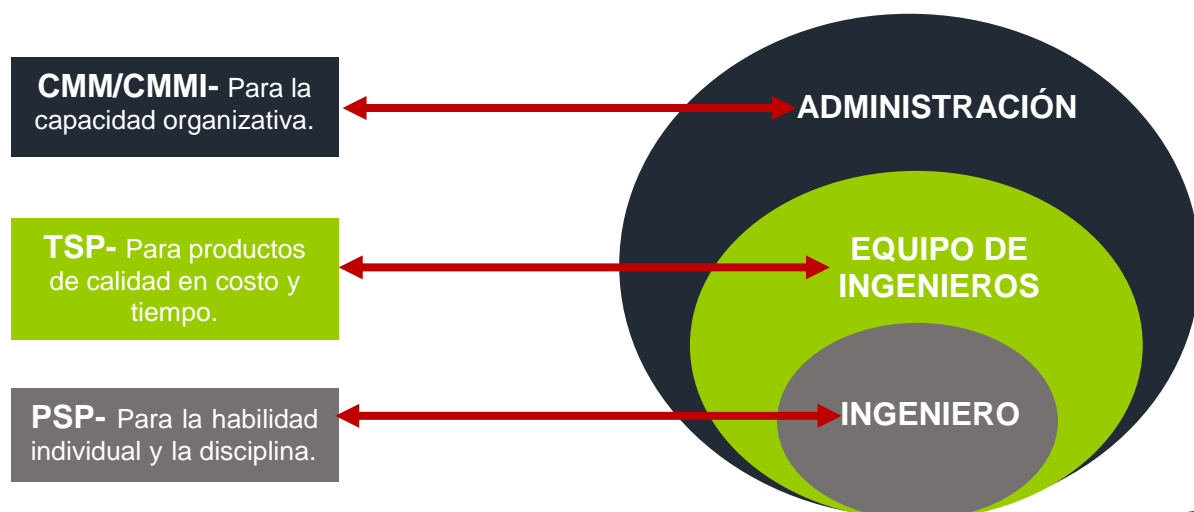
Objetivos

Los objetivos de TSP son:

- Tener equipos rápidos y confiables.
- Maximizar calidad del software, minimizar costos.
- Integrar equipos independientes de alto rendimiento que planeen y registren su trabajo, establezcan metas, y sean dueños de sus procesos y planes.
- Mostrar a los gerentes como monitorear y motivar a sus equipos de trabajo y como ayudarlos a alcanzar su máxima productividad.
- Acelerar la mejora continua de procesos.
- Proveer de una guía para el mejoramiento en organizaciones maduras.

Entornos

En el siguiente esquema se muestra cómo interactúan los modelos PSP y TSP.



Fases del ciclo de vida

1

Lanzamiento

1. Revisión de objetivos a perseguir.
2. Asignación de equipos y roles.
3. Se describen las necesidades del cliente.
4. Se establece las metas individuales y del equipo.



2

Estrategia

1. Crear diseño conceptual para el producto.
2. Se establece la estrategia de desarrollo (Se decide que será producido en cada ciclo).
3. Se hacen estimaciones iniciales de esfuerzos y tamaño.
4. Se establecen riesgos de administración.



3

Planeación

1. Estima el tamaño de cada artefacto a ser desarrollado.
2. Se identifican las tareas, (se estima el tiempo para completar cada tarea; se asignan tareas a los miembros de grupo).
3. Hacer un cronograma semanal para tareas terminadas.



4

Requerimientos

1. Se entrevista el cliente y se analizan las necesidades.
2. Se especifican los requerimientos.
3. Se hace inspección de los requerimientos.
4. Se diseña un plan de pruebas del sistema.



5

Diseño

1. Se crea un diseño de alto nivel.
2. Se especifica e inspecciona el diseño.
3. Se desarrolla un plan de pruebas de integración.



6

Implementación

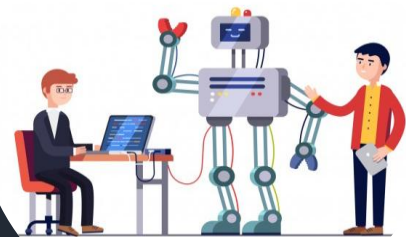
1. Se crea el diseño detallado de los módulos y unidades.
2. Se revisa el diseño.
3. Se convierte el diseño al código.
4. Se inspecciona el código.
5. Se compilan y prueban los módulos y unidades.
6. Se analiza la calidad de los módulos y unidades.



7

Pruebas

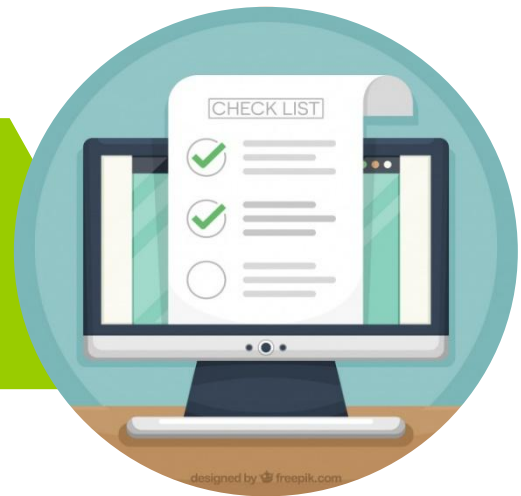
1. Se construye e integra el sistema.
2. Se llevan a cabo las pruebas al sistema.
3. Se crea la documentación para el usuario.



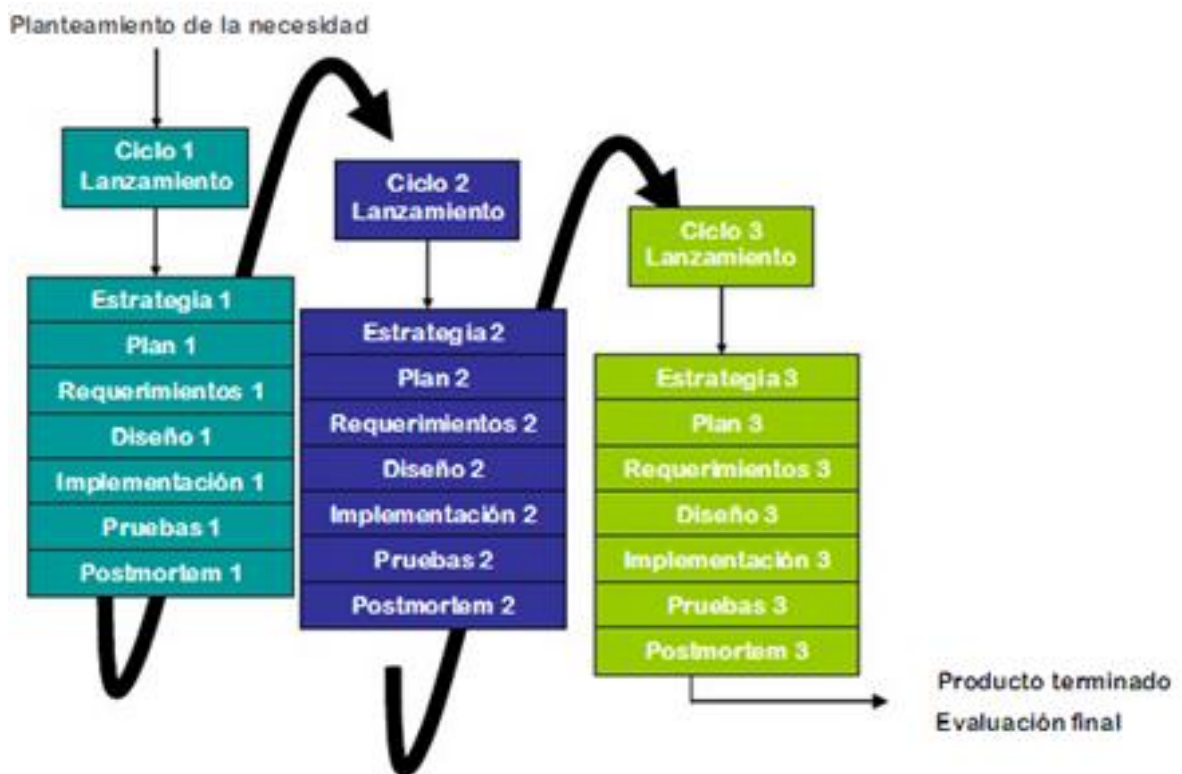
8

Postmortem

1. Análisis de resultados.
2. Se genera un reporte del ciclo.
3. Se generan evaluaciones de pares y equipo.



Estructura TSP



Relación PSP-TSP

- Ambos pueden usarse juntos
- PSP y TSP son aplicables tanto a pequeña como a gran escala
- Puede utilizarse en equipos sencillos, 5 – 15 profesionales
- Se puede aplicar en multi-equipos, en muchas docenas de profesionales

Ventajas y desventajas



Ventajas

- Detección temprana de defectos y riesgos.
- Mejora el desempeño de equipos o individuos.
- Provee beneficios inmediatos y medibles.
- Mejora hábitos en programación.
- Acelera las iniciativas de mejora de procesos organizacionales.




Desventajas

- Los miembros involucrados deben tener un compromiso y disciplina de seguir el plan.
- Toda la documentación debe de ser llenada.
- Se debe contar con un buen conjunto de métricas y parámetros.

Roles en los equipos TSP

Rol	Objetivo	Responsabilidades
<p>Líder de equipo</p> 	<p>Dirige al equipo y asegura que todos reporten sus datos de las tareas realizadas y completen su trabajo.</p>	<ul style="list-style-type: none"> • Motivar al equipo, dirigir las reuniones semanales. • Generar los informes semanales del avance del equipo. • Se encarga de la dirección y aseguramiento de las actividades realizadas por el resto del equipo
<p>Gestor de desarrollo</p> 	<p>Guiar al equipo en el diseño y desarrollo del producto.</p>	<ul style="list-style-type: none"> • Dirigir la realización de las fases de desarrollo siguiendo los estándares propuestos y generando los productos de cada fase. Integrar el trabajo de todos. • Dirección en requerimientos, diseño, desarrollo y pruebas.

TSP (Team Software Process)

<p>Gestor de planificación</p>  <p>designed by freepik.com</p>	<p>Guiar al equipo en la planificación y seguimiento del trabajo.</p>	<ul style="list-style-type: none">• Apoyar y guiar la planeación y seguimiento.• Efectuar la planificación del común acuerdo con el equipo y asegurarse de que se cumpla el calendario. Resolver los riesgos que se presenten.
<p>Gestor de Calidad/Procesos</p>  <p>PRODUCT LIST</p>	<p>Propone un plan de calidad tanto para el proceso como para el producto.</p>	<ul style="list-style-type: none">• Determina necesidades en el proceso de calidad, mantiene y sigue la calidad del producto.• Apoyar al equipo en la definición del proceso, gestionar el plan de calidad, generar estándares para obtener un trabajo uniforme, moderar las inspecciones y revisiones de los productos generados.
<p>Administrador de Requerimientos/ Soporte</p> 	<p>Ayuda al equipo a conseguir las herramientas para que pueda realizar su trabajo y gestiona la configuración.</p>	<ul style="list-style-type: none">• 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.

Conclusiones

El TSP es de gran ayuda para los ingenieros informáticos, el cual provee métodos para el fácil desarrollo de software por medio de miembros que llegan a formarse en equipos, en el cual se desenvuelven de una manera organizada; estos miembros tienen su función propia y los coordina un líder de proyecto el cual recopila información y los mantiene ordenados, para conseguir los objetivos planteados.

En lo general, un desarrollo de proyectos de software, viene a ser realizado por equipos de ingenieros, este desarrollo es, un esfuerzo colectivo que necesita conocimientos y orientación de las actividades planeadas.

Si se desea desarrollar un software, siempre es imprescindible utilizar un método como lo es el TSP, para lograr un producto confiable, organizado y de buena calidad.

Referencias

Farid, Bobadilla, F. (2013). *Blogspot*. Recuperado el 25 de septiembre del 2017 de <http://adsi6.blogspot.mx/2013/10/fases-del-ciclo-de-vida-de-tsp.html>

Formación en Ambientes Virtuales de Aprendizaje. (2015). *Senaintro*. Recuperado el 25 de septiembre del 2017 de https://senaintro.blackboard.com/bbcswebdav/institution/semillas/228106_2_VIRTUAL-2015/contenido/oaaps/oaap10/aa2/oa_calidad/oa.pdf

S/A. (2012). *GutiMar*. Recuperado el 25 de septiembre del 2017 de <https://gutimarsoluciones.wordpress.com/2012/05/09/team-software-process-tsp-y-team-process-software-tsp/>

Scalone, Fernanda. (2006). Recuperado el 25 de septiembre del 2017 de <http://laboratorios.fi.uba.ar/lsi/scalone-tesis-maestria-ingenieria-en-calidad.PDF>

Vidal Miranda, I. E. (2011). *SlideShare*. Recuperado el 25 de septiembre del 2017 de <https://es.slideshare.net/ivanvidal1/modelo-tsp>