

## **MODELOS DE CALIDAD EN EL DESARROLLO DE SOFTWARE**

---



### INTRODUCCIÓN

Los Modelos de Calidad son herramientas que guían a las Organizaciones a la Mejora Continua y la Competitividad dando les especificaciones de que tipo de requisitos deben de implementar para poder brindar productos y servicios de alto nivel. Un modelo de calidad del software es un conjunto de buenas prácticas para el ciclo de vida del software, enfocado en los procesos de gestión y desarrollo de proyectos.

Construir un modelo de calidad es bastante complejo y es usual que estos modelos descompongan las características de calidad del producto software, y que estas características se usen para determinar los ítems de una lista de comprobación de la misma.

A continuación se dan a conocer algunos de los modelos de gestión de la calidad en el desarrollo de software que se pueden implementar en cada uno de los proyectos donde participemos.

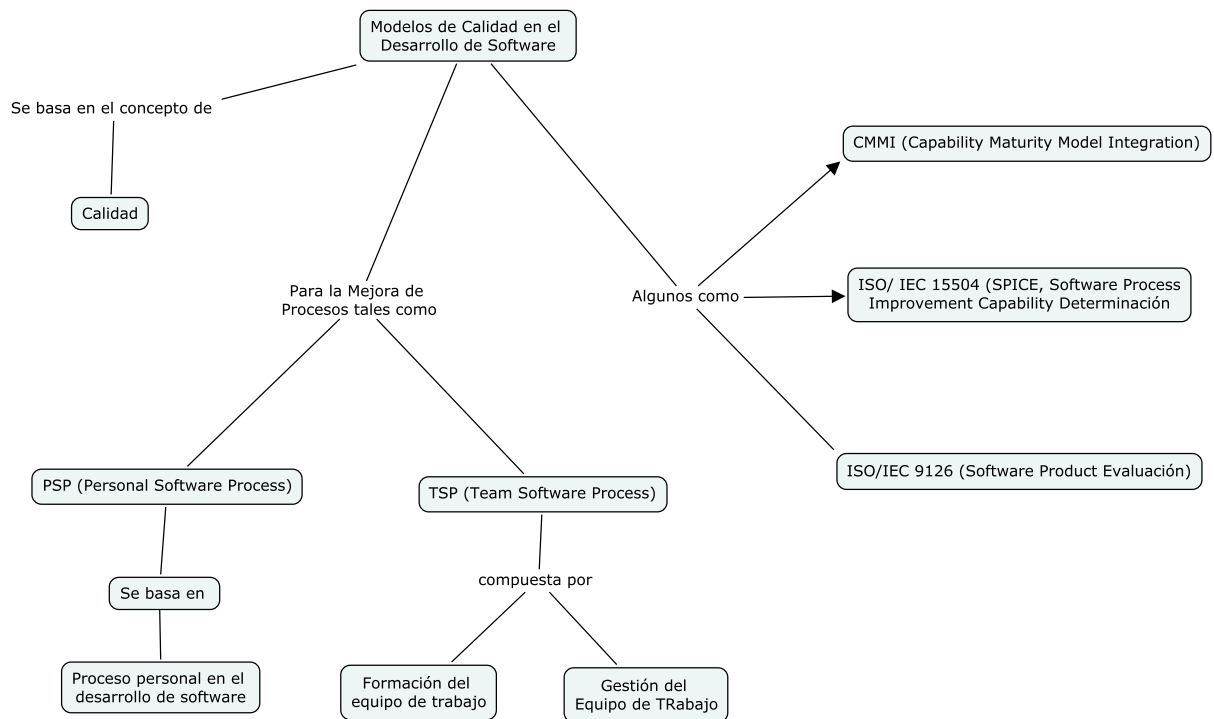


## ESTRUCTURA DE CONTENIDOS

Introducción .....	2
Mapa Conceptual .....	4
<b>1. CALIDAD DEL SOFTWARE .....</b>	<b>5</b>
<b>2. MODELOS DE CALIDAD .....</b>	<b>5</b>
2.1. Modelo CMMI .....	5
2.2. Modelo ISO/ IEC 15504 (SPICE) .....	6
2.3. Modelo ISO/IEC 9126 .....	9
<b>3. MEJORA DEL PROCESO DE SOFTWARE .....</b>	<b>14</b>
3.1. PSP .....	15
3.2. TSP .....	16
Bibliografía .....	19
Glosario .....	20
Control de Documentos .....	21
Creative Commons y Marca Registrada .....	21



### MAPA CONCEPTUAL



### 1. CALIDAD DEL SOFTWARE

Actualmente en el área de la computación la calidad de software se ha convertido en uno de los principales problemas que tiene que afrontar la Ingeniería de Software. Tanto las comercializadoras de software y los investigadores han venido realizando una gran cantidad de investigaciones sobre: ¿Cómo lograr software de calidad?, ¿Cómo evaluar el software de calidad?

Estos dos grandes interrogantes han dado una serie de respuestas, donde estas están estrechamente relacionadas. Pero primero se debe definir el término Calidad.

**Calidad de acuerdo a la Real academia Española:**

“Propiedad o conjunto de propiedades inherentes a una cosa, que permiten apreciarla como igual, mejor o peor que las restantes de su especie”



**Calidad del Software según la IEEE:** “Grado con el cual el cliente o usuario percibe que el software satisface sus expectativas”.



**Calidad del Software según la ISO:** “El conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas”.



La calidad del software es el conjunto de cualidades medibles y específicas que varía de un sistema a otro, dependiendo de tipo de software que se va a desarrollar, para determinar su utilidad y existencia. Este desarrollo debe ser confiable, mantenible y flexible para disminuir los costos de mantenimiento y perfeccionamiento durante el tiempo de utilización y durante las etapas del ciclo de vida del software.

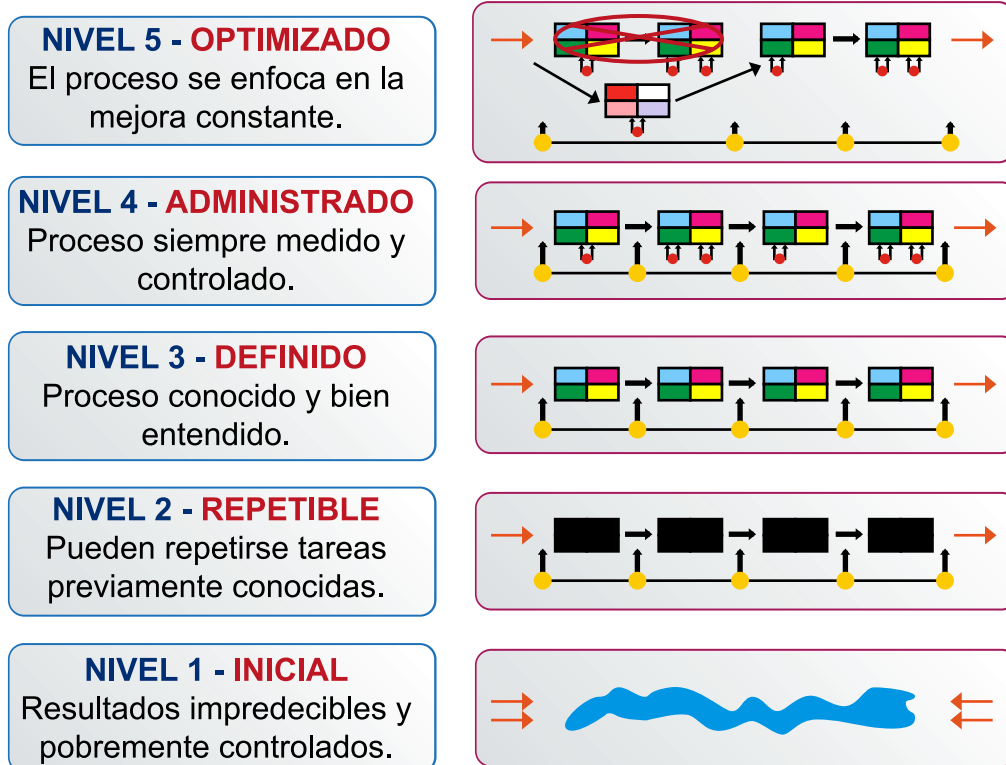
### 2. MODELOS DE CALIDAD

#### 2.1 Modelo CMMI (Capability Maturity Model Integration)

Básicamente CMMI es un conjunto de normas para calidad enfocada al mundo del Software. Estas se aplican a los diferentes procesos que hay que llevar a cabo para lograr producir software con calidad, es muy importante mencionar que igual que las normas ISO 90003, este modelo nos dice que hay que hacer, y no como hay que hacerlo.

## El modelo CMMI permite:

- Describir los componentes del modelo y sus relaciones.
- Comprender las áreas de proceso.
- Localizar información relevante en el modelo.
- Aplicar los conocimientos a su entorno de trabajo y en un equipo de evaluación de componentes y sus relaciones de un modelo.



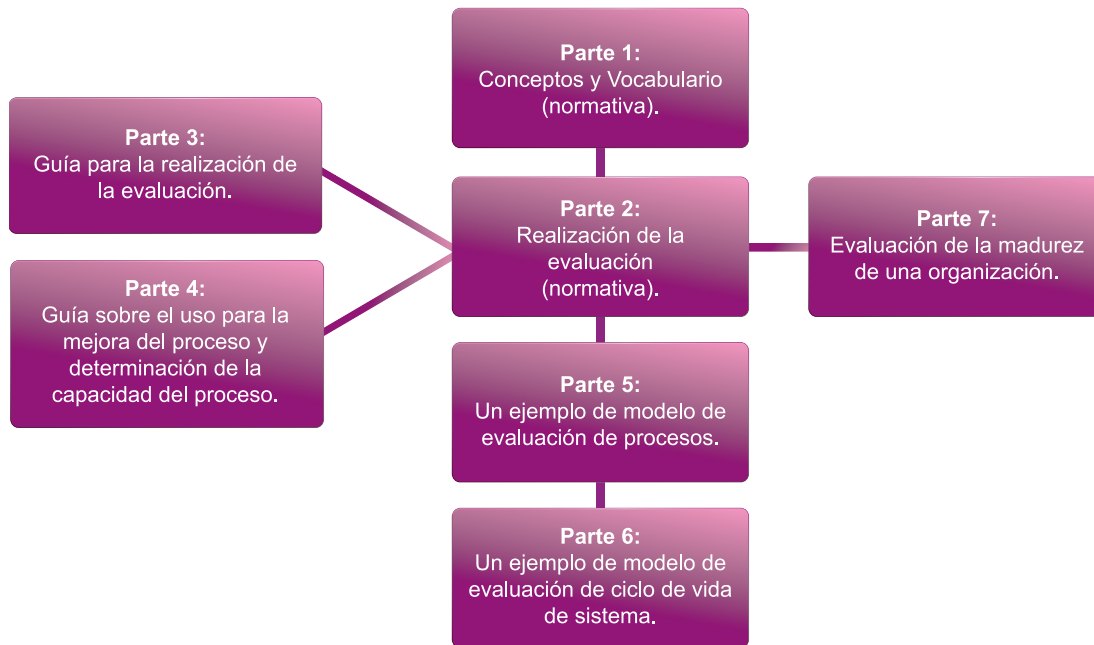
## 2.2 Modelo ISO/ IEC 15504 (SPICE)

El ISO/IEC 15504, también conocido como Software Process Improvement Capability Determinación, abreviado SPICE, en español, «Determinación de la Capacidad de Mejora del Proceso de Software» es un modelo para la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas de información y productos de software.

La norma ISO 15504 SPICE es una norma abierta e internacional para evaluar y mejorar la capacidad y madurez de los procesos. Junto con la ISO 12207, la norma aplica a la evaluación y mejora de la calidad del proceso de desarrollo y mantenimiento de software.

## Estructura de la Norma:

La norma ISO/IEC 15504 proporciona un marco de trabajo para la evaluación de los procesos y establece los requisitos mínimos para realizar una evaluación de forma consistente. Actualmente esta norma está estructurada en siete partes, la “Parte 7: evaluación de la madurez de una organización”.



Estructura de la Norma ISO/IEC 15504

La norma ISO/IEC 15504-7 define un marco de trabajo para determinar la madurez de la organización, de esta forma, se incorpora la posibilidad de evaluar a las organizaciones en ISO/IEC 15504 por niveles de madurez, dando así una “puntuación” a la organización y no sólo a nivel de proceso.

El modelo de procesos de referencia que utiliza ISO/IEC 15504-7, propio de la industria del software, es la norma ISO/IEC 12207.

La norma ISO/IEC 15504-7 establece 6 niveles de madurez para clasificar a las organizaciones, tal y como se muestra en la Figura 2.

CALIDAD EN LOS PROCESOS DEL CICLO DE VIDA DEL SOFTWARE  
Desde la identificación de requisitos, hasta la construcción,  
pruebas y soporte.

## ISO 15504 NIVELES DE MADUREZ

El nivel de madurez se obtiene según una evaluación  
del nivel de capacidad de un conjunto de procesos.

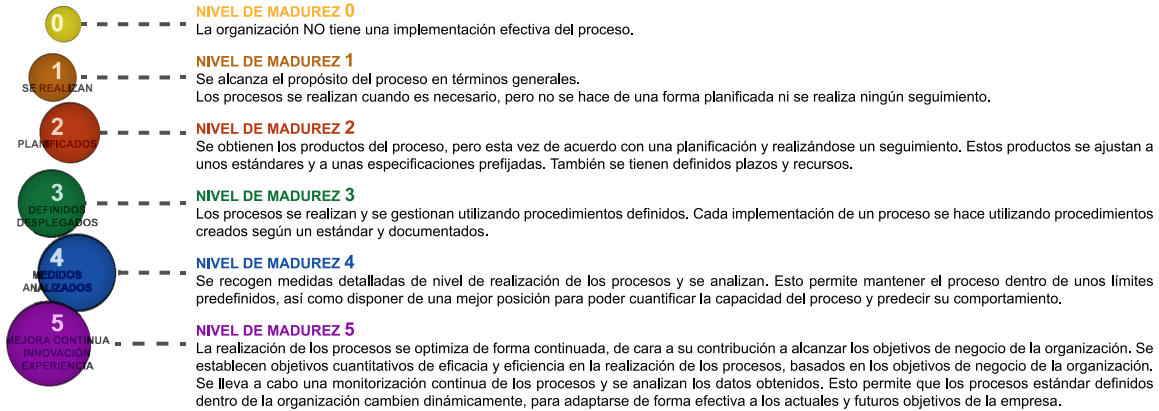


Ilustración por Toni Martín Avila  
IT360.es

Niveles de Madurez ISO/IEC 15504

### Características:

- Establece un marco y los requisitos para cualquier proceso de evaluación de procesos
- Provee requisitos para los modelos de evaluación de los procesos y para modelos de evaluación de organizaciones.
- Suministra guías para la definición de las competencias de un evaluador de procesos.
- Alcance: mejora y evaluación de procesos, y determinación de capacidad.

### El modelo de procesos ISO/IEC 15504 – ISO/IEC 12207 - 2008:

Normalmente, en la mejora de la calidad de los procesos participan dos tipos de modelos, el modelo de procesos y el modelo de evaluación. El modelo de procesos define un catálogo o colección estructurada de buenas prácticas que describen las características de un proceso efectivo, mientras que el modelo de evaluación proporciona los principios requeridos para realizar una evaluación de la calidad, e implantación, de dicho modelo de procesos en una organización.





Modelo de Procesos ISO ISO/IEC 15504

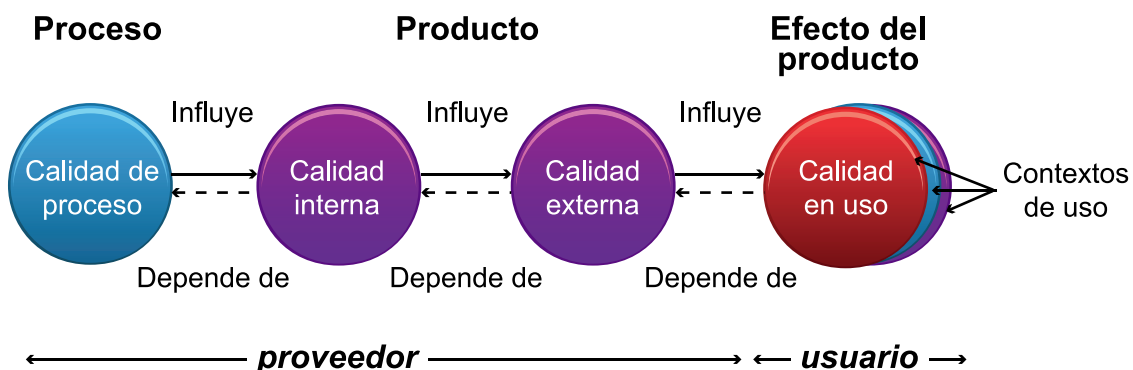
### 2.3 Modelo ISO/IEC 9126

La norma ISO/IEC 9126 Software Product Evaluación, (Evaluación de los productos de Software), indica las características de calidad y los lineamientos para su uso, la cual fue desarrollada para dar soporte a esas necesidades.

Define 6 características de calidad y describe un modelo de procesos para la evaluación de productos de software.

El estándar ISO/IEC 9126-1 define un marco conceptual que considera los siguientes factores:

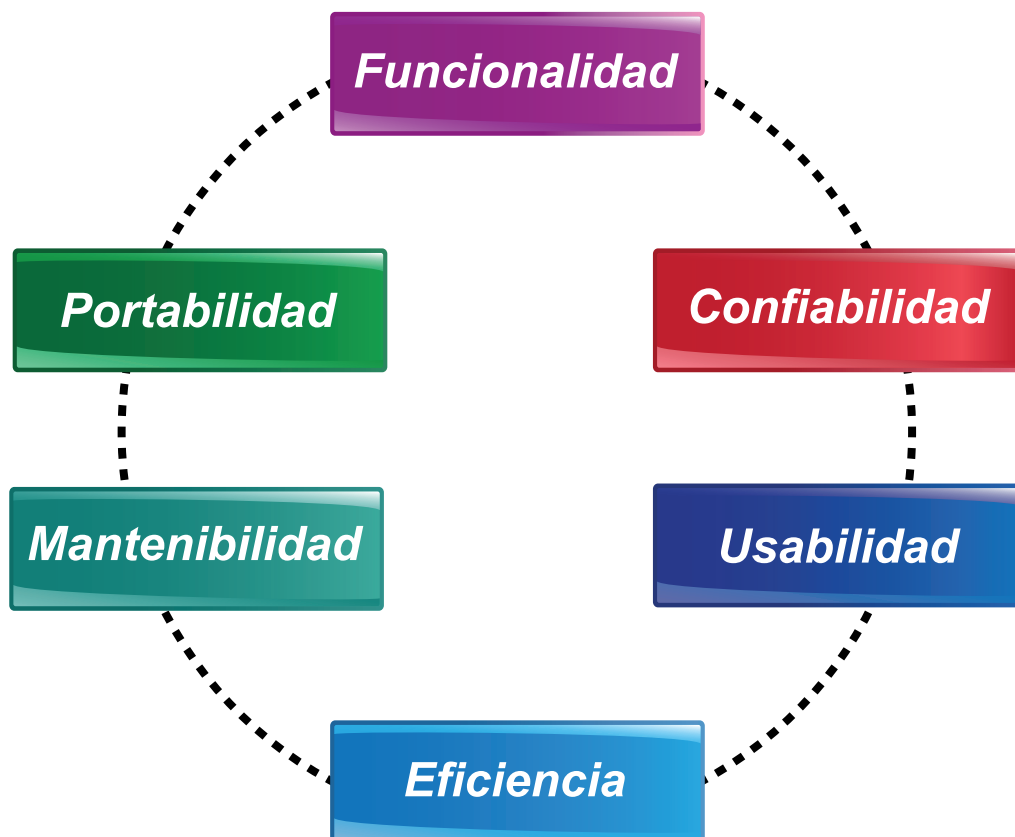
- Calidad del proceso.
- Calidad del producto software (Calidad interna y externa).
- Calidad en el uso.



**Calidad Interna:** Toma en cuenta todas las características del producto software a nivel interno, debe ser medido y evaluado tomando en cuenta los parámetros de calidad definidos, están sujetos a mejora en las etapas de pruebas e implementación sin perder la esencia definida para ello.

**Calidad Externa:** Desde una visión externa enfocadas a la ejecución del producto, se determinan las características de calidad que debe tener, es evaluada en un ambiente simulado y con métricas de la industria. Se espera eliminar la mayor cantidad de fallas posibles durante las pruebas, sin afectar la arquitectura del software.

### Atributos de la Norma para Calidad Externa e Interna:



### FUNCIONALIDAD

#### Adecuación:

Capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.

#### Exactitud:

Capacidad del producto software para proporcionar los resultados o efectos correctos o acordados, con el grado necesario de precisión.

#### Interoperabilidad:

Capacidad del producto software para interactuar con uno o más sistemas especificados.

#### Seguridad de acceso:

Capacidad del producto software para proteger información y datos de manera que las personas o sistemas no autorizados no puedan leerlos o modificarlos, al tiempo que no se deniega el acceso a las personas o sistemas autorizados.

#### Cumplimiento funcional:

Capacidad del producto software para adherirse a normas, convenciones o regulaciones en leyes y prescripciones similares relacionadas con funcionalidad.

### CONFIABILIDAD (FIABILIDAD)

#### Madurez:

Capacidad del producto software para evitar fallar como resultado de fallos en el software.

#### Tolerancia a fallos:

Capacidad del software para mantener un nivel especificado de prestaciones en caso de fallos software o de infringir sus interfaces especificados.

#### Capacidad de recuperación:

Capacidad del producto software para restablecer un nivel de prestaciones especificado y de recuperar los datos directamente afectados en caso de fallo.

#### Cumplimiento de la fiabilidad:

Capacidad del producto software para adherirse a normas, convenciones o regulaciones relacionadas con la fiabilidad.

### FACTIBILIDAD DE USO (USABILIDAD)

#### Capacidad para ser entendido:

Capacidad del producto software que permite al usuario entender si el software es adecuado y cómo puede ser usado para unas tareas o condiciones de uso particulares.

#### Capacidad para ser aprendido:

Capacidad del producto software que permite al usuario aprender sobre su aplicación.

#### Capacidad para ser operado:

Capacidad del producto software que permite al usuario operarlo y controlarlo

#### Capacidad de atracción:

Capacidad del producto software para ser atractivo al usuario.

#### Cumplimiento de la usabilidad:

Capacidad del producto software para adherirse a normas, convenciones, guías de estilo o regulaciones relacionadas con la usabilidad.

### ATRIBUTOS PARA CALIDAD EN USO

#### Efectividad:

Capacidad del producto software para permitir a los usuarios alcanzar objetivos especificados con exactitud y completitud, en un contexto de uso especificado.

#### Productividad:

Capacidad del producto software para permitir a los usuarios gastar una cantidad adecuada de recursos con relación a la efectividad alcanzada, en un contexto de uso especificado.

#### Seguridad física:

Capacidad del producto software para alcanzar niveles aceptables del riesgo de hacer daño a personas, al negocio, al software, a las propiedades o al medio ambiente en un contexto de uso especificado.

#### Satisfacción:

Capacidad del producto software para satisfacer a los usuarios en un contexto de uso especificado.

### EFICIENCIA

#### Comportamiento temporal:

Capacidad del producto software para proporcionar tiempos de respuesta, tiempos de proceso y potencia apropiados, bajo condiciones determinadas.

#### Utilización de recursos:

Capacidad del producto software para usar las cantidades y tipos de recursos adecuados cuando el software lleva a cabo su función bajo condiciones determinadas.

#### Cumplimiento de la eficiencia:

Capacidad del producto software para adherirse a normas o convenciones relacionadas con la eficiencia.

### MANTENIBILIDAD

#### Capacidad para ser analizado:

Es la capacidad del producto software para serle diagnosticadas deficiencias o causas de los fallos en el software, o para identificar las partes que han de ser modificadas.

#### Capacidad para ser cambiado:

Capacidad del producto software que permite que una determinada modificación sea implementada.

#### Estabilidad:

Capacidad del producto software para evitar efectos inesperados debidos a modificaciones del software.

#### Capacidad para ser probado:

Capacidad del producto software que permite que el software modificado sea validado.

#### Cumplimiento de la mantenibilidad:

Capacidad del producto software para adherirse a normas o convenciones relacionadas con la mantenibilidad.

### PORTABILIDAD

#### Adaptabilidad:

Capacidad del producto software para ser adaptado a diferentes entornos especificados, sin aplicar acciones o mecanismos distintos de aquellos proporcionados para este propósito por el propio software considerado..

#### Instalabilidad:

Capacidad del producto software para ser instalado en un entorno especificado.

#### Coexistencia:

Capacidad del producto software para coexistir con otro software independiente, en un entorno común, compartiendo recursos comunes.

#### Capacidad para reemplazar:

Capacidad del producto software para ser usado en lugar de otro producto software, para el mismo propósito, en el mismo entorno.

#### Cumplimiento de la portabilidad:

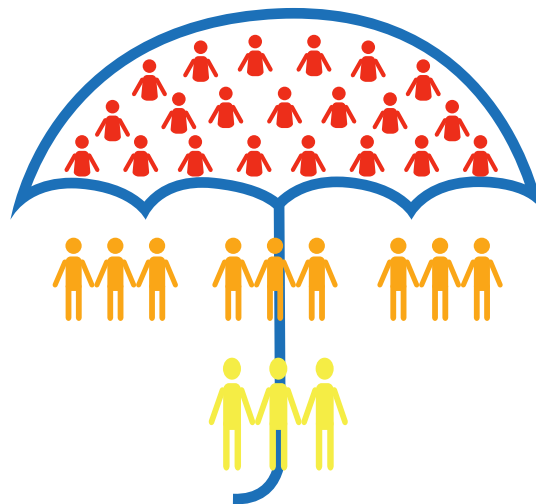
Capacidad del producto software para adherirse a normas o convenciones relacionadas con la portabilidad

## 3. MEJORA DEL PROCESO DE SOFTWARE

CMM/CMMI - para capacidad organizacional.

TPS - para productos con calidad en costo y calendario.

PSP- para habilidad y disciplina individual.



### 3.1 PSP (Process software Personal)

En el año de 1995 el PSP fue propuesto por Watts Humphrey, este inicialmente estaba dirigido para estudiantes, para 1997 con el lanzamiento del libro "An Introduction to the Personal Software Process" el PSP ya estaba destinado a los ingenieros.

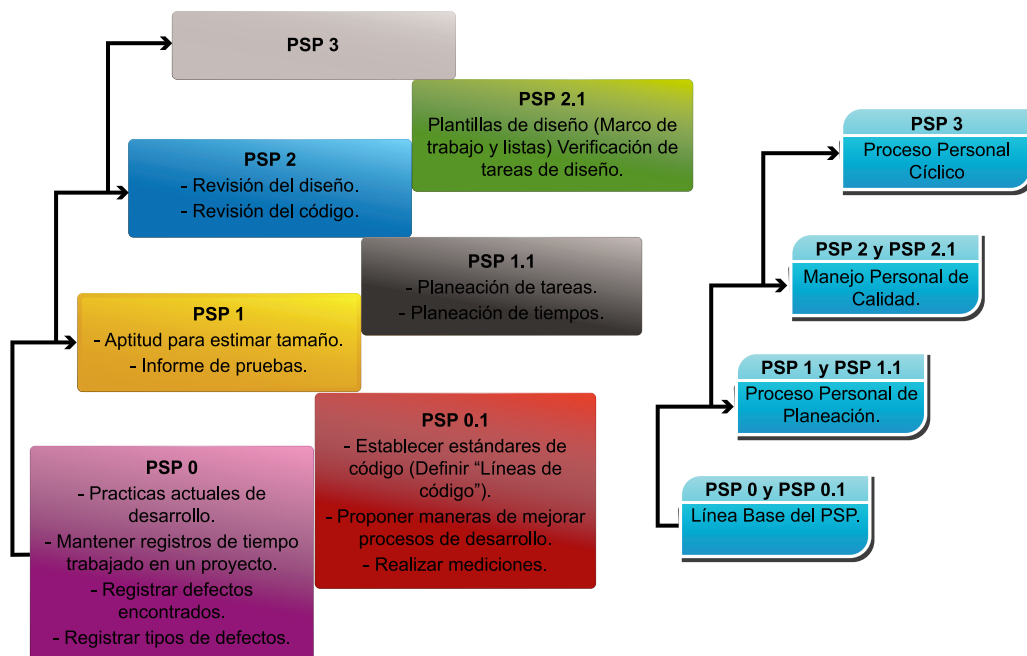
#### Características:

- El PSP busca proporcionar un marco de trabajo para el personal involucrado en el proceso de desarrollo de software.
- El PSP se centra en la administración del tiempo y en la administración de la calidad a través de la eliminación temprana de defectos.
- PSP demuestra cómo manejar la calidad desde el principio del trabajo.

#### Objetivos del PSP:

- Planificar, estimar, medir, seguir y controlar todo el proceso de desarrollo de software.
- Lograr una disciplina de mejora continua en el proceso de desarrollo.
- Mejorar los niveles de calidad del proceso de desarrollo de software
- En general, PSP provee calidad y productividad a todo el proceso de desarrollo de software.

#### Estructura PSP por Niveles:



### **PSP puede ser aplicado en:**

- Desarrollo de programas.
- Definición de requerimientos.
- Documentación.
- Pruebas de sistemas.
- Mantenimiento de sistemas.

### **Los principios de planeación y de calidad en PSP son:**

- Cada ingeniero es diferente, para ser más eficiente, debe planificar su trabajo basándose en su experiencia personal.
- Para mejorar constantemente su desempeño, los ingenieros deben utilizar personalmente procesos bien definidos y medidos.
- Los ingenieros deben asumir la responsabilidad personal de la calidad de sus productos.
- Cuanto antes se detecten y corrijan los errores menos esfuerzo será necesario
- Es mas efectivo evitar los defectos que detectarlos y corregirlos.
- La manera correcta de hacer las cosas es siempre la manera más rápida y más económica de hacer un trabajo.

## **3.2 TSP (Team Software Process)**

TSP fue creado en 1999 por Watts Humphrey con el fin de proporcionarles a los estudiantes de ingeniería de software una visión total del ciclo de vida del software

TSP busca a través de la definición de procesos, ayudar a predecir costos y tiempos de proyectos, mejorando la productividad y en general todo el ciclo de desarrollo proporcionando características de calidad.

### **Características:**

- Se fundamenta en el trabajo en equipo al establecer directrices para la definición de objetivos, planificación y control de sus procesos, enfocados en prácticas de ingeniería avanzada. Se establecen actividades de apoyo para la conformación de equipos de trabajo y su administración.



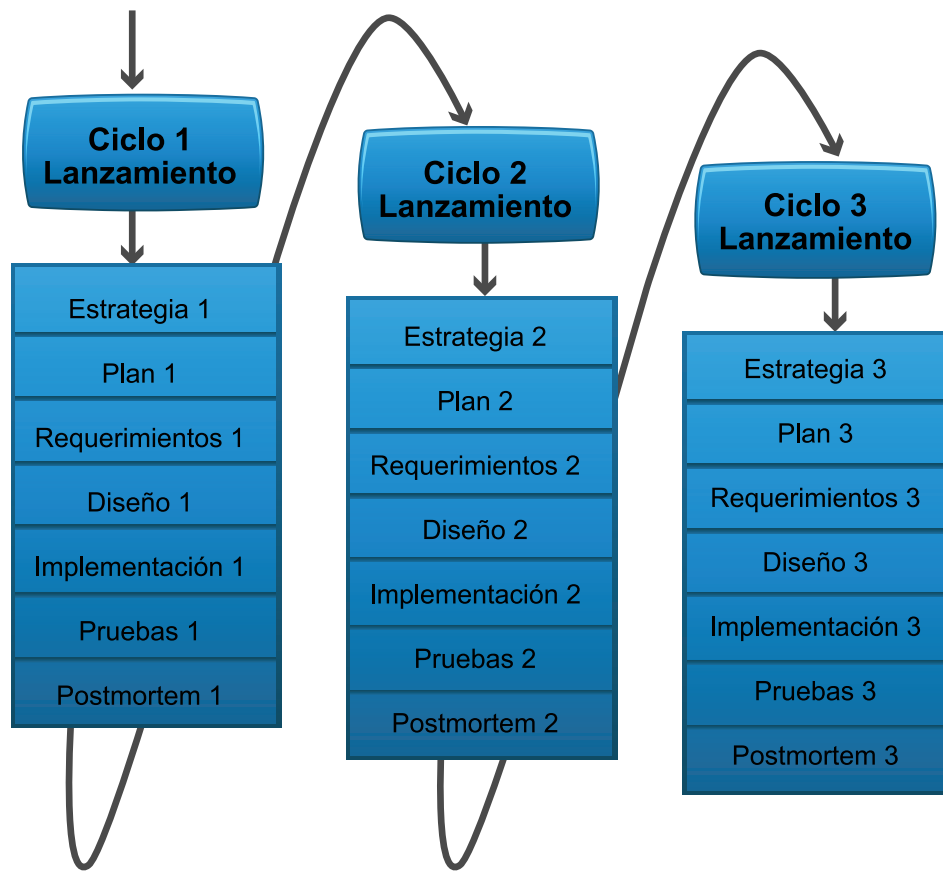


- Se posibilita encontrar defectos en las etapas iniciales del ciclo de desarrollo del software, al ser reparados tempranamente, se reduce el tiempo de pruebas y por tanto el tiempo total.
- La aplicación de esta metodología ayuda a mejorar el desempeño de los individuos y por tanto de los equipos, haciéndolos más disciplinados y ágiles.

### Principios del TSP:

- Es necesario incorporar a los técnicos en las actividades de planeación, lo cual ayuda a obtener planes más detallados e incentiva su compromiso con el mismo.
- La carga de trabajo del equipo debe estar equilibrada para mejorar los tiempos requeridos en el proyecto.
- Debe primar el interés por mantener los atributos de calidad definidos para el producto y así aumentar la productividad.

### Etapas TSP:



### Los Roles (responsabilidades) en los equipos en TSP son:

**Líder del Equipo:**

Dirige al equipo, se asegura que todos reporten sus datos de los procesos y completen su trabajo tal y como se planeó. Realiza los reportes semanales del avance del equipo.

**Gestor de desarrollo:**

Guía al equipo en el diseño y desarrollo del producto.

**Gestor de Planificación:**

Apoya y guía al equipo en la planificación y seguimiento del trabajo.

**Gestor de Calidad/Proceso:**

Apoya al equipo en definir sus necesidades acerca del proceso y a establecer y administrar el plan de calidad. Genera estándares para obtener un trabajo uniforme. Modera las inspecciones y revisa cada artefacto generado.

**Administrador de Requerimientos/Soporte:**

Dirige al equipo en el desarrollo de requerimientos de software y ayuda a dar a conocer la tecnología y en las necesidades de apoyo administrativo. Administra el plan de configuración

### BIBLIOGRAFÍA

Calero, C. y otros (2010). Calidad del producto y proceso software. Editorial Ra-ma

ISO-IEC (2009). Norma ISO 9126: "Information technology-Software product evaluation-Quality characteristics and guidelines for their use". Disponible en: <http://www.issco.unige.ch/en/research/projects/ewg96/node13.html>

PortalISO25000 (2013). Norma 14598. Consultada en diciembre de 2013 y disponible en: <http://iso25000.com/>

ISO-IEC. ISO/IEC 15504, Modelo Spice. Disponible en <http://software-ufps.blogspot.com/search/label/CLASE%2014%3A%20Modelo%20de%20Calidad%20SPICE>

Oré Bravo, Alexander. (2008). Introducción al Modelo CMMI-CMM. Disponible en [http://www.calidadyssoftware.com/otros/introduccion\\_cmmi.php](http://www.calidadyssoftware.com/otros/introduccion_cmmi.php)

Software Engineering Institute (2013). The Personal Software Process (PSP) Body of Knowledge, Version 2.0. disponible en: <http://resources.sei.cmu.edu/library/asset-view.cfm?assetID=8907>

Villalta, Carlos. (2009) Ejemplo de Aplicación de PSP. Disponible en: <http://proceso-software-personal.blogspot.com/>



### GLOSARIO

**Calidad:** Propiedad o conjunto de propiedades inherentes a una cosa, que permiten apreciarla como igual, mejor o peor que las restantes de su especie.

**CMMI Capability Maturity Model Integration):** Es un modelo de calidad del software que clasifica las empresas en niveles de madurez.

**Defectos:** Es el resultado de un fallo o deficiencia durante el proceso de creación del software, donde los más representativos son en la fase de desarrollo.

**ISO:** Organización internacional para la estandarización, que regula una serie de normas para fabricación, comercio, y comunicación, en todas las ramas industriales

**Mejora Continua:** Acción recurrente que aumenta la capacidad para cumplir los requisitos

**Mejora de Calidad:** Parte de la gestión de la calidad enfocada a aumentar la capacidad de cumplir con los requisitos de la calidad.

**Modelo de Calidad:** Conjunto de buenas prácticas para el ciclo de vida del software, enfocado en los procesos de gestión y desarrollo de proyectos.

**Proceso:** Conjunto definido de pasos para hacer un trabajo.

**Proceso definido:** Cuando un proceso está totalmente descrito.

**Productividad desarrollo de software:** Número de líneas de código fuente producidas por persona en unidad de tiempo.

**PSP (Personal Software Process):** Es un marco de trabajo para la gestión del tiempo y la productividad personal en los proyectos de desarrollo de software.

**Rol:** El papel de un individuo o grupo para desarrollar una actividad determinada.

**TSP (Team Software Process):** Es 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.



<b>Control de documento</b> <b>Construcción Objeto de Aprendizaje</b> <b>Modelos de Calidad en el Desarrollo de Software</b>	
Desarrollador de contenido Experto temático	César Marino Cuéllar Chacón
Asesor pedagógico	Rafael Neftali Lizcano Reyes
Producción Multimedia	Julio Cesar Orduz Tarazona
Programador	Francisco José Lizcano Reyes
Líder expertos temáticos	Ana Yaqueline Chavarro Parra
Líder línea de producción	Santiago Lozada Garcés

