



CONTROL DE CALIDAD DE SOFTWARE Y SISTEMAS

Prefacio:

La asignatura es de carácter teórico-práctico. Ésta, tiene como fin desarrollar en el estudiante habilidades para el control de calidad de software y sistemas informáticos. Además, le brinda los conocimientos necesarios para reconocer, aplicar y analizar los diferentes modelos de evaluación y control de software de manera que el discente pueda desempeñarse en su entorno social y profesional con propiedad.



La calidad de software es todo el conjunto de cualidades que lo caracterizan determinando su eficiencia y utilidad, satisfaciendo las necesidades tanto implícitas como explícitas del cliente. La IEEE.Std.610-1990 la define como el grado con el que un sistema, componente o proceso cumple con los requisitos especificados y las necesidades o expectativas del cliente o usuario. [IEEE.Std.610-1990].

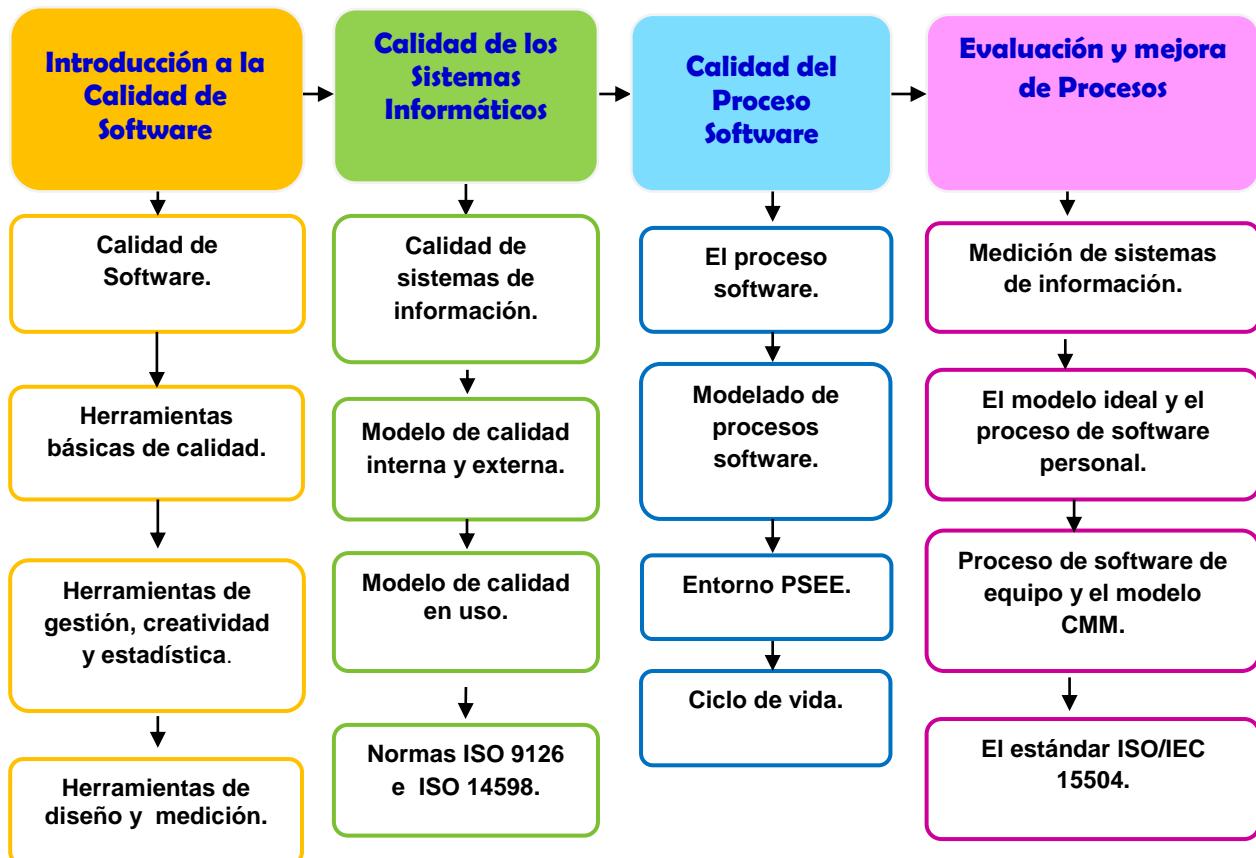
Comprende cuatro Unidades de Aprendizaje:

- **Unidad I: Introducción a la calidad de software.**
- **Unidad II: Calidad de los sistemas informáticos.**
- **Unidad III: Calidad del proceso software.**
- **Unidad IV: Evaluación y mejora de procesos.**





Estructura de los Contenidos



La competencia que el estudiante debe lograr al final de la asignatura es:

“Desarrollar y fortalecer habilidades para aplicar los diferentes modelos y normas estandarizadas en el control de calidad de los distintos sistemas informáticos”.

Índice del Contenido

I. PREFACIO	02
II. DESARROLLO DE LOS CONTENIDOS	03 - 152
UNIDAD DE APRENDIZAJE 1: INTRODUCCIÓN A LA CALIDAD DE SOFTWARE	05-41
1. Introducción	06
a. Presentación y contextualización	06
b. Competencia	06
c. Capacidades	06
d. Actitudes	06
e. Ideas básicas y contenido	06
2. Desarrollo de los temas	07-37
a. Tema 01: Calidad de software .	07
b. Tema 02: Herramientas básicas de calidad.	12
c. Tema 03: Herramientas de gestión, creatividad y estadística.	18
d. Tema 04: Herramientas de diseño y medición.	27
3. Lecturas recomendadas	38
4. Actividades	38
5. Autoevaluación	39
6. Resumen	41
UNIDAD DE APRENDIZAJE 2: CALIDAD DE LOS SISTEMAS INFORMÁTICOS	42-72
1. Introducción	43
a. Presentación y contextualización	43
b. Competencia	43
c. Capacidades	43
d. Actitudes	43
e. Ideas básicas y contenido	43
2. Desarrollo de los temas	44-68
a. Tema 01: Calidad de sistemas de información.	44
b. Tema 02: Modelo de calidad interna y externa.	52
c. Tema 03: Modelo de calidad en uso.	58
d. Tema 04: Normas ISO 9126 e ISO 14598.	63
3. Lecturas recomendadas	69
4. Actividades	69
5. Autoevaluación	71
6. Resumen	72
UNIDAD DE APRENDIZAJE 3: CALIDAD DEL PROCESO SOFTWARE	73-110
1. Introducción	74
a. Presentación y contextualización	74
b. Competencia	74
c. Capacidades	74
d. Actitudes	74
e. Ideas básicas y contenido	74
2. Desarrollo de los temas	75-106
a. Tema 01: El proceso software.	75
b. Tema 02: Modelado de procesos software.	84
c. Tema 03: Entorno PSEE.	95
d. Tema 04: Ciclo de vida.	101
3. Lecturas recomendadas	107
4. Actividades	107
5. Autoevaluación	108
6. Resumen	110
UNIDAD DE APRENDIZAJE 4: EVALUACIÓN Y MEJORA DE PROCESOS	111-149
1. Introducción	112
a. Presentación y contextualización	112
b. Competencia	112
c. Capacidades	112
d. Actitudes	112
e. Ideas básicas y contenido	112
2. Desarrollo de los temas	113-145
a. Tema 01: Medición de sistemas de información.	113
b. Tema 02: El modelo ideal y el proceso de software personal.	124
c. Tema 03: Proceso de software de equipo y el modelo CMM.	133
d. Tema 04: El estándar ISO/IEC 15504.	139
3. Lecturas recomendadas	146
4. Actividades	146
5. Autoevaluación	147
6. Resumen	149
III. GLOSARIO	150
IV. FUENTES DE INFORMACIÓN	151
V. SOLUCIONARIO	152



UNIDAD 1



Introducción a la Calidad de Software

Introducción

a) Presentación y contextualización

La calidad del software es un concepto complejo que no es directamente comparable con la calidad de la manufactura de producto. Los productos de software son uno de los principales objetivos estratégicos de muchas organizaciones debido a que los procesos más importantes de las organizaciones dependen del buen funcionamiento de los sistemas de software.

b) Competencia

Reconoce las principales herramientas y estrategias aplicadas al control de calidad de software.

c) Capacidades

1. Comprende la calidad del software como el conjunto de propiedades y características de un producto o servicio para satisfacer necesidades expresadas.
2. Reconoce las herramientas básicas de calidad aplicado a la ingeniería del software.
3. Describe las herramientas de gestión, creatividad y estadística en el control de calidad de software.
4. Aplica las fórmulas adecuadas de diseño y medición en el control de calidad de software.

d) Actitudes

- ✓ Valora las cualidades y beneficios de un producto software en el proceso de control de calidad.
- ✓ Pone en práctica las distintas herramientas de control de calidad de software.

e) Presentación de Ideas básicas y contenido esenciales de la Unidad:

La Unidad de Aprendizaje 01: Introducción a la Calidad de Software, comprende el desarrollo de los siguientes temas:

TEMA 01: Calidad de software.

TEMA 02: Herramientas básicas de calidad.

TEMA 03: Herramientas de gestión, creatividad y estadística.

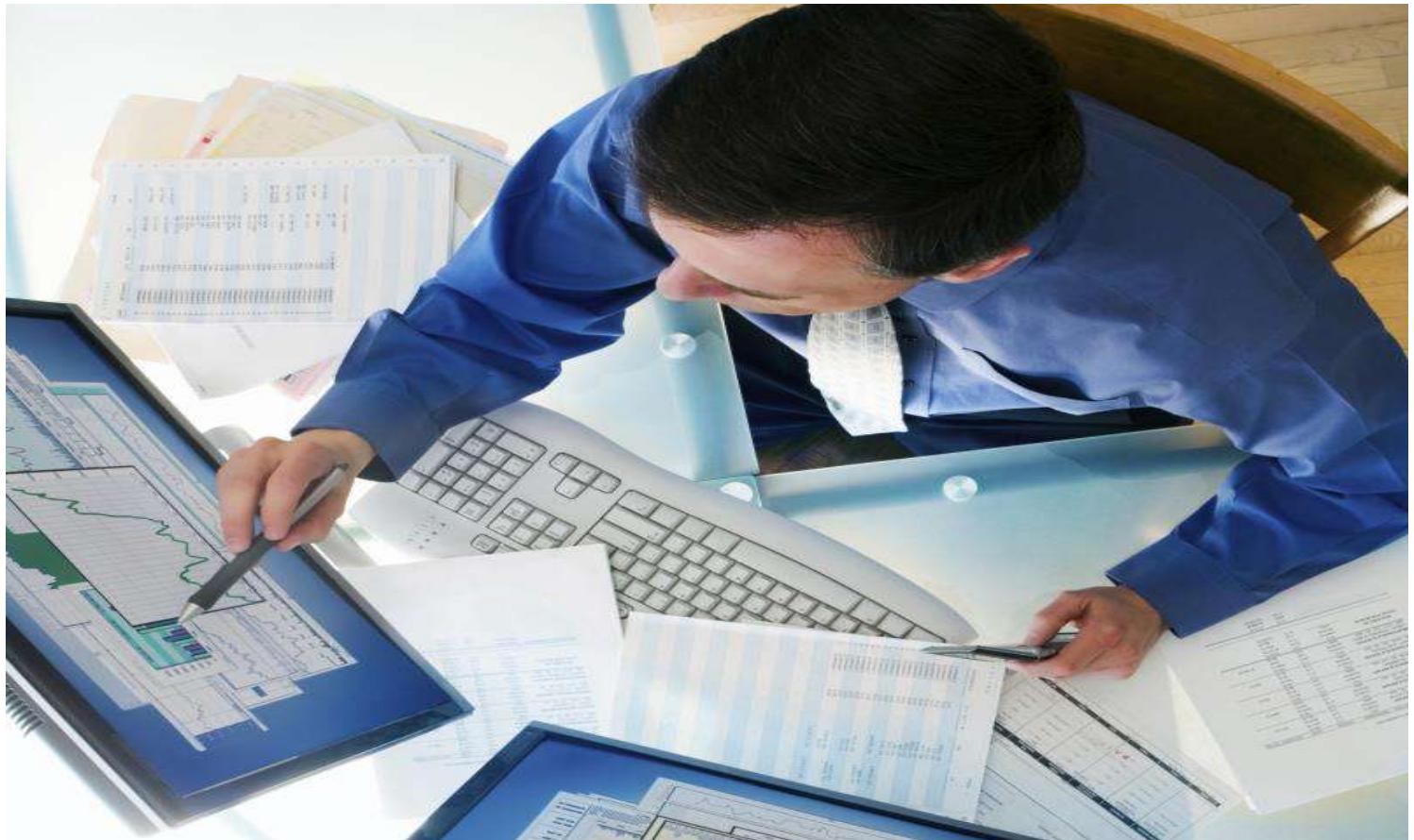
TEMA 04: Herramientas de diseño y medición.

Calidad de Software

TEMA 1



Competencia:
Comprender la calidad del software como el conjunto de propiedades y características de un producto o servicio para satisfacer necesidades expresadas.



Desarrollo de los Temas



Tema 01: Calidad de Software

ALGUNAS DEFINICIONES DE CALIDAD

Propiedad o conjunto de propiedades inherentes a un objeto que permiten apreciarlo como mejor, igual o peor que otros objetos de su especie [DRAE: Diccionario de la Real Académica Española]

Conjunto de propiedades y de características de un producto o servicio que le confieren capacidad para satisfacer necesidades expresadas o implícitas. [ISO 8042:1994]

Grado en el que un conjunto de características inherentes cumple con los requisitos. [ISO 9000: 2000]

Calidad, significa desarrollar, diseñar y producir y mantener un producto que sea el más económico, el más útil y siempre satisfactorio para el consumidor. [Kaoru Ishikawa 1943, ¿Qué es el control total de calidad?: la modalidad japonesa.]

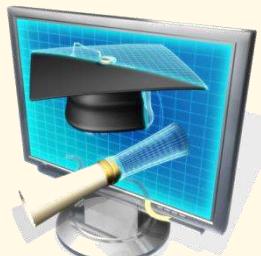
Calidad, es la aplicación de los principios y técnicas estadísticas en todas las fases de la producción, dirigida a la fabricación más económica de un producto (o servicio) que es útil en grado máximo y que tiene mercado. [William Edwards Deming 1986, Out of the Crisis. MIT Press]

Sobre el concepto de la calidad podemos decir que:

- ➡ No es absoluto
- ➡ Está sujeto a restricciones
- ➡ Trata de compromisos aceptables
- ➡ Es multidimensional
- ➡ Los criterios de calidad no son independientes



Calidad (Concepto Dinámico): La calidad está muy relacionada al desarrollo del ser humano. Por lo tanto es un concepto dinámico sujeto a diferentes definiciones según la época y el entorno en que se desenvuelve



Calidad (William Deming, 1986): Ofrecer a bajo costo productos y servicios que satisfagan a los clientes. Implica un compromiso con la innovación y mejoras continuas.

Calidad (Philip Crosby, 1995): La explica desde una perspectiva ingenieril como el cumplimiento de normas y requerimientos precisos. Su lema es “Hacerlo bien a la primera vez y conseguir cero defectos” Con estas definiciones como antecedente podemos concluir que la calidad no es un concepto absoluto más bien es algo multidimensional, ya que está sujeta a restricciones y ligada a compromisos aceptables.

La evaluación de la calidad de un producto implica una comparación entre requisitos preestablecidos y el producto desarrollado.



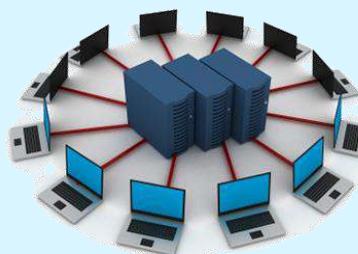
Orígenes de la Calidad

Calidad Realizada, la calidad que se ha conseguido.

Calidad Programada o Especificada, la calidad que se pretende obtener.

Calidad Necesaria o Requerida, la calidad que el cliente exige.

Lo ideal es que las tres coincidan, a la intersección entre la calidad Requerida y la calidad Realizada se llama calidad Percibida, y es la única que el cliente valora; toda calidad que se realiza pero no se necesita es un gasto inútil de tiempo y dinero.

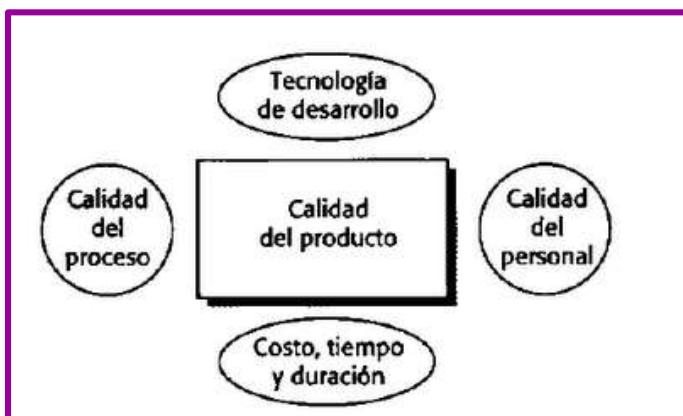


Calidad del Software

La calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario. [IEEE Standard Glossary of terminoloy of software]. La obtención de un software con calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del software.



La calidad del software no depende de un proceso de manufactura sino de un proceso de diseño en el que las capacidades del individuo son importantes. Para algunas clases de productos, el proceso utilizado es el determinante más importante de la calidad del producto. Sin embargo, para aplicaciones innovadoras en particular, la gente involucrada en el proceso es más importante que el proceso utilizado.



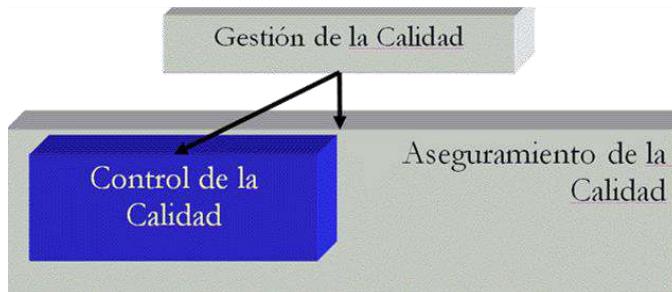
Principales factores de la calidad del producto



Terminología de Calidad del Software

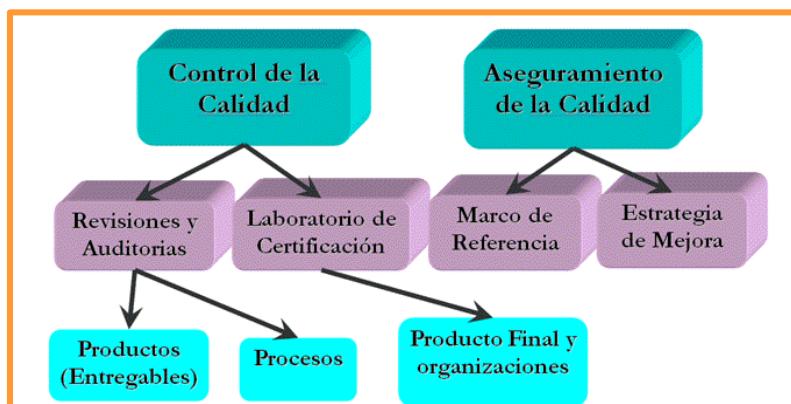
Gestión de la Calidad de Software (Software Quality Management): Conjunto de actividades de la función general de la dirección que determina la calidad, los objetivos y las responsabilidades. Se basa en la determinación y aplicación de las políticas de calidad de la empresa. La gestión o administración de la calidad se aplica normalmente a nivel empresa o dentro de la gestión de cada proyecto.

El propósito de la gestión de la calidad del software es entender las expectativas del cliente en términos de calidad, y poner en práctica un plan proactivo para satisfacer esas expectativas.



Aseguramiento de la Calidad Software (Software Quality Assurance): Conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza en que el producto (software) satisfará los requisitos dados de calidad.

Control de la Calidad de Software (Software Quality Control): Conjunto de técnicas y actividades de carácter operativo, utilizadas para verificar los requisitos relativos a la calidad, centrados en mantener bajo control el proceso de desarrollo y eliminar las causas de los defectos en las diferentes fases del ciclo de vida.



Verificación y Validación de Software (Software Verification and Validation): Conjunto de técnicas y actividades ligadas al control de calidad del software se trata de comprobar si los productos construidos en una fase de ciclo de vida satisfacen los requisitos establecidos en una fase anterior y/o si el software construido satisface los requisitos del usuario, es decir si el producto de software funciona como el usuario quiere y realiza las funciones que se habían solicitado.

Herramientas Básicas de Calidad

TEMA 2



Competencia:

Reconocer las herramientas básicas de calidad aplicado a la ingeniería del software.





Tema 02: Herramientas Básicas de Calidad

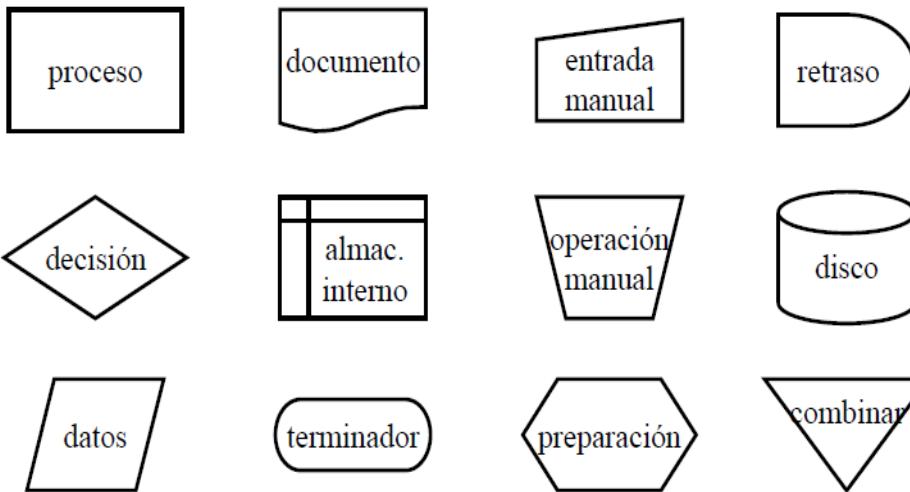


Diagrama de Flujo

Es una representación gráfica de la secuencia de etapas, operaciones, movimientos, decisiones y otros eventos que ocurren en un proceso. Puede mostrar el flujo de materiales, acciones o servicios entrando y saliendo del proceso, las decisiones a tomar y el recurso humano necesario. El diagrama de flujo nos permitirá tener una visión y compresión global del proceso, ver como se vinculan las distintas etapas, descubrir fallas presentes, además de analizar cómo se producen los problemas.

En conclusión, este diagrama de flujo nos ayuda a lograr una mejor comunicación en las discusiones y análisis. Es importante que no olvide que para desarrollar un diagrama de flujo debe utilizar los símbolos adecuados, como algunos que se muestran en la figura.

Diagrama de flujo



Para desarrollar un diagrama de flujo se recomienda seguir estos pasos:

1. Definir el proceso que debe ser representado.
2. Identificar y definir las actividades que deben ser desarrolladas y el orden en el que deben hacerlo.
3. Representar las actividades como cajas y la transición entre actividades como flechas de manera que sea posible hacer una traza de este desarrollo.
4. Revisar el diagrama de flujo con otras personas implicadas en el proceso para llegar a un consenso sobre su validez.



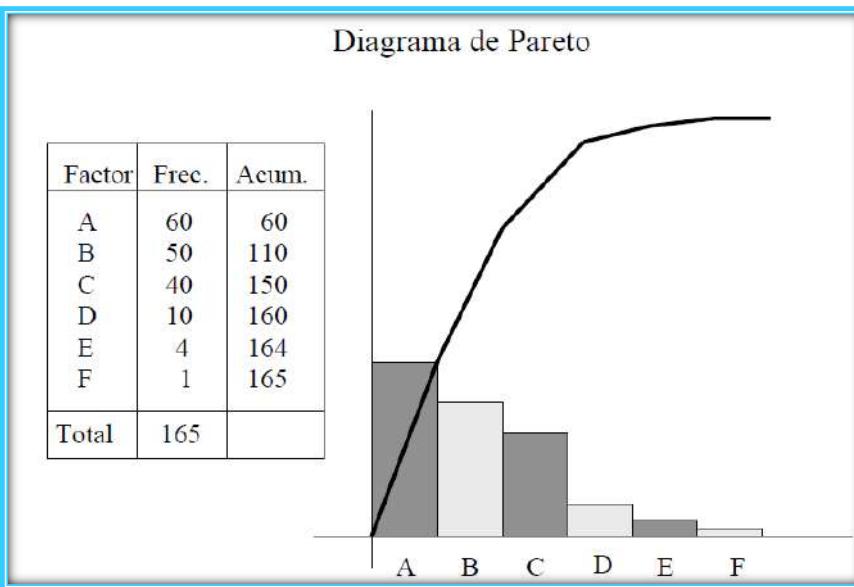
Ejemplo de diagrama de flujo



Diagrama de Pareto

La idea central del diagrama de Pareto es localizar los pocos defectos, problemas o fallas vitales para concentrar los esfuerzos de solución o mejora en estos.

Se representa a través de una gráfica de datos de conteo, donde se muestra la frecuencia de cada conteo en el eje vertical y la clasificación sobre el eje horizontal. Según la regla enunciada por Wilfrido Pareto, si se tiene un problema con muchas causas, podemos decir que el 20 % de las causas resuelven el 80 % del problema y el 80 % de las causas solo resuelven el 20 % del problema. Regla del 80 - 20



Una vez que, en un problema se ha localizado *dónde, cuándo y bajo qué circunstancias* ocurre, aplicando el *diagrama de Pareto*, entonces es el momento de localizar la causa fundamental del mismo, para ello se puede utilizar el *diagrama de Ishikawa*.

Diagrama Ishikawa (causa – efecto)

Llamada diagrama de espina de pescado (por su forma) o diagrama de Ishikawa (por su creador), el diagrama causa-efecto es una herramienta que se utiliza para identificar, explorar, y mostrar todas las posibles causas de un problema específico (efecto). Es una herramienta que, combinada con otras de identificación de problemas como la tormenta de ideas, facilita y potencia el trabajo en grupo.

Su representación consiste en un rectángulo situado a la derecha del esquema donde se indica el efecto que se quiere analizar. Se dibuja una flecha de entrada (a modo de columna vertebral del pescado) a este rectángulo, a donde llegarán las otras flechas provenientes de los posibles focos de los problemas que generan el efecto que se está estudiando.



A estas flechas, le llegarán otras secundarias con posibles sub causas relacionadas con dichos focos. A medida que el análisis vaya teniendo niveles más profundos, las subdivisiones irán ampliándose. Los focos principales suelen enunciarse como las 5 o 6 M: "**M**anos a la obra", "**M**áquinas", "**M**ateriales", "**M**edidas", "**M**edio Ambiente" y "**M**étodos".

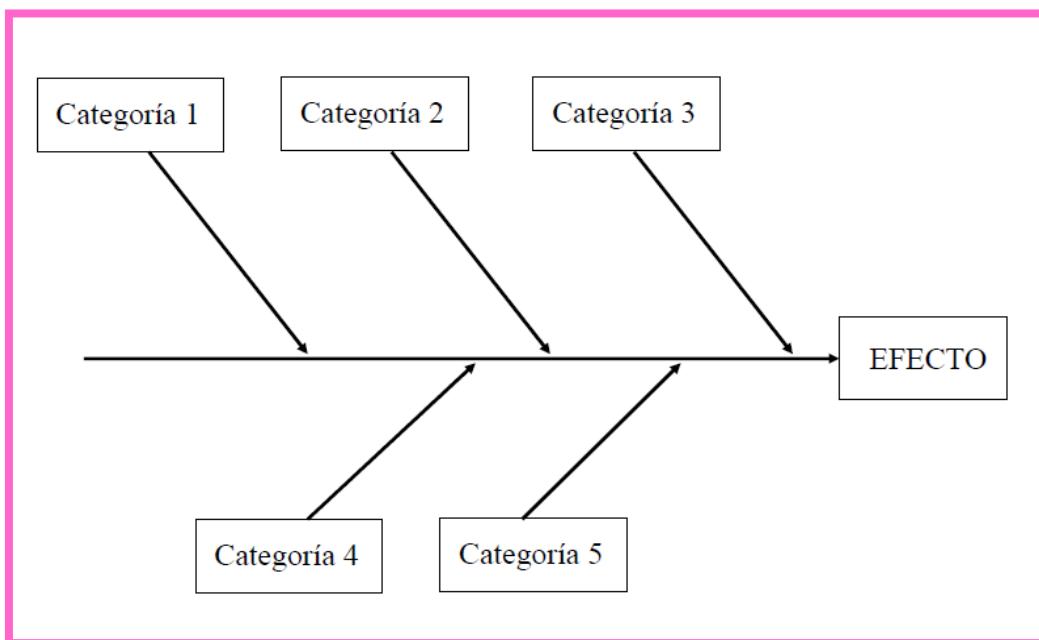


Diagrama causa – efecto

Para elaborar un diagrama de causa - efecto como el de la figura anterior se puede seguir este procedimiento:

1. Elaborar un enunciado claro del efecto (problema) que se ha detectado.
2. Dibujar el diagrama de la espina de pescado, colocando el efecto (problema) en un cuadro en el lado derecho.
3. Identificar de 3 a 6 espinas mayores que puedan ser las causas del problema / efecto principal.
4. Dibujar las espinas mayores como flechas inclinadas dirigidas a la flecha principal.

5. Identificar causas de primer nivel relacionadas con cada espina mayor.
6. Identificar causas de segundo nivel para cada causa de primer nivel.
7. Identificar causas de tercer nivel para cada causa de segundo nivel, y así sucesivamente.
8. Observando los resultados, identificar la causa raíz que permita obtener conclusiones en la resolución del problema.



Hoja de chequeo de comprobación

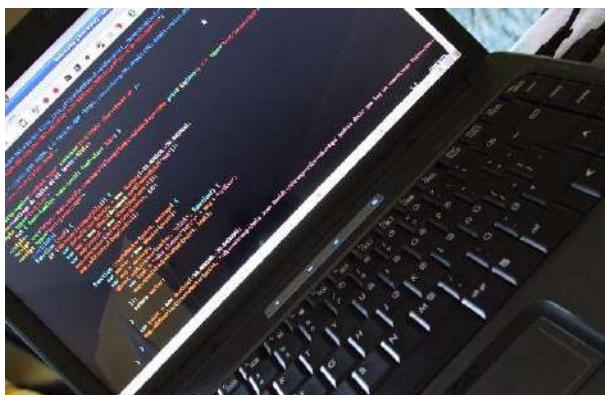
La Hoja de Recopilación de Datos también llamada Hoja de Registro. Lista de Verificación. Chequeo o Cotejo sirve para identificar y analizar tanto los problemas como sus causas. Para ello establece los Mecanismos necesarios para reunir y clasificar los datos recabados según determinadas categorías, mediante la anotación y registro de sus frecuencias para cada uno de los contextos posibles:



verificación (inspección, chequeo o tareas de mantenimiento), localización de defectos en las piezas, distribución de variaciones de variables de los artículos, clasificación de artículos defectuosos, etc. Para ello es preciso, por un lado, definir una estructura, en la que se almacenarán los datos: por otro, especificar el procedimiento de recopilación y análisis de dichos datos, indicando quien, como y cuando hacer la planificación y la captura.

De modo general las hojas de recopilación de datos se pueden clasificar según el tipo de datos en:

- ④ De verificación, inspección, chequeo o tareas de mantenimiento.
- ④ De localización de defectos en las piezas.
- ④ De distribución de variaciones de variables de los artículos (peso, volumen, longitud, calidad, etc.).
- ④ De clasificación de artículos defectuosos.



Herramientas de Gestión,

TEMA 3

Creatividad y Estadística



Competencia:

Describir las herramientas de gestión, creatividad y estadística en el control de calidad de software.





Tema 03: Herramientas de Gestión, Creatividad y Estadística

HERRAMIENTAS DE GESTIÓN

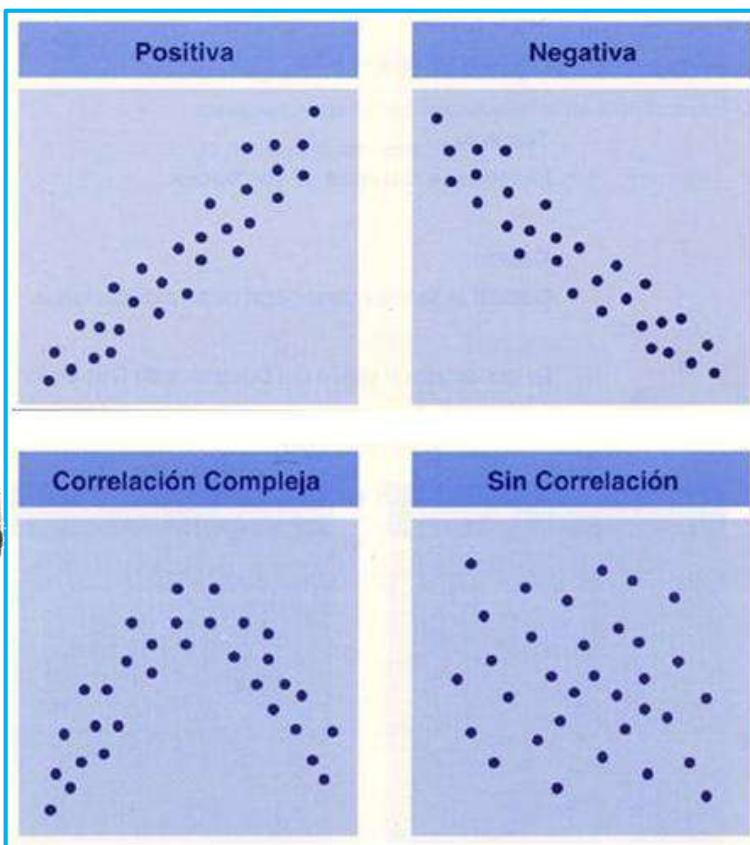
Diagrama de afinidad

Los diagramas de afinidad sirven para organizar un gran número de ideas en categorías relacionadas, o afines. Fue creado por Kawakita en los años sesenta. Las ideas suelen venir de sesiones de trabajo o de sesiones de Tormentas de Ideas.



Para elaborar un diagrama de afinidad, se recomienda seguir estos pasos:

1. Registrar todas las ideas y conceptos que sigan en el grupo de trabajo.
2. Crear categorías generales para esas ideas basándose en criterios de afinidad.
3. Asignar cada idea o concepto a dichas categorías, en función del grado de afinidad.



Tipos de correlación

Diagrama de relaciones

Es una herramienta utilizada para identificar las causas más significativas de un problema y representar gráficamente los vínculos que puedan existir entre los factores relacionados con ese problema. Esta herramienta ayuda a un grupo de trabajo a identificar los enlaces naturales entre diferentes aspectos de una situación compleja. Los diferentes elementos del diagrama se relacionan entre sí con flechas.



Ejemplo de Diagrama de Relación



Diagrama de redes de actividad o de flechas

Son una herramienta de planificación que se emplea para representar significativamente y de forma estructurada la secuenciación de actividades que hay que desarrollar en un plan de mejora de calidad siguiendo un orden cronológico. La información que se debe mostrar es la duración de cada tarea, holgura, dependencias entre actividades. Tienen un principio y un final, con lo que es posible estimar cuánto tiempo se va a necesitar para desarrollar el mencionado plan. Como las flechas indican caminos, es posible identificar caminos críticos en la realización del plan.

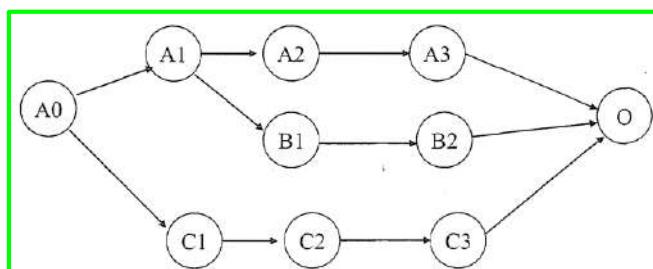


Diagrama de matriz o matricial

Al igual que las herramientas de las citadas hasta ahora, permite representar gráficamente la relación existente entre varios factores. Para ello hay que colocar los factores sobre las filas y columnas de una matriz. Si existe relación, se marca en la intersección de los factores. Es posible indicar el grado o intensidad de la relación existente. Se suele utilizar para definir la relación entre los distintos factores que intervienen directa o indirectamente en un proceso de mejora de calidad.



CARACTERÍSTICAS DE CALIDAD		Característica A	Característica B	Característica C	...	Característica M
REQUERIMIENTOS DEL CLIENTE	Requerimiento A	●	●			
	Requerimiento B	●	△	○		
	Requerimiento C		●	△		
⋮					○	
	Requerimiento N	○		●		

● Relación fuerte ○ Relación media △ Relación débil

Ejemplo de Diagrama de Relación

Diagrama de árbol

Se utiliza para representar jerárquicamente los diferentes niveles de complejidad de un determinado proceso o producto, partiendo de un primer nivel genérico que se va descomponiendo en niveles de mayor detalle hasta alcanzar un nivel básico o autodescriptivo.

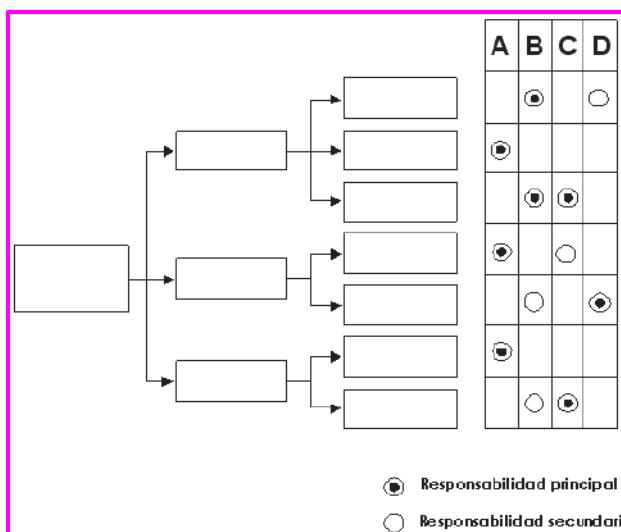


Diagrama de proceso de decisiones

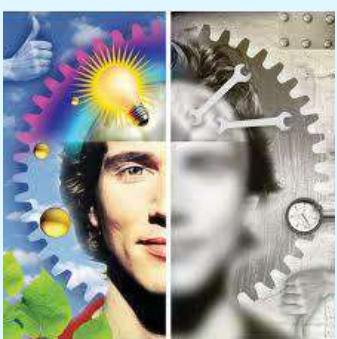
Define un plan de actuación de cara a resolver un problema determinado. Se suele utilizar para implantar planes de actuación de cierta complejidad.



Para elaborar un diagrama de este tipo, se debería seguir este procedimiento:

1. Obtener o desarrollar un diagrama de árbol con el plan propuesto, teniendo en el primer nivel el objetivo del plan, en el segundo, las actividades principales para conseguirlas y en el tercero, una lista de tareas para cada una de esas actividades.
2. Para cada tarea del tercer nivel identificar que es 10 que podría salir mal.
3. Revisar todas las listas de problemas potenciales y eliminar aquellos que sean improbables o cuyas consecuencias pudieran llegar a ser poco significativas.
4. Los problemas resultantes podrán mostrarse como un cuarto nivel.
5. Para cada problema potencial, identificar planes o acciones de contingencia que mitiguen los efectos de esos problemas. Estos planes se pueden mostrar en un quinto nivel.
6. Estudia la viabilidad de cada plan de contingencia, marcando con una "X" los impracticables y con una "O" los que Sí podrán llegarse a dar.

HERRAMIENTAS DE CREATIVIDAD



Aunque existen herramientas de creatividad como los mapas conceptuales, el uso de analogías, etc. aquí se presentan solo la tormenta de ideas como la más utilizada. Es una herramienta de trabajo en grupo basada en la creatividad de los componentes del grupo de trabajo. Se pretende obtener el mayor número de ideas o soluciones en el menor tiempo posible, seleccionando posteriormente las más indicadas, es decir, aquellas que mejor se adaptan a los objetivos del problema.

Para ella es necesario que el equipo de trabajo conozca dichos objetivos. Existen dos modos de realización de esta técnica:

- **Modo estructurado:** todos los miembros del grupo se ven forzados a participar, siguiendo un turno riguroso.
- **Modo libre:** los miembros del grupo van aportando ideas según se les van ocurriendo sin seguir ningún turno preestablecido. Se crea un ambiente más relajado pero se corre el peligro de que haya personas que no participen y por tanto no se conozcan sus ideas.

Las fases de una tormenta de idea son:

- ❖ **Definición y comunicación del asunto a tratar** a todos y cada uno de los miembros del grupo. Se tiene que planificar una agenda para facilitar la asistencia a todos los miembros.
- ❖ **Exposición de ideas.** Los participantes van apoyando ideas en alguno de los modos expuestos anteriormente y el moderador o director de la reunión las va a anotando en algún lugar visible por todos los participantes.
- ❖ **Selección de ideas.** Cuando ya no haya más ideas, todos los miembros deben seleccionar aquellas dimensiones que mejor se adapten al objetivo de la medición, descartando las peores.

HERRAMIENTAS ESTADÍSTICAS

Control estadístico del proceso

Se entiende por capacidad de un proceso el grado de aptitud que tiene para cumplir con las especificaciones técnicas deseadas. Cuando la capacidad de un proceso es alta, se dice que es capaz. Cuando se mantiene estable a lo largo del tiempo se dice que está bajo control. Para determinar si un proceso es o no capaz, se pueden utilizar las siguientes herramientas:

- Histogramas
- Gráficos de Control
- Gráficos de Probabilidad
- Estudios de índices de capacidad



Índice de capacidad

Se considera un índice de Capacidad como la relación entre la variación natural del proceso y el nivel de variación especificada. Se pueden hacer dos clasificaciones:

- Respecto a su posición:
 - a. Índices centrados con respecto a los límites
 - b. Índices descentrados con respecto a los límites pero contenidos en ellos.
 - c. Sólo con límite superior
 - d. Sólo con límite inferior
- Respecto a su alcance temporal
 - a. A corto plazo o *intragrupo* (Capacidad Potencial)
 - b. A largo plazo o *intragrupo e intergrupo* (Capacidad Global)



	CENTRADO	NO CENTRADO	CON LÍMITE SUPERIOR	CON LÍMITE INFERIOR
CORTO PLAZO (INTRAGRUPO)	C_P	C_{PK}	C_{PU}	C_{PL}
LARGO PLAZO (INTERGRUPO)	P_P	P_{PK}	P_{PU}	P_{PL}

DEFINICIÓN DE LOS TIPOS DE ÍNDICE DE CAPACIDAD

Índice de capacidad, CP PP, CPK, PPK

Sean L_S Y L_I los límites de tolerancia exigidos en las especificaciones, se define el índice de capacidad de proceso como:

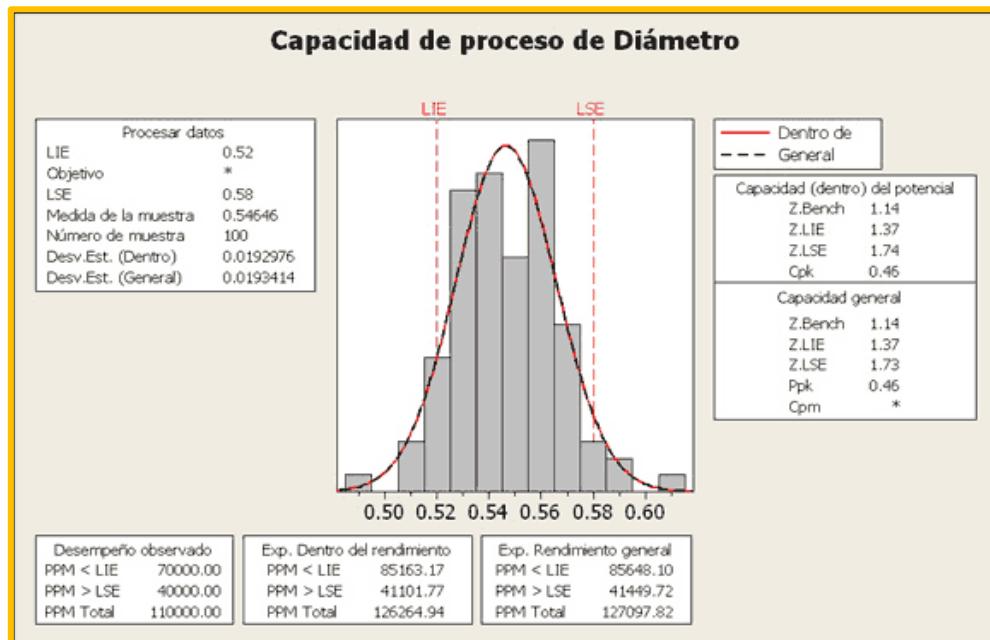


$$CP = \frac{LS - LI}{6\sigma}$$

$$C_{PK} = \text{Min} \left\{ \frac{LS - \mu}{3\sigma}, \frac{\mu - LI}{3\sigma} \right\}$$

Para afirmar que un proceso es capaz C_p y/o C_{pk} deben ser mayor o igual que 1,33, lo que garantiza que el 99,994 % de los productos fabricados o servicios prestados por el proceso centrado está dentro de las especificaciones.

En caso de ser necesario estudiar los dos, ambos deben valer como mínima 1,33. En otro caso, habrá que aplicar acciones correctoras.



Ejemplo de Estudio de Capacidades de Procesos

En la figura se muestra un ejemplo donde mediante la observación de los índices de capacidad potencial y global puede deducirse que el proceso no es capaz, ni a largo ni a corto plazo, y que por tanto habrá que aplicar correcciones.

ÍNDICES DE CAPACIDAD CPU, PPU, CPL, PPL

Se utilizan cuando el proceso solo tiene un límite de especificación, bien superior (CPU y PPU), bien inferior (CPL, PPL). Se calculan:

$$CPU = \frac{LS - \mu}{3\sigma}$$

$$CPL = \frac{\mu - LI}{3\sigma}$$

Diseño de experimentos

El Diseño de Experimentos (DDE, DOE, Design of Experiments) tiene como objetivo averiguar si unos determinados factores influyen en una o varias variables de interés para la calidad, y si se demostraría dicha influencia, cuantificarla. Las etapas de las que consta un DOE pueden resumirse en:

1. Definir los objetivos del experimento.
2. Identificar las causas posibles de variación.
3. Elegir el desafío experimental adecuado.
4. Especificar medidas y procedimiento experimental.
5. Ejecutar un experimento piloto.
6. Especificar el modelo (lineal, etc.).
7. Esquematizar los pasos del análisis estadístico.
8. Determinar el temario muestra.
9. Revisar las decisiones anteriores.



Herramientas de Diseño y Medición

TEMA 4



Competencia:

Aplicar las fórmulas adecuadas de diseño y medición en el control de calidad de software.





Tema 04: Herramientas de Diseño y Medición

HERRAMIENTAS DE DISEÑO

QFD (Quality Function Deployment)

El Diagrama de Despliegue de la Función de Calidad (*Quality Function Deployment*) es una técnica utilizada para planificar nuevos productos y servicios o realizar mejoras en los existentes a partir de métodos matriciales, cuyo objetivo es que los requisitos del cliente lleguen a estar completamente contenidos en las especificaciones técnicas del producto o servicio. La principal ventaja de esta técnica es la reducción del tiempo del desafío y la disminución de los costes, manteniendo y mejorando la calidad.



Para realizar un proyecto usando QFD se deberían seguir estos pasos:

1. **Fase de Organización:** donde se delimitará el alcance del proyecto, definiendo tanto el objetivo del proyecto como los miembros del equipo que deben trabajar en él. Estas personas deben tener experiencia demostrable y pertenecer a los distintos departamentos implicados en el proyecto.
2. **Fase de Definición.** En esta fase se realiza la programación temporal del proceso, delimitándolo en el tiempo, y planificando temporalmente la duración y las precedencias de cada una de sus tareas. También es necesario revisar el objetivo del proyecto para adaptarlo a los recursos de la empresa.
3. **Fase de Identificación y Análisis de Necesidades.** A partir de este punto comienza el desarrollo del QFD. En esta fase es donde se recopilan los requisitos del cliente, se analizan y se interpretan por los miembros del grupo de trabajo y finalmente se relacionan con las características del producto que deben sintonizar con los requisitos de los clientes. Para ella se suelen utilizar cuatro tipos de matrices importantes:

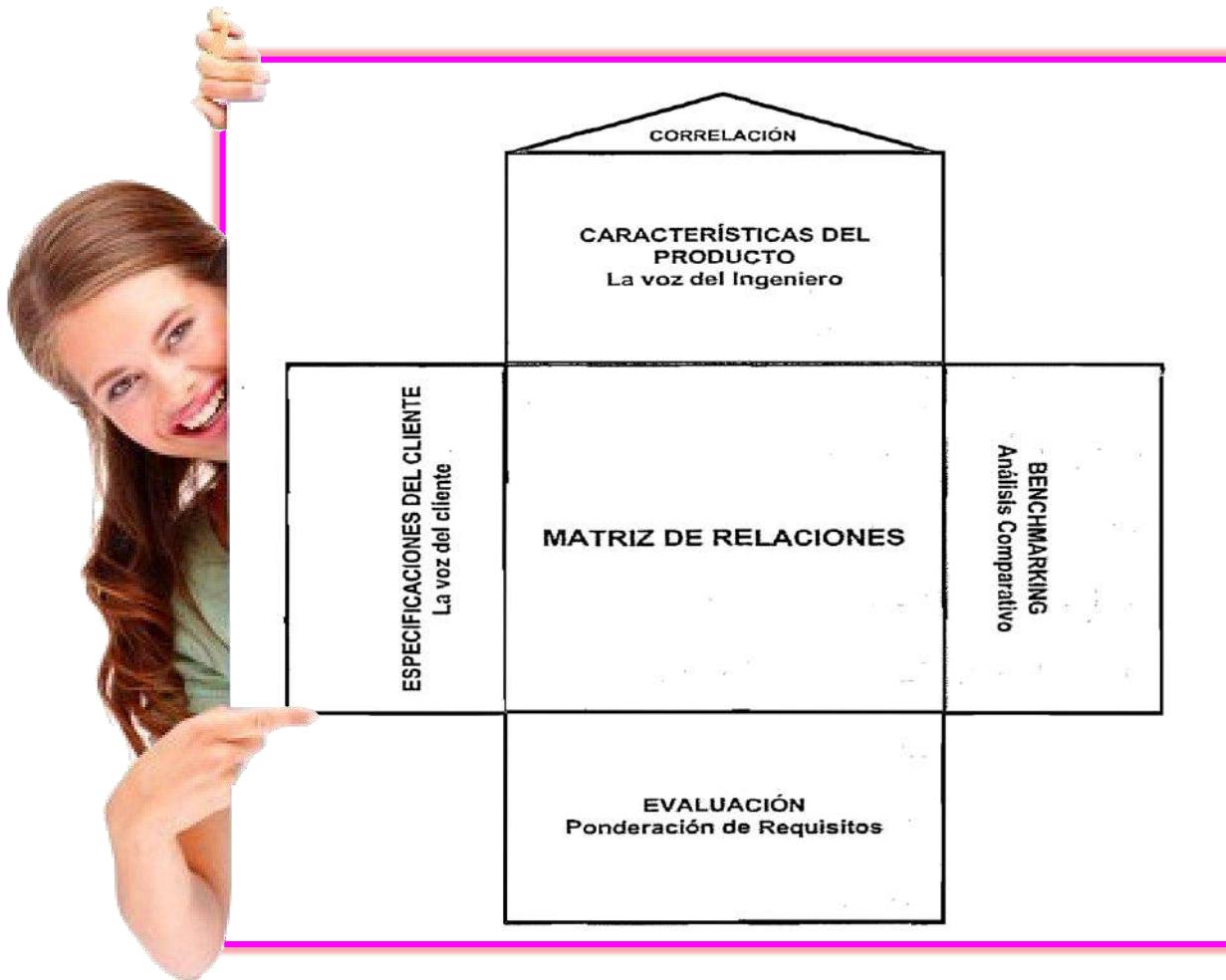


- a. **Matriz de planificación del producto o servicio ("casa de la calidad")**, donde se relacionan las necesidades del cliente con las características del producto o servicio a diseñar.
 - b. **Matriz de despliegue de componentes**, siendo su finalidad definir las especificaciones o características de las piezas, componentes o subsistemas más significativos del proceso.
-
- c. **Matriz de planificación del proceso**, donde se van a relacionar las características y requisitos de los componentes analizados y ponderados en la matriz anterior con las especificaciones del proceso de fabricación o prestación del servicio.
 - d. **Matriz de planificación de la producción**, que va a recopilar la relación entre las especificaciones del diseño (registradas en la matriz de planificación del proceso) y la normas de producción, estableciendo el procedimiento, la programación y los puntos de control del proceso de producción.

En la fase de identificación y análisis de necesidades es donde tiene lugar la planificación crítica, centrándose principalmente en las definiciones del producto o servicio. Para completar esta fase, habría que trabajar sobre cada una de las matrices de la siguiente figura; así, habría que realizar las siguientes actividades:

1. Seleccionar un Producto/Servicio Importante a Mejorar.
2. Obtener la Voz del Cliente.
3. Identificar las Necesidades del Cliente.
4. Organizar las Necesidades del Cliente.
5. Priorizar las Necesidades del Cliente.
6. Establecer los Parámetros de Diseño.
7. Generar la Matriz de Relaciones
8. Obtener la Evaluación de Desempeño del Cliente.
9. Correlacionar los Parámetros de Diseño.
10. Analizar los Resultados.
11. Iterar el Proceso.





Estructura de la Matriz "Casa de la Calidad"

AMFE (ANÁLISIS MODAL DE FALLOS Y EFECTOS)

AMFE (*Failure Modes and Effects Analysis-FMEA*) es un proceso sistemático, planificado y participativo que se aplica cuando se diseñan nuevos productos o procesos o cuando se realizan modificaciones importantes para evaluar o detectar fallos y causas que se originan antes de que lleguen al cliente. Los fallos se priorizan de acuerdo a la gravedad de sus consecuencias, de su frecuencia de aparición y de lo fácil que sea detectar esos fallos. Este proceso permite reducir costes y tiempos, mejorar y establecer un contexto de aseguramiento continuo de la calidad y aumentar la fiabilidad de los productos.



La siguiente tabla muestra la información básica que se necesita manejar para realizar un AMFE. Consta de los siguientes campos:

FUNCIÓN Y/O PROCESO	FALLO			EVALUACIÓN DE PRIORIDAD				ACCIONES CORRECTIVAS	RESULTADO					
	MODO	EFFECTO	CAUSA	CONTROLES	PREVENTIVOS	FRECUENCIA	DETECCIÓN		RESPONSABLES Y PLAZOS	FECHA APLICACIÓN	FRECUENCIAS	GRAVEDAD	DETECCION	IPR

- **Función y/o Proceso:** describe la función del elemento analizado. Si se presentan varias funcionalidades, se separan adecuadamente, ya que pueden dar lugar a distintos modos de fallo.
- **Fallo:** se refiere al incumplimiento de uno o varios requisitos o especificaciones del elemento, aunque no esté observado por el cliente

✓ *Moda de Falla:* es la forma en la que el elemento estudiado puede dejar de cumplir las especificaciones para las que fue diseñado.

✓ *Efecto de Falla:* en el caso de que se produzca el fallo, en este apartado deben completarse todos los datos correspondientes a las diferencias de funcionamiento observadas. Habría que describir a qué áreas puede afectar el fallo: si a seguridad, salud, medio ambiente, funcionamiento, correcto.

✓ *Causa de Falla:* hay que describir las anomalías de las que se tiene sospecha que han podido producir el fallo: variaciones en los parámetros. de manipulación optima, deficiencias en el diseño del producto, servicio o proceso, deficiencia en los materiales usados, uso indebido por parte del cliente, etc.

Evaluación de la Prioridad: que comprende los siguientes conceptos:

- ⊕ **Controles preventivos:** hay que reflejar los resultados de los controles preventivos previamente realizados a la aparición del fallo, para estudiar si es el resultado de un accidente fortuito o bien es por causa de alguien tipo de desgaste.
- ⊕ **Índice de Frecuencia (F):** permite asignar una probabilidad de que ocurra la causa potencial del modo de fallo.



- ⊕ **Índice de Gravedad (G):** sirve para estimar el nivel de consecuencias sentidas por el cliente. Este índice de gravedad está tabulado y es función creciente de estos factores: insatisfacción del cliente, degradación en las prestaciones. coste de reparación.
- ⊕ **Índice de Detección (D):** es el valor que mide la probabilidad de que la causa y el fallo lleguen al cliente, es decir, la probabilidad de que los índices de detección no funcionen.
- ⊕ **Índice de Prioridad de Riesgo (JPR):** mide cuales son los fallos cuyas probabilidad de riesgo es mayor. Esto permite identificar los fallos en los que se deben concentrar principalmente la atención para empezar a aplicar ahí las acciones correctoras oportunas. Se obtiene calculando el producto de los tres índices anteriores: **JPR= F·G·D.**

- ⊕ **Acciones Correctoras:** para determinar las acciones correctoras es conveniente seguir cada fallo, por lo que se debe tener en cuenta el valor del índice de Prioridad de Riesgo. En función de este índice, las acciones que se pueden asociar se puede clasificar en "*Eliminar la causa del fallo*", "*Reducir la probabilidad de ocurrencia*", "*Reducir la gravedad del fallo*", "*Aumentar la probabilidad de detección*".



- ⊕ **Responsabilidad y plazo:** sirve para anotar la persona o área que se hará cargo de la ejecución de las acciones conectoras indicadas anteriormente en los plazos previstos.
- ⊕ **Resultados:** tras adoptar las correspondientes acciones conectoras se refleja la fecha de aplicación. Tras esta fecha, se señalan los nuevos valores de los índices de frecuencia, de gravedad y de detección y se calcula de nuevo el IPR.

El procedimiento general para desarrollar cualquier tipo de AMFE podría ser el siguiente:

1. Formar un equipo multifuncional con conocimiento amplio y diverso sobre los productos, servicios, procesos y necesidades de los usuarios. Otras funciones que deberían ser capaces de desarrollar son las de diseño, producción, calidad, pruebas, fiabilidad, mantenimiento, compras (y suministros), ventas (y atención al cliente) y servicios a clientes.
2. Estimar el alcance y los límites de aplicación del AMFE, identificando el producto, proceso o servicio a estudiar, así como los modos de fallos potenciales, las causas y las posibles consecuencias.
3. Elaborar y llenar toda la documentación relativa a la evaluación de Prioridad para cada uno de los modos de los fallos potenciales objeto del estudio: esto implica llenar todos los campos de la tabla anterior. Lo que acarrea calcular los correspondientes índices de Gravedad (G) , Frecuencia (F), Detección (D) e IPR.

4. En función de los valores de IPR, estimar las acciones conectoras, identificar los responsables y estimar el plazo.
5. Ejecutar las acciones conectoras, y una vez pasada la fecha de aplicación, volver a calcular los correspondientes índices para comprobar la validez de las acciones conectoras ejecutadas.

HERRAMIENTAS DE MEDICIÓN

COQ (coste de la calidad)

Llamado también **Análisis de Costes de Pobre Calidad**, el COQ es un proceso utilizado para identificar problemas potenciales, y cuantificar los costes en los que habría que incurrir por no hacer las cosas bien desde el principio.



Para realizar un análisis COQ se recomienda seguir estos pasos:

1. Obtener o dibujar un diagrama de flujo detallado del proceso.
2. Identificar todas las fases y actividades del proceso y marcar aquellas que incurran en costes de calidad: inspección, reparación y control de daños. Cuestionarse las razones de que haya tanto muchas como pocas actividades marcadas que incurran en costes de calidad.
3. Para cada actividad marcada estimar el coste que puede implicar los fallos procedentes de una deficiente calidad y el coste que puede suponer implementar acciones bien correctoras, bien preventivas para erradicar/evitar esos problemas.
4. Estimar la viabilidad de las acciones correctoras.
5. Proponer aquellas acciones correctoras cuya viabilidad sea posible.

Benchmarking

El **benchmarking** es un proceso estructurado que permite comparar las mejores prácticas de las organizaciones, de manera que se pueden incorporar aquellas que no se desarrollan o mejorar las que se desarrollan a la propia organización, o a los procesos de la organización.



Las fases para desarrollar un benchmarking es el siguiente:

Planificar:

- a. Definir los objetivos del estudio. Hay que elegir aquellos que sean críticos para el éxito organizacional.
- b. Formar un equipo multidisciplinar que afronte firmemente el estudio que se va a desarrollar.
- c. Estudiar los propios procesos de la organización: es preciso conocer cómo funcionan las cosas internamente para hacer un buen trabajo en la comparación.
- d. Identificar los profesionales de la organización que podrían desarrollar las mejores prácticas.

Recopilar Datos:

- a. Recopilar los datos directamente de los profesionales de la organización. Hay que recoger tanto las descripciones de los procesos como los datos numéricos, usando cuestionarios, entrevistas telefónicas y/o visitas.

Analizar:

- a. Comparar los datos recolectados, tanto los numéricos como los descriptivos.
- b. Determinar las brechas entre las medidas de rendimiento de los procesos de la propia organización con los de las otras organizaciones.
- c. Determinar las diferencias en las prácticas que provocan dichas brechas.

Adaptar:

- a. Desarrollar objetivos para los procesos de la organización.
- b. Desarrollar planes de acción para conseguir esos objetivos.
- c. Implementar y monitorizar dichos planes.

Encuestas

Están destinadas a determinar la naturaleza de los procesos. Existen dos modalidades:



- ❖ Interrogación directa: los trabajadores del conocimiento interrogan verbalmente al encuestado y anota sus respuestas.
- ❖ Interrogación indirecta: se propone un cuestionario escrito.

NIVELES DE MADUREZ

Varios autores han señalado que las organizaciones pueden presentar diferentes niveles en la gestión de la calidad. Así, por ejemplo, Crosby (1979) distingue los siguientes cinco niveles:

1. Incertidumbre (Uncertain). La dirección no entiende la calidad, por lo que el personal apaga fuegos constantemente sin investigar las causas de los problemas. No hay mejora de calidad ni medidas del coste de la calidad ni muchas inspecciones.
2. Despertar (Awakening). La dirección no invierte tiempo ni dinero en la calidad, se pone énfasis en la valoración pero no en la prevención.
3. Iluminación (Enlightenment). La dirección soporta la mejora de la calidad, existiendo un departamento de calidad que reporta a la dirección.
4. Sabiduría (Wisdom). La dirección comprende la importancia de la calidad y participa en el programa de calidad, haciendo énfasis en la prevención de defectos.
5. Certidumbre (Certainty). Toda la organización está involucrada en la mejora continua.



En el mismo sentido, Silverman (1999) distingue los niveles de: aseguramiento de la calidad, resolución de problemas, alineamiento e integración, obsesión por el cliente y "despertar espiritual" (spiritual awakening); mientras que Westcott (2001) los denomina: disfuncional, despertar, desarrollo, madurez y sistema de clase mundial.

Herramientas y niveles de madurez

NIVELES DE MADUREZ	DESCRIPCIÓN	HERRAMIENTAS
BAJO	No existe sistema de calidad formal o no se usa. Reclamaciones y costes de fallos son altos. No hay mejora continua formal. Departamento de Calidad es responsable.	Auditorias Coste de Calidad Control Estadístico de Proceso
MEDIO	Costes de calidad internos altos, los externos bajos. Cada departamento acepta su papel en sistemas de gestión de calidad. Proyectos de mejora con empleados	Encuestas clientes FMEA/Dis. Exp Benchmarking
ALTO	Los sistemas de gestión de calidad, seguridad, finanzas, etc. están integrados y dirigidos por la estrategia de la organización. Los departamentos y procesos monitorizan desempeño y mejora diaria.	Herramientas de gestiόn Encuestas QFD



Lecturas Recomendadas

- ❖ **HERRAMIENTAS BÁSICAS DE CALIDAD**
<http://www.tuveras.com/calidad/herramientas/herramientas.html>

- ❖ **HERRAMIENTAS DE LA CALIDAD Y CONTROL ESTADÍSTICO DEL PROCESO**
<http://cargainfo.com/upload/CyT2012/PresentacionMabe.pdf>

Actividades y Ejercicios



1. En un documento en Word explique los sistemas (S.) de calidad (C.) existentes en los imperios Inca y Azteca, que funciones o roles existían relacionados con la calidad y quiénes las desarrollaban.
Envíalo a través de "**S. C. Incaico y Azteca**".

2. En un documento en Word señale las interpretaciones estadísticas de cada uno de los test aplicables a los gráficos X-Barra/R o X-Barra/S. Explique la interpretación estadística de los distintos coeficientes definidos para el estudio de capacidad de procesos.
Envíalo a través de "**Interpretaciones Estadísticas**".

Autoevaluación

- 1) No es absoluto, Está sujeto a restricciones, trata de compromisos aceptables, es multidimensional, los criterios de calidad no son independientes, es uno de los conceptos de:
 - a. Evaluación.
 - b. Calidad.
 - c. Estrategias.
 - d. Validación.
 - e. Calidad de software.

- 2) Menciona tres de los cinco principales factores de la calidad del producto.
 - a. Tecnología de desarrollo, Calidad de proceso, Calidad de estrategia.
 - b. Calidad del personal, Calidad de software, Calidad del producto.
 - c. Tecnología de desarrollo, Calidad de proceso, Calidad del producto.
 - d. Tecnología de desarrollo, Costo tiempo y Duración calidad en evaluación.
 - e. Calidad del personal, Calidad de sistema, Calidad del producción.

- 3) La calidad del software depende de _____ en el que las capacidades del individuo son importantes.
 - a. Un proceso de diseño.
 - b. Un proceso de manufactura.
 - c. El lenguaje de programación usado.
 - d. La calidad requerida y la calidad realizada.
 - e. La calidad específica.

- 4) Conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza en que el producto (software) es el concepto de:
 - a. Control de la calidad.
 - b. Aseguramiento de la Calidad de Software.
 - c. Revisiones y auditorias.
 - d. Producto final y organización.
 - e. Estrategia de mejora.

- 5) _____ es una representación gráfica de la secuencia de etapas, operaciones, movimientos, decisiones y otros eventos que ocurren en un proceso.
 - a. Operación manual.
 - b. Almacenamiento.
 - c. Diagrama de Pareto.
 - d. Diagrama de Flujo.
 - e. Diagrama de Ishikawa.

- 6) El _____ llamado es también diagrama de espina de pescado.
- a. Diagrama de Ishikawa.
 - b. Diagrama de Pareto.
 - c. Diagrama de Flujo.
 - d. Diagrama de Comprobación.
 - e. Diagrama de Almacenamiento.
- 7) Pertenece a las herramientas estadísticas
- a. Histogramas y Gráficos de probabilidad.
 - b. Gráficos de control y proceso de decisiones.
 - c. Diagrama de árbol y estudio de índice de capacidad.
 - d. Diagrama de matriz e histogramas.
 - e. Gráficos de control y diagrama de árbol.
- 8) La _____ es la que va a recopilar la relación entre las especificaciones del diseño y la normas de producción, estableciendo el procedimiento, la programación y los puntos de control del proceso de producción.
- a. Matriz de planificación del proceso.
 - b. Matriz de despliegue de componentes.
 - c. Matriz de planificación de la producción.
 - d. Matriz de planificación del producto o servicio.
 - e. Matriz de recopilación de diseño.
- 9) El _____ es un proceso estructurado que permite comparar las mejores prácticas de las organizaciones.
- a. Benchmarking.
 - b. Recopilar datos.
 - c. Analizar.
 - d. Planificar.
 - e. AMFE.
- 10) En la _____ se realiza la programación temporal del proceso, delimitándolo en el tiempo, y planificando temporalmente la duración y las precedencias de cada una de sus tareas.
- a. Fase de organización.
 - b. Fase de identificación.
 - c. Fase de definición.
 - d. Fase de análisis.
 - e. Fase de diseños.

Resumen

UNIDAD DE APRENDIZAJE I: INTRODUCCIÓN A LA CALIDAD DE SOFTWARE

El texto es un conjunto ordenado de ideas relacionadas entre sí y en torno a un mismo tema. La calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario. La calidad del software no depende de un proceso de manufactura sino de un proceso de diseño en el que las capacidades del individuo son importantes.

Herramientas básicas de calidad: Diagrama de Flujo, que es una representación gráfica. Diagrama de Pareto, que consiste en localizar los defectos. Diagrama Ishikawa (causa – efecto), también llamado diagrama de espina de pescado, es una herramienta que se utiliza para identificar, explorar, y mostrar todas las posibles causas de un problema específico. Hoja de chequeo de comprobación, es una lista de Verificación.

Herramientas de gestión, creatividad y estadística. Diagrama de afinidad. Diagrama de relaciones. Diagrama de redes de actividad o de flechas. Diagrama de matriz o matricial. Diagrama de árbol. Diagrama de proceso de decisiones. Herramientas de creatividad. Herramientas estadísticas. Control estadístico del proceso. Índice de capacidad.

Herramientas de diseño y de medición: Diagrama de Despliegue de la Función de Calidad, es una técnica utilizada para planificar nuevos productos y servicios. Análisis Modal de Fallos y Efectos (AMFE), es un proceso sistemático, planificado y participativo que se aplica cuando se diseñan nuevos productos. Herramientas de medición: COQ (coste de la calidad), Benchmarking y Encuestas.



UNIDAD 2



Calidad de los Sistemas Informáticos

Introducción

a) Presentación y contextualización

La calidad de una empresa u organización depende de la calidad de los procesos de negocio soportados por el sistema de información, así como la propia calidad de este. En la calidad de un producto software, así como en las métricas asociadas en las diferentes etapas del ciclo de vida del software, se suelen distinguir tres aspectos diferentes: calidad interna: medible a partir de las características intrínsecas, como el código fuente; calidad externa; medible en el comportamiento del producto, como en una prueba; o en uso: medible durante la utilización efectiva por parte del usuario en un contexto determinado.

b) Competencia

Analiza las principales características de los modelos de control de información, describiendo su funcionalidad para su empleo adecuado dentro de cualquier organización.

c) Capacidades

1. Comprende la importancia de los sistemas de información y el proceso adecuado de control de calidad de software.
2. Reconocer las principales estrategias que representan al modelo de calidad externa e interna.
3. Describe las características del modelo de calidad de uso y la evaluación de un producto software.
4. Aplica las normas ISO 9126 e ISO 14598 en el control de calidad de software.

d) Actitudes

- ✓ Promueve el cumplimiento de las normas ISO 9126 e ISO 14598.
- ✓ Valora los distintos modelos de control de calidad de la información.

e) Presentación de Ideas básicas y contenido esenciales de la Unidad:

La Unidad de Aprendizaje 02: Calidad de los Sistemas Informáticos, comprende el desarrollo de los siguientes temas:

TEMA 01: Calidad de Sistemas de Información.

TEMA 02: Modelo de Calidad Interna y Externa.

TEMA 03: Modelo de Calidad en Uso.

TEMA 04: Normas ISO 9126 e ISO 14598.

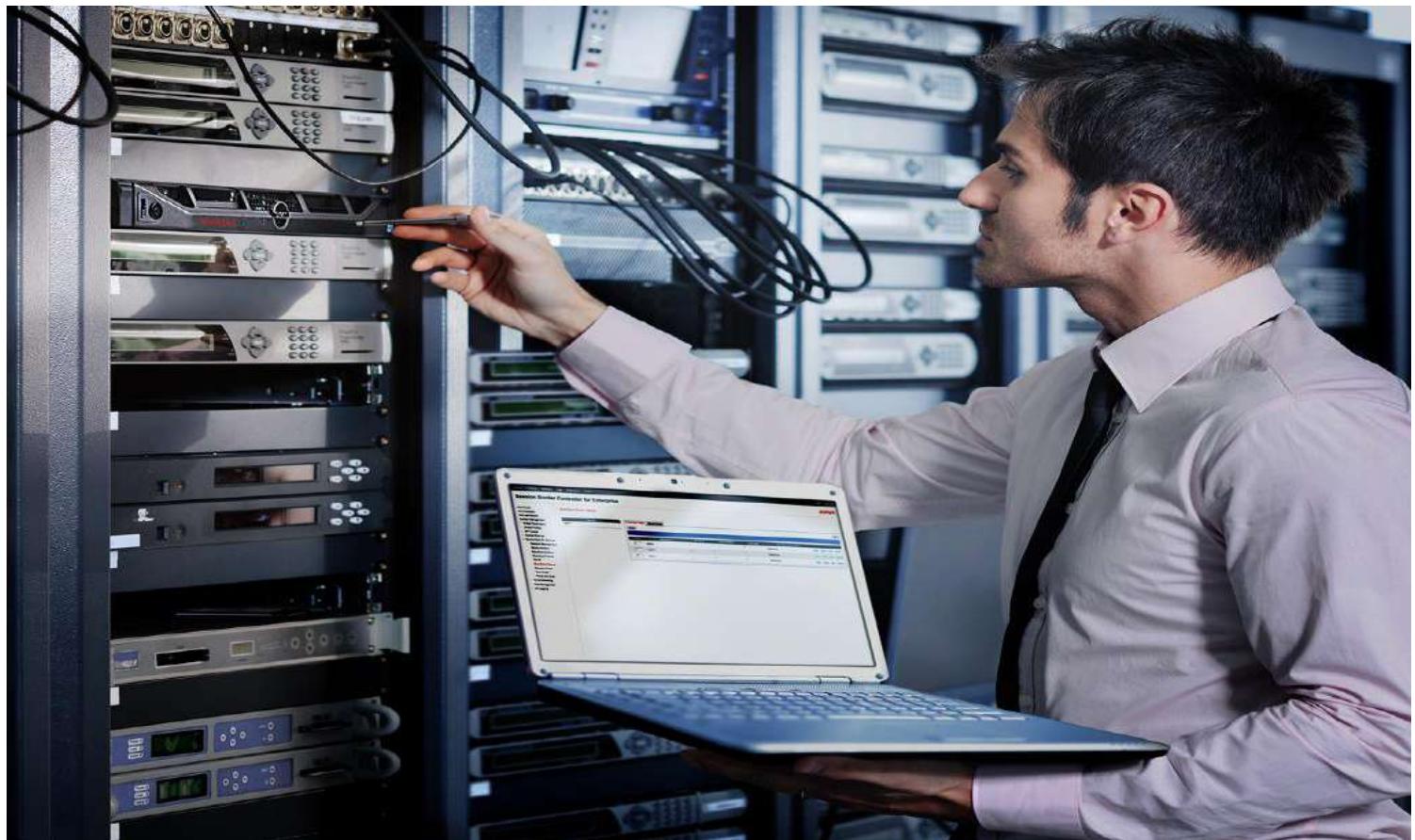
Calidad de Sistemas de Información

TEMA 1



Competencia:

Comprender la importancia de los sistemas de información y el proceso adecuado de control de calidad de software.



Desarrollo de los Temas



Tema 01: Calidad de Sistemas de Información

COMPONENTES DE LA CALIDAD

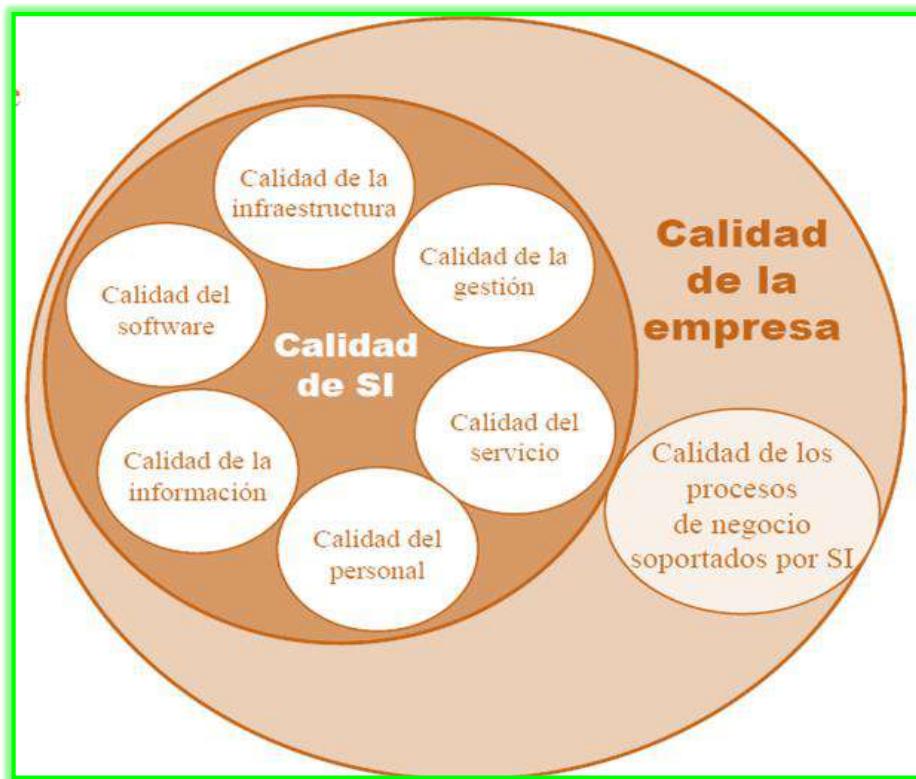
La calidad de un sistema informático (SI) puede descomponerse en diferentes factores que contribuyen a la misma.



En Wilkin y Castleman (2003) se describe un instrumento multidimensional (denominado QUALIT) capaz de medir la calidad de los sistemas de información entregados, en el que se diferencia entre la calidad del sistema (entendida como juicio global sobre el grado en que los componentes técnicos del mismo proporcionan la calidad de la información y servicio requerido por los stakeholders), calidad de la información proporcionada a los stakeholders (excluyendo manuales de usuario y pantallas de ayuda), calidad del servicio (proporcionado por el departamento de SI y el personal de soporte).

Para cada uno de estos componentes los autores han identificado varias dimensiones; así por ejemplo, para la calidad del sistema se consideran: funcionalidad, integración, usabilidad, fiabilidad y seguridad. Por otra parte, Stylianou y Kumar (2000) proponen una "visión holística" de la calidad de los sistemas de información, en la que se consideren diferentes dimensiones.





Dimensiones de calidad de SI, basada en Stylianou y Kumar (2000)

La calidad de una empresa u organización depende de la calidad de los procesos de negocio soportados por el sistema de información, así como la propia calidad de este.

A su vez en la calidad del sistema de información podremos distinguir:

- ❖ Calidad de la infraestructura
- ❖ Calidad de la gestión
- ❖ Calidad del servicio
- ❖ Calidad del personal
- ❖ Calidad de la información.

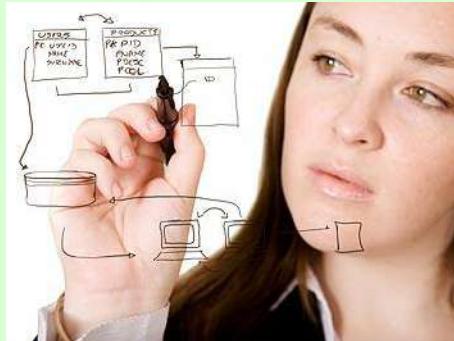


CALIDAD DEL SOFTWARE

Modelos clásicos

Históricamente se han desarrollado para evaluar la calidad de los productos software diferentes modelos que pretenden seguir las directrices de calidad de otros tipos de productos: descomponer la calidad en una categoría de características más sencillas que facilita su estudio (Galin, 2004).

Uno de los modelos clásicos más utilizados desde su creación, incluso con vigencia en nuestros días, es el desarrollado por McCall (McCall et al., 1977), en el que la calidad de un producto software se descompone en once características o factores de calidad agrupados en tres categorías: Operación de producto, Revisión de producto y transición de producto.



A finales de los años ochenta, fueron propuestos dos modelos alternativos a los de McCall basados igualmente en la identificación de factores: el modelo de factores de Evans y Marciniak (1987) y el modelo de factores de Deutsch y Willis (1988).

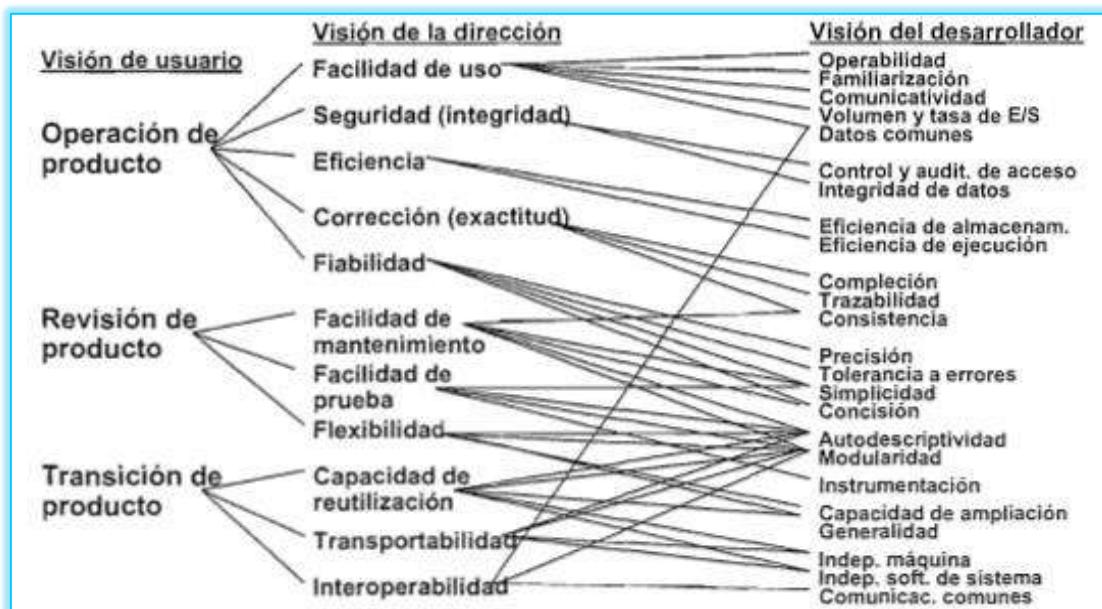
En la siguiente tabla puede encontrarse una comparativa entre los distintos modelos donde se muestran los factores observados por cada uno de los autores en sus correspondientes trabajos.

Otro modelo considerado como clásico es el reconocido como FURPS, acrónimo compuesto por las iniciales en inglés de las categorías Funcionalidad, Facilidad de uso, Fiabilidad, Rendimiento y Capacidad del software;



esta lista es una de las muchas adaptaciones y/o complementaciones del modelo de McCall que cada organización ha ideado para sus propios trabajos, como la dada por Grady y Caswell (1987) para Hewlett Packard.

Modelo de calidad de McCall (1976)



Comparación entre modelos de calidad de producto software (Galin, 2004)

Factor Calidad Software	McCall (1976)	Evansy Marcinlak (1987)	Deutsch y Willis (1988)
Corrección	✓	✓	✓
Fiabilidad	✓	✓	✓
Eficiencia	✓	✓	✓
Integridad	✓	✓	✓
Usabilidad	✓	✓	✓
Mantenibilidad	✓	✓	✓
Flexibilidad	✓	✓	✓
Testeabilidad	✓		
Portabilidad	✓	✓	✓
Reusabilidad	✓	✓	✓
Inter operatividad	✓		✓
Verificabilidad		✓	✓
Expandibilidad		✓	✓
Seguridad de uso			✓
Manejabilidad			✓
Capacidad de supervivencia			✓

Normas ISO 25000

El SC7 de ISO está desarrollando la familia de normas ISO 25000 (ISO 2005a-n) conocida con el nombre de SQuaRE (Software product Quality Requirements and Evaluation) que se organiza en cinco apartados y que sustituye y amplia las actuales normas ISO 9126 (ISO, 1991; Tecnología de la Información - Calidad de un producto software) y 14598 (ISO, 1999; Tecnología de la Información- Evaluación de un producto software).



Organización de la familia de normas ISO 25000

Requisitos de Calidad 2503n	Modelo de Calidad 2501n	Evaluación de Calidad 2504n
	Gestión de Calidad 2500n	
	Medición de calidad 2502n	



ISO/IEC 2500n - División de Gestión de Calidad. Las normas que forman este apartado definen todos los modelos, términos y definiciones comunes referenciados por todas las otras normas de la serie SQUARE.

ISO/IEC 2501n - División de Modelo de Calidad. La norma de este apartado presenta un modelo de calidad detallada incluyendo características para calidad interna, externa y en uso.

ISO/IEC 2502n - División de Medición de Calidad. Estas normas incluyen un modelo de referencia de la medición de la calidad del producto, definiciones de medidas de calidad (interna, externa y en uso) y guías prácticas para su aplicación.

ISO/IEC 2503n - División de Requisitos de Calidad. Estas normas ayudan a especificar requisitos de calidad que pueden ser utilizados en el proceso de elicitación de requisitos de calidad del producto software a desarrollar o como entrada del proceso de evaluación.

ISO/IEC 2504n -División de Evaluación de Calidad. Este apartado incluye normas que proporcionan requisitos, recomendaciones y guías para la evaluación de productos software.



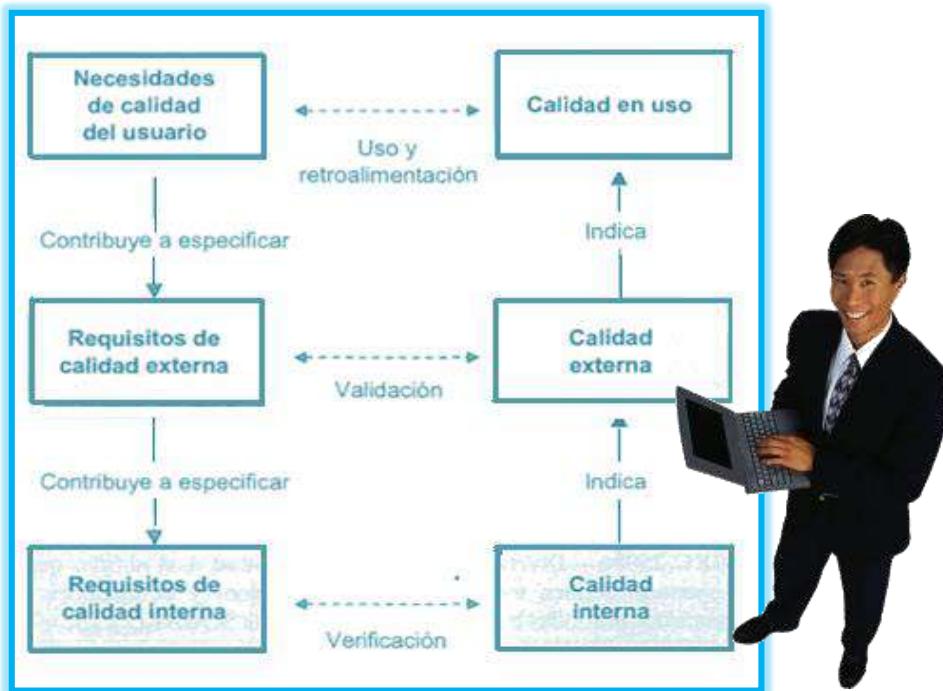
Divisiones de SQuaRE



ISO 9126 e ISO 14598, ya que probablemente los conceptos básicos se mantengan con pocos cambios significativos en las nuevas normas.



Perspectivas de calidad según la norma ISO 9126



Aspectos de la calidad de un producto software

En la calidad de un producto software, así como en las métricas asociadas en las diferentes etapas del ciclo de vida del software, se suelen distinguir tres aspectos diferentes: calidad interna: medible a partir de las características intrínsecas, como el código fuente; calidad externa; medible en el comportamiento del producto, como en una prueba; o en uso: medible durante la utilización efectiva por parte del usuario en un contexto determinado.

Siguiendo la filosofía de los modelos clásicos de calidad de un producto software, la norma ISO 9126 descompone la calidad jerárquicamente en una serie de características y subcaracterísticas que pueden usarse como una lista de comprobación de aspectos relacionados con la calidad.

Modelo de Calidad Internacional y Externa

TEMA 2



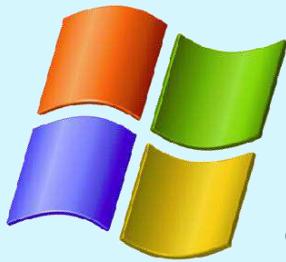
Competencia:

Reconocer las principales estrategias que representan al modelo de calidad externa e interna.





Tema 02: Modelo de Calidad Interna y Externa



El modelo de calidad para calidad interna y externa categoriza los atributos de calidad software en seis características (funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad), que se subdividen a su vez en subcaracterísticas, que se resume a continuación (ISO, 2001).

MODELO PARA LA CALIDAD EXTERNA E INTERNA (ISO, 2001)

Categorías del Producto	Características Externas	Características Internas
Funcionalidad	Exactitud Interoperabilidad Seguridad, Conformidad	Estructurado, Especificado
Fiabilidad	Madurez, Tolerancia a Fallos, Recuperabilidad	Correcto
Usabilidad	Compresibilidad, Facilidad de Aprendizaje, Interfaz Grafica, Operabilidad.	Completo, Efectivo, Documentado.
Eficiencia	Comportamiento Temporal, Utilización de Recursos	-
Mantenibilidad	Capacidad de Análisis, de Cambio, de Pruebas, Estabilidad	Acoplado, Cohesivo, Encapsulado, Simple, Estructurado, Descriptivo, Modular
Portabilidad	Adaptabilidad, Facilidad de Instalación, Conformidad	Parametrizado, Documentado,

Funcionalidad

Capacidad del producto software para proporcionar funciones que satisfacen necesidades declaradas e implícitas cuando se usa bajo condiciones especificadas. Esta característica se subdivide a su vez en:

- **Adecuación.** Capacidad del producto software para proporcionar un con junto 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.

Fiabilidad

Capacidad del producto software para mantener un nivel especificado de prestaciones cuando se usa bajo condiciones especificadas. Esta característica se subdivide a su vez en:

- ⊕ **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.

Usabilidad

Capacidad del producto software para ser entendido, aprendido, usado y ser atractivo para el usuario, cuando se usa bajo condiciones especificadas. Esta característica se subdivide a su vez en:

- + **Capacidad para ser entendido.** Capacidad del producto software que permite al usuario entender si el software es adecuado y como 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.

Eficiencia

Capacidad del producto software para proporcionar prestaciones apropiadas, relativas a la cantidad de recursos usados, bajo condiciones determinadas. Esta característica se subdivide a su vez en:

- **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

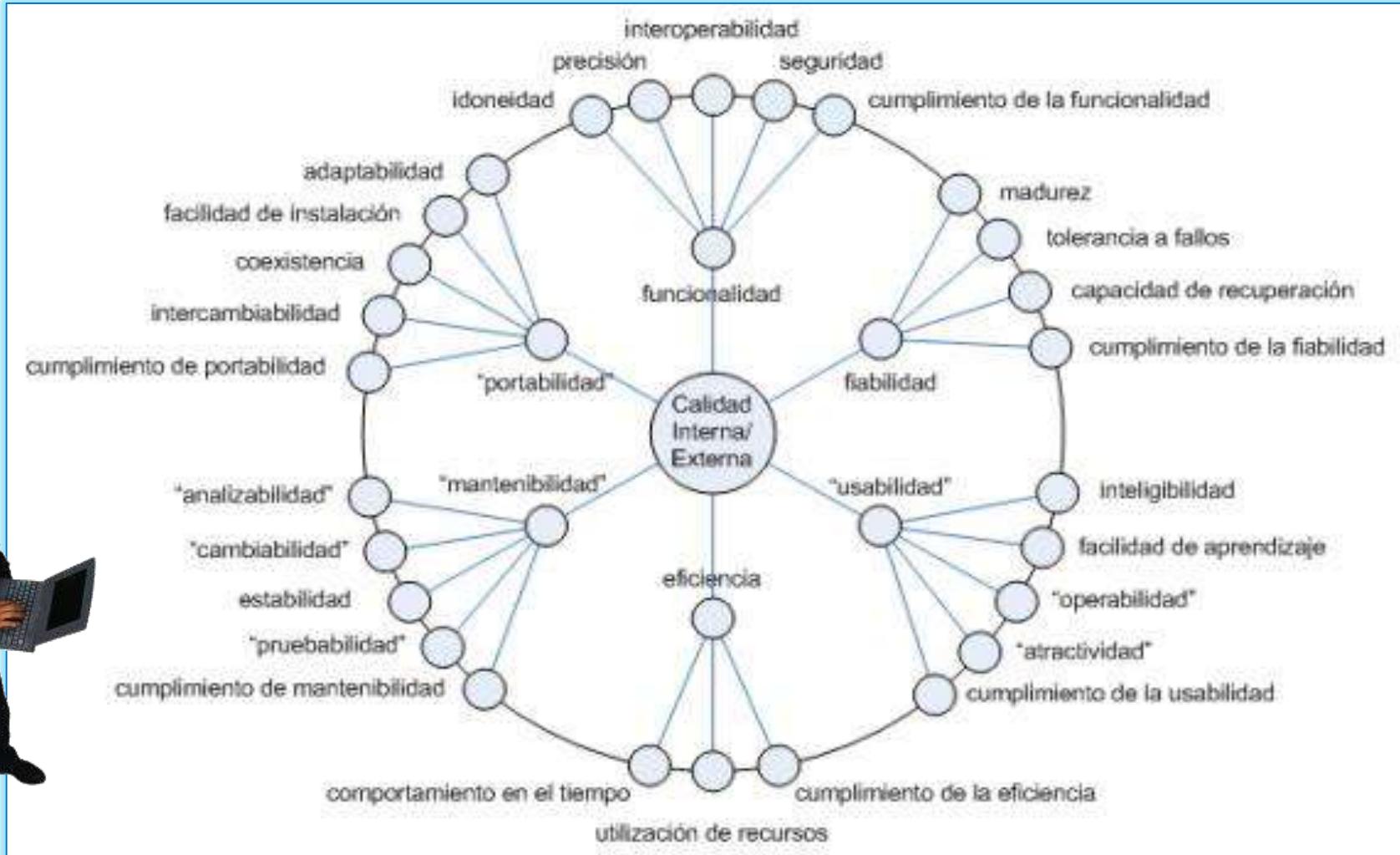
Capacidad del producto software para ser modificado. Las modificaciones podrían incluir correcciones, mejoras o adaptación del software a cambios en el entorno, y requisitos y especificaciones funcionales. Esta característica se subdivide a su vez en:

- **Capacidad para ser analizado.** Es la capacidad del producto software para serle diagnosticada las 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

Capacidad del producto software para ser transferido de un entorno a otro. Esta característica se subdivide a su vez en:

- **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 ser reemplazado.** 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.



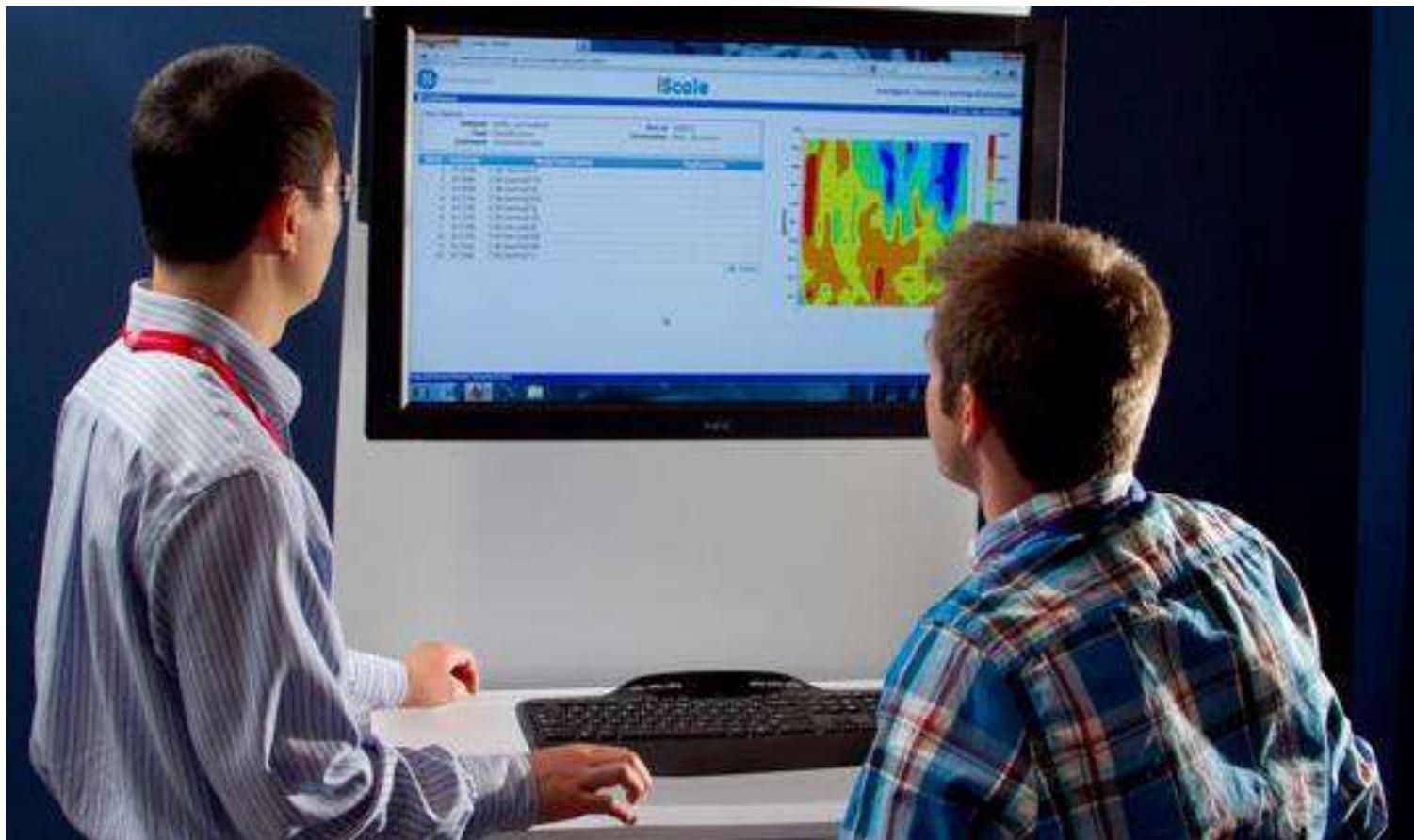
TEMA 3

Modelo de Calidad en Uso



Competencia:

Describir las características del modelo de calidad de uso y la evaluación de un producto software.





Tema 03: Modelo de Calidad en Uso

La norma ISO 9126 entiende por calidad en uso "la capacidad del producto software para permitir a determinados usuarios alcanzar objetivos especificados con efectividad, productividad, seguridad y satisfacción, en contextos de uso especificados".



La calidad en uso contempla las siguientes características:



Efectividad

Capacidad del producto software para permitir a los usuarios alcanzar objetivos especificados con exactitud y compleción, 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.

Modelo para la calidad en uso (ISO, 2005)





Seguridad de uso

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.

EVALUACIÓN DE UN PRODUCTO SOFTWARE

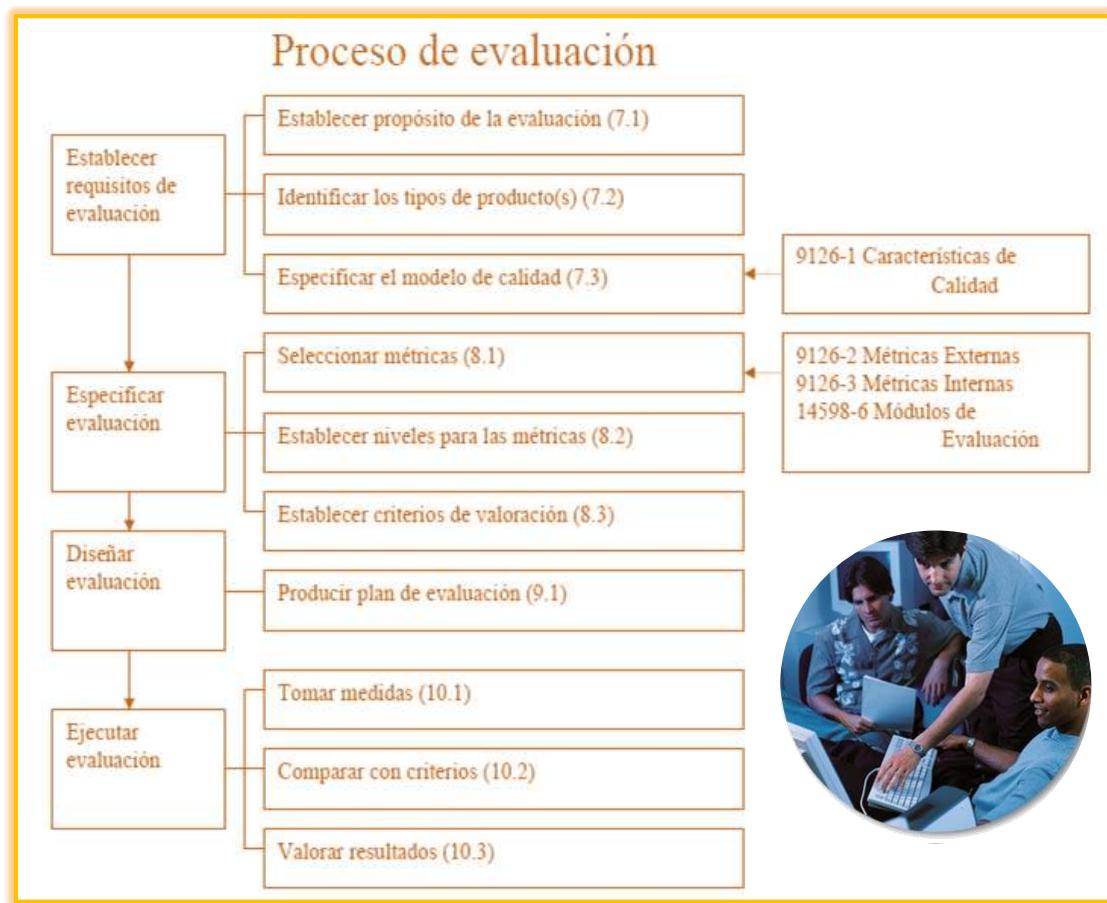
La norma ISO 14598 da una visión general del proceso de evaluación de un producto software, explicando en sus diferentes partes como aplicar el proceso en diferentes circunstancias. Esta norma se apoya en la ISO 9126 ya que los aspectos cuantificables pueden medirse cuantitativamente usando métricas de calidad, cuyo valor medido se sima en una escala. La escala ha de dividirse en rangos que corresponden a diferentes niveles de satisfacción de los requisitos.

Por ejemplo:

- ❖ La división de la escala en dos categorías: satisfactorio e insatisfactorio.
- ❖ La división de la escala en cuatro categorías limitadas por el nivel actual de un producto existente o alternativo, el peor caso y el nivel proyectado. El nivel actual se declara con el fin de controlar que el nuevo sistema no suponga un deterioro en relación a la situación actual. El nivel proyectado es lo que se considera alcanzable con los recursos disponibles. El peor caso es la frontera para aceptación del usuario por si acaso el producto no cubre el nivel proyectado.



Proceso de evaluación de un producto software (ISO, 1999)



Rangos de una escala de medida (ISO, 1999)



Criterios para medir y evaluar calidad en uso

Si se considera que la evaluación de calidad en uso se realiza sobre un producto en funcionamiento, es necesario emplear



De entre los modelos de proceso de evaluación es importante referirse al [ISO14598-5], que estaba incluido originalmente en [ISO9126], y que tiene como objetivo principal proveer un marco de evaluación genérico, abstracto, que permita a los evaluadores, junto con desarrolladores, compradores, vendedores y usuarios en general expresar requerimientos de calidad siguiendo el modelo de calidad definido en el estándar [ISO9126-1]. La evaluación debe tener en cuenta una variedad de documentos que pueden ser considerados parte del producto de software, tales como documentación de diseño, código fuente, tests o documentación para el usuario final.



Normas

ISO

9126 e ISO

14598

TEMA 4



Competencia:

Aplicar las normas ISO 9126 e ISO 14598 en el control de calidad de software.





Tema 04: Normas ISO 9126 e ISO 14598



ISO 9126 es un estándar internacional para la evaluación de la calidad del software. Está reemplazado por el proyecto SQuaRE, ISO 25000:2005, el cual sigue los mismos conceptos. El estándar está dividido en cuatro partes las cuales dirigen, respectivamente, lo siguiente: modelo de calidad, métricas externas, métricas internas y calidad en las métricas de uso y expendido.

El modelo de calidad establecido en la primera parte del estándar, ISO 9126-1, clasifica la calidad del software en un conjunto estructurado de características y subcaracterísticas de la siguiente manera:

- | | |
|-------------------------|-------------------------------|
| Funcionalidad. | Eficiencia. |
| Idoneidad. | Atractividad. |
| Exactitud. | Comportamiento en el tiempo. |
| Interoperabilidad. | Comportamiento de recursos. |
| Seguridad. | Mantenibilidad. |
| Cumplimiento de normas. | Estabilidad. |
| Fiabilidad. | Facilidad de análisis. |
| Madurez. | Facilidad de cambio. |
| Recuperabilidad. | Facilidad de pruebas. |
| Tolerancia a fallos. | Portabilidad. |
| Usabilidad. | Capacidad de instalación. |
| Aprendizaje. | Capacidad de reemplazamiento. |
| Comprensión. | Adaptabilidad. |
| Operatividad. | Co-Existencia |

Cada subcaracterística (como adaptabilidad) está dividida en atributos. Un atributo es una entidad la cual puede ser verificada o medida en el producto software. Los atributos no están definidos en el estándar, ya que varían entre diferentes productos software.



Un producto software está definido en un sentido amplio como: los ejecutables, código fuente, descripciones de arquitectura, y así, como resultado, la noción de usuario se amplía tanto a operadores como a programadores, los cuales son usuarios de componentes como son bibliotecas software.

El estándar provee un entorno para que las organizaciones definan un modelo de calidad para el producto software. Haciendo esto así, sin embargo, se lleva a cada organización la tarea de especificar precisamente su propio modelo. Esto podría ser hecho, por ejemplo, especificando los objetivos para las métricas de calidad las cuales evalúan el grado de presencia de los atributos de calidad.

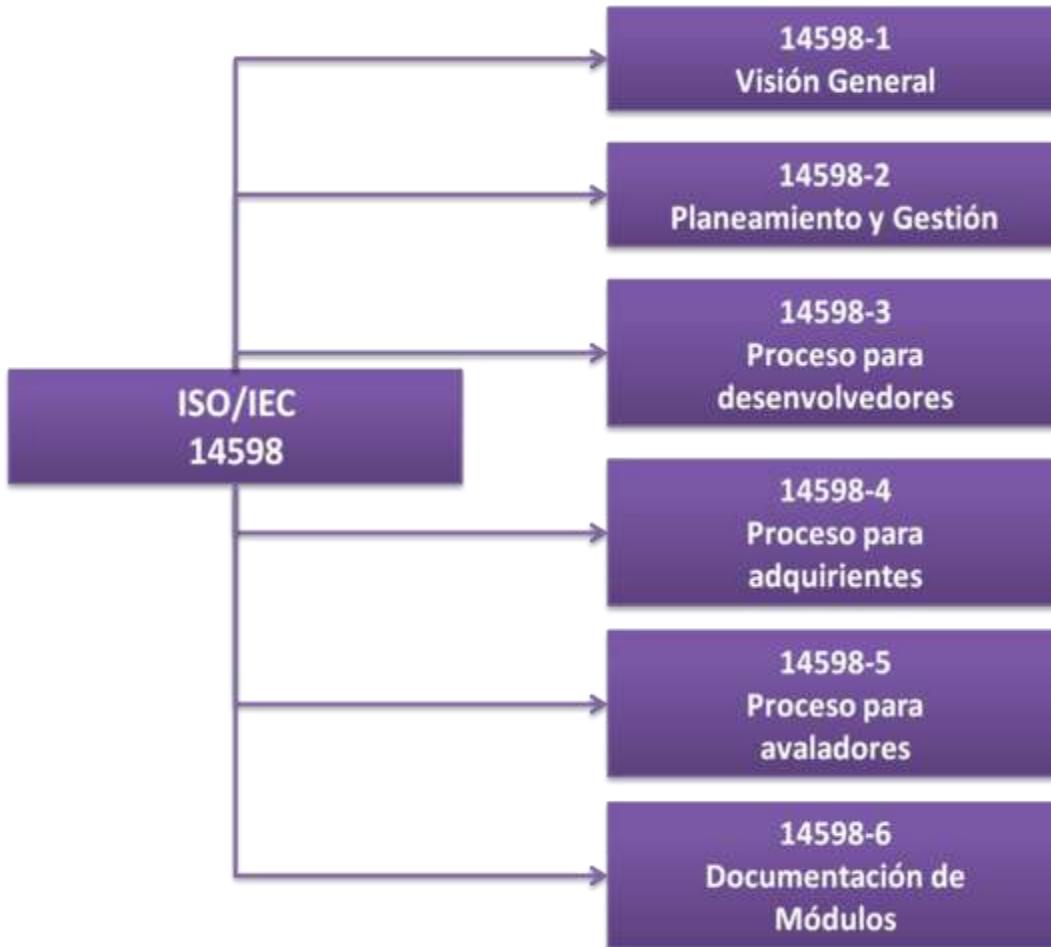


ISO 9126 distingue entre fallo y no conformidad. Un fallo es el incumplimiento de los requisitos previos, mientras que la no conformidad es el incumplimiento de los requisitos especificados. Una distinción similar es la que se establece entre validación y verificación.





La norma ISO/IEC 14598, en sus diferentes etapas, establece un marco de trabajo para evaluar la calidad de los productos de software proporcionando, además, métricas y requisitos para los procesos de evaluación de los mismos.



Existen multitud de trabajos basados en las normas ISO 9126 e ISO 14598 que puede ser de interés a la hora de plantearse la evaluación de productos software. Citamos algunos de los más relevantes:

- SQUID, permite la especificación, planificación, evaluación y control de la calidad de software a través de los procesos de desarrollo. La calidad queda definida como un comportamiento operacional de los productos requeridos por sus usuarios. Ofrece un método y una herramienta de soporte que reciben el nombre de SQUID.



- QUINT2, amplia el modelo de la ISO 9126 para evaluar la calidad de arquitecturas software.
- COTS (commercial off-the-shelf), refinamiento del modelo de calidad de la ISO 9126.

- Esta metodología consta de siete pasos:
 1. Definir el dominio.
 2. Determinar subcaracterísticas de calidad.
 3. Definir una jerarquía de subcaracterísticas.
 4. Descomponer subcaracterísticas en atributos.
 5. Descomponer atributos derivados (aquellos que no sean medibles directamente) en atributos básicos.
 6. Establecer relaciones entre entidades de calidad (por ejemplo, aumentar la subcaracterísticas de seguridad lleva consigo que aumente la madurez de un producto)
 7. Determinar métricas para los atributos.

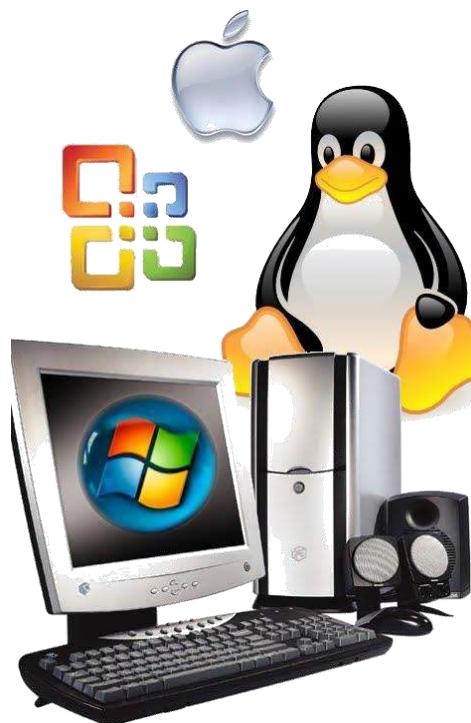


- Simlo y Belchior (2003). Amplían las subcaracterísticas y atributos propuestos por la norma ISO 9126 llegando a identificar 124 atributos de calidad para los componentes software.
- Moraga et al., (2005) proponen un modelo de calidad para **portlet**s basada en la adaptación de ISO 9126 así como en algunos de los trabajos anteriormente citados



Relaciones y proceso de transición entre las series ISO/IEC 9126 e ISO/IEC 14598 a la serie de normas SQuaRE

ACTUALMENTE		SQuaRE
9126: Calidad del producto		2500n: Gestión de calidad
1 – Modelo de calidad		25000 – Guía de SQuaRE
2 – Métricas externas		25001 – Planificación y Gestión
3 – Métricas internas		2501n: Modelo de Calidad
4 – Métricas de calidad en uso		25010 – Guía y modelo de calidad
		2502n: Medidas de Calidad
		25020 – Guía y modelo de referencia
		25021 – Medidas primitivas
		25022 – Medidas internas
		25023 – Medidas externas
		25024 – Medidas de calidad en uso
		2503n: Requisitos de Calidad
		25030 – Guía y Requisitos de Calidad
14598: Evaluación del producto		2504n: Evaluación de Calidad
1 – Resumen general		25040 – Guía y resumen de evaluación
2 – Planificación y gestión		25041 – Módulos de evaluación
3 – Proceso para desarrolladores		25042 – Proceso para desarrolladores
4 – Proceso para compradores		25043 – Proceso para compradores
5 – Proceso para evaluadores		25044 – Proceso para evaluadores
6 – Módulos de evaluación		



Lecturas Recomendadas

❖ CALIDAD DE SISTEMAS INFORMÁTICOS

<http://es.scribd.com/doc/95955163/Calidad-en-Sistemas-Informaticos>

❖ MODELO DE LA CALIDAD

<http://www.mginformatica.com.ar/modelo-de-calidad.htm>

Actividades y Ejercicios



1. En un documento en Word elabore un cuadro comparativo sobre la calidad del modelo de McCall y el modelo propuesto en la norma ISO 9126, indique cuál le parece más completo, y a qué elementos de la calidad le concedería más importancia.

Envíalo a través de "**Calidad de los Modelos**".

2. En un documento en Word defina un proceso (P.) de selección (S.) para herramientas (H.) de análisis (A.) y diseño (D.) orientado (O.) a objetos (O.), adaptando si fuera necesario el modelo de calidad de la ISO 9126. Tomando como base el proceso de selección propuesto por la norma ISO 14598.

Envíalo a través de "**P. S. H. A. D. O. O.**".

Autoevaluación

- 1) Uno de los modelos clásicos en el que la calidad de un producto software se descompone en once características o factores de calidad agrupados en tres categorías: Operación de producto, revisión de producto y transición de producto, es el modelo propuesto por:**
 - a. McCall, 1977.
 - b. Gallin, 2004.
 - c. Evans 1987.
 - d. Caswell 1987.
 - e. Grady 1990.
- 2) Son características del modelo de calidad interna y externa:**
 - a. Funcionalidad y comprobación.
 - b. Funcionalidad y planificación.
 - c. Funcionalidad y codificación.
 - d. Fiabilidad y eficiencia.
 - e. Funcionalidad y evacuación.
- 3) Es la capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados:**
 - a. Codificación.
 - b. Exposición.
 - c. Seguridad.
 - d. Fiabilidad.
 - e. Adecuación.
- 4) La mantenibilidad es la capacidad de ser:**
 - a. Modificado.
 - b. Aprobado.
 - c. Constituido.
 - d. Proyectado.
 - e. Analizado.
- 5) La norma ISO 9126 está relacionado con:**
 - a. La calidad de software.
 - b. La calidad de uso.
 - c. La calidad de proceso.
 - d. La calidad individual.
 - e. La calidad estándar.

- 6) Es una característica de la calidad de uso:**
- a. Portabilidad.
 - b. Utilidad.
 - c. Efectividad.
 - d. Fiabilidad.
 - e. Productividad.
- 7) El ISO 9126 es un estándar internacional para _____ de la calidad del software.**
- a. La proyección.
 - b. La codificación.
 - c. La renovación.
 - d. La evaluación.
 - e. La corrección.
- 8) La norma que establece un marco de trabajo para evaluar la calidad de los productos de software proporcionando métricas es :**
- a. La ISO/IEC 15597.
 - b. La ISO/IEC 13596.
 - c. La ISO/IEC 12595.
 - d. La ISO/IEC 16593.
 - e. La ISO/IEC 14598.
- 9) Son componentes de la calidad del sistema de información:**
- a. Tecnología de desarrollo y calidad de estrategia.
 - b. Calidad de gestión y calidad de servicio.
 - c. Calidad del personal y calidad de software.
 - d. Tecnología de desarrollo y calidad de evaluación.
 - e. Tecnología de recursos y calidad de software.
- 10) La familia de normas ISO 25000 (ISO 2005a-n) es conocida con el nombre de:**
- a. Quality.
 - b. Secure.
 - c. Square.
 - d. Sunthuar.
 - e. Caswell.

Resumen

UNIDAD DE APRENDIZAJE II: CALIDAD DE LOS SISTEMAS INFORMÁTICOS

Uno de los modelos clásicos más utilizados desde su creación, incluso con vigencia en nuestros días, es el desarrollado por McCall (McCall et al., 1977), en el que la calidad de un producto software se descompone en once características o factores de calidad agrupados en tres categorías: Operación de producto, Revisión de producto y transición de producto.

A finales de los años ochenta, fueron propuestos dos modelos alternativos a los de McCall basados igualmente en la identificación de factores: el modelo de factores de Evans y Marciniak (1987) y el modelo de factores de Deutsch y Willis (1988).

El modelo de calidad para calidad interna y externa categoriza los atributos de calidad software en seis características: Funcionalidad (adecuación, exactitud, interoperabilidad,...), Fiabilidad (madurez, tolerancia de fallos...), Usabilidad, Eficiencia, Mantenibilidad y Portabilidad. El modelo para la calidad externa e interna está indicado con la norma ISO, 2001.

La norma ISO 9126 entiende por calidad en uso "la capacidad del producto software para permitir a determinados usuarios alcanzar' objetivos especificados con efectividad, productividad, seguridad y satisfacción, en contextos de uso especificados". La norma ISO 14598 da una visión general del proceso de evaluación de un producto software, explicando en sus diferentes partes como aplicar el proceso en diferentes circunstancias. Esta norma se apoya en la ISO 9126 ya que los aspectos cuantificables pueden medirse cuantitativamente usando métricas de calidad, cuyo valor medido se sima en una escala.

ISO 9126 es un estándar internacional para la evaluación de la calidad del software. Está reemplazado por el proyecto SQuaRE, ISO 25000:2005, el cual sigue los mismos conceptos. Un producto software está definido en un sentido amplio como: los ejecutables, código fuente, descripciones de arquitectura, y así, como resultado, la noción de usuario se amplía tanto a operadores como a programadores, los cuales son usuarios de componentes como son bibliotecas software.



UNIDAD 3



Calidad del Proceso Software

Introducción

a) Presentación y contextualización

Tradicionalmente la Ingeniería del Software se ha centrado en metodologías y lenguajes de programación, modelos de desarrollo y herramientas. Sin embargo, y teniendo en cuenta la creciente complejidad de los sistemas, se hacía necesario incluir determinadas áreas que hoy en día son críticas para la ingeniería del software, como las infraestructuras de gestión y organización, por lo que surge la denominada ingeniería del software basada en el proceso.

b) Competencia

Describe los diferentes modelos de calidad del proceso software y sus características.

c) Capacidades

1. Analiza los procesos básicos de un proceso de software.
2. Aplica diversos métodos en el control de procesos de software.
3. Comprende los diferentes entornos de ingeniería del software orientados al proceso.
4. Reconoce los diversos procesos en el ciclo de vida del software.

d) Actitudes

- ✓ Participa activamente en el desarrollo de las tareas de proceso de software.
- ✓ Cumple con rigurosidad las actividades relacionados con los diversos métodos de control de proceso de software.

e) Presentación de Ideas básicas y contenido esenciales de la unidad:

La Unidad de Aprendizaje 03: Calidad del Proceso Software, comprende el desarrollo de los siguientes temas:

TEMA 01: El Proceso Software.

TEMA 02: Modelado de Procesos Software.

TEMA 03: Entorno PSEE.

TEMA 04: Ciclo de Vida.

TEMA 1

El Proceso Software



Competencia:

Analizar los procesos básicos de un proceso de software.

The image shows a woman in a professional setting, smiling and working on a computer. Overlaid on the screen are several software interface windows displaying data analysis and reporting. These include:

- A menu tree on the left side of the screen.
- An "ENTRADA/MODIFICACIÓN PROYECTO" window showing project details like Project ID, Client Reference, and Information Address.
- A "Acceso a Compañía para el Año Especificado" window showing company information and access details.
- A line graph titled "Ingreso Vs. Ganancia" comparing Bruto and Neto prices.
- A table titled "Márgenes por Categoría de Producto" showing margins for various product categories.
- A pie chart titled "% Cumplimiento de Cuota" showing completion percentages.
- A pie chart titled "% Ventas por Área" showing sales distribution across different areas.

Desarrollo de los Temas



Tema 01: El Proceso Software

¿QUÉ ES UN PROCESO?

Un proceso se define como un conjunto de actividades interrelacionadas que se transforman en entradas y en salidas (ISO, 1995). Un proceso define quien está haciendo que, cuando, y como alcanzar un determinado objetivo.

El proceso software, algunas definiciones:

El conjunto de actividades, métodos, prácticas y transformaciones que la gente usa para desarrollar, mantener software y los productos de trabajo asociados (planes de proyecto, diseño de documentos, código, pruebas y manuales de usuario) (SEI, 1995). El proceso o conjunto de procesos usados por una organización o proyecto, para planificar, gestionar, ejecutar, monitorizar, controlar y mejorar sus actividades software relacionadas" (ISO, 1998). El proceso software define como se organiza, gestiona, mide, soporta y mejora el desarrollo, independientemente de las técnicas y métodos usados. El proceso software es un proceso con una naturaleza especial, determinada por las siguientes características (Demame et al., 1999)



El proceso software es un proceso con una naturaleza especial, determinada por las siguientes características (Demame et al., 1999):

- Es complejo.
- No es un proceso de producción típico; ya que está dirigido por excepciones, se ve muy determinado por circunstancias impredecibles, y cada uno tiene peculiaridades que lo distingue de los demás. Tampoco es un proceso de ingeniería "pura"; ya que se desconocen las abstracciones adecuadas, depende en gran medida de demasiada gente, el diseño y la producción no están claramente diferenciados, y los presupuestos, calendarios y calidad no pueden ser planificados de forma suficientemente fiable.

- No es (completamente) un proceso creativo; ya que algunas partes pueden ser descritas en detalle y algunos procedimientos son impuestos previamente.
- Está basado en descubrimientos que dependen de la comunicación, coordinación y cooperación dentro de marcos de trabajo predefinidos: los entregables generan nuevos requisitos; los costes del cambio del software no suelen reconocerse; y el éxito depende de la implicación del usuario y de la coordinación de muchos roles (ventas, desarrollo técnico, cliente, etc.).

La necesidad de participación humana de forma creativa y la ausencia de acciones repetitivas hacen que ni el desarrollo ni el mantenimiento del software sean procesos de fabricación, pero existen algunas similitudes entre ambos tipos de procesos que son útiles para comprender los procesos software con una perspectiva más amplia. Al igual que los procesos de fabricación, los procesos software constan de dos subprocessos interrelacionados.

Proceso de Producción: Relacionado con la construcción y mantenimiento del producto software.

Proceso de Gestión: Que es el encargado de estimar, planificar y controlar los recursos necesarios (personas, tiempo, tecnología, etc.) para poder llevar a cabo y poder controlar el proceso de producción. Este control genera información sobre el proceso de producción, que puede ser usada posteriormente para mejorar el proceso y por tanto, mejorar la calidad del producto software.



Por lo tanto, el proceso software es un campo de estudio amplio y complejo en el mundo de la ingeniería del software, en el que debido a la gran cantidad y diversidad de elementos relacionados, se podrían establecer en las siguientes categorías (Fuggetta, 2000):

- **Tecnología de Desarrollo Software:** relacionado con el soporte tecnológico, en forma de herramientas, infraestructuras y entornos.
- **Métodos y Técnicas de Desarrollo Software:** que constituyen una guía sobre cómo se deben hacer las cosas: uso de la tecnología y realización de las actividades.

- **Comportamiento Organizacional:** relacionado con los recursos humanos. Los procesos software son llevados a cabo por equipos de personas que tienen que estar coordinados y deben gestionarse desde una eficiente estructura organizacional.
- **Economía y Marketing:** relacionado con la gestión de proyectos, debido a que el producto software final debe cumplir con unos plazos y costes determinados y debe satisfacer las necesidades del cliente al que va destinado.



La integración de las tecnologías de producción y de gestión en un entorno de trabajo constituye la esencia de la Tecnología del Proceso Software y como resultado se han desarrollado los denominados Entornos de Ingeniería del Software orientados al Proceso (PSEE, Process-Centered Software Engineering Environment).

A pesar de su importancia y de los avances en la investigación en estos temas, muy pocas propuestas de PSEE han sido aplicadas de forma práctica en la industria. Tal y como se ha comentado las razones son variadas, desde la rigidez de muchas de las propuestas que ha dificultado su aplicación industrial en mercados dinámicos, a la necesidad de introducir este tipo de entornos poco a poco, permitiendo de un modelado descriptivo de los procesos que ayude a su entendimiento y comunicación, para posteriormente añadir los detalles necesarios para proporcionar soporte a su ejecución.

Sin embargo y a pesar de que el tema de los procesos software no se ha establecido tal como una disciplina que se enseñe y practique universalmente por la industria del software es de esperar que en el futuro las tecnologías de soporte a los procesos software maduren y sean adoptadas por las organizaciones.



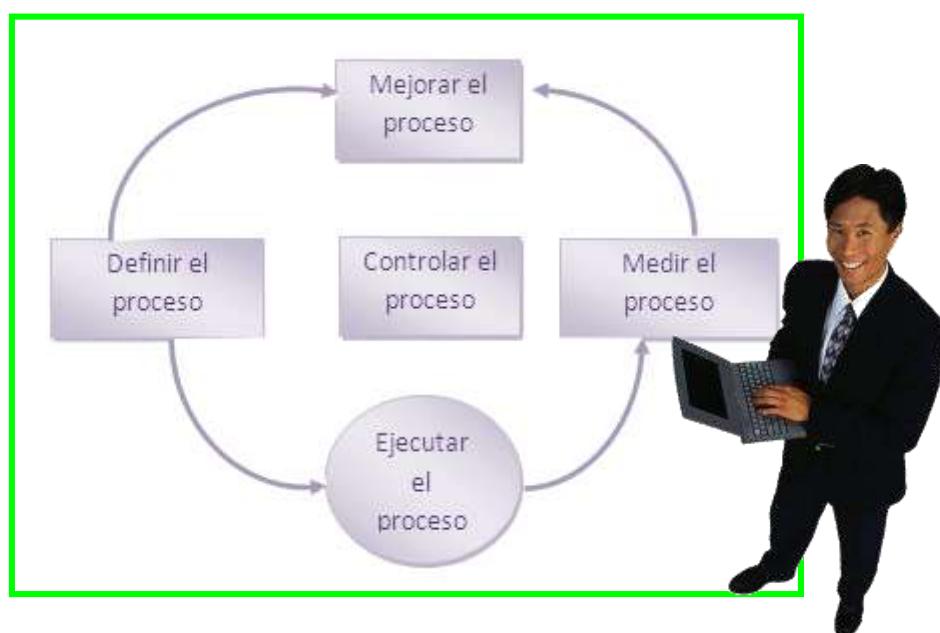
GESTIÓN DE LOS PROCESOS SOFTWARE



Los requisitos de calidad más significativos de los procesos software son: (1) que produzcan los resultados esperados, (2) que estén basados en una correcta definición y (3) que sean mejorados en función de los objetivos de negocio, muy cambiantes ante la gran competitividad de las empresas hoy en día, estos son los objetivos de la Gestión del Proceso Software. Para aplicar esta gestión de forma efectiva es necesario asumir cuatro responsabilidades clave: Definir, Medir, Controlar y Mejorar el Proceso.

Estas responsabilidades y sus relaciones se representan de acuerdo a estas responsabilidades, para llevar a cabo de una forma eficiente la mejora del proceso es necesario tener en cuenta los siguientes aspectos:

- **Definición del Proceso:** La definición del proceso es la primera responsabilidad clave que hay que asumir para poder realizar una gestión efectiva de los mismos. Para ello, es necesario modelar los procesos, es decir, representar los elementos de interés que intervienen. El modelado de los procesos software, por lo tanto, constituye un paso fundamental para la comprensión y mejora continua de los procesos de una organización.



- **Ejecución y Control del Proceso:** los proyectos software de una empresa se llevan a cabo de acuerdo a los modelos de procesos definidos. En este sentido, es importante poder controlar en todo momento la ejecución de estos proyectos (y en consecuencia, de los procesos correspondientes) para garantizar que se obtienen los resultados esperados.

Para ello, se han desarrollado en las dos últimas décadas los denominados "Entornos de Ingeniería del Software orientados a Procesos" (PSEE), que son los sistemas software que ayudan en el modelado de los procesos software utilizando un determinado lenguaje y en su posterior automatización.

- **Medición y Mejora:** Existe una importante constelación entre la medición y la mejora de los procesos software. Antes de poder mejorar un proceso es necesario llevar a cabo una evaluación, cuyo objetivo es detectar los aspectos que se pueden mejorar. Para ello, es conveniente disponer de un marco de trabajo efectivo que facilite la identificación de las entidades relevantes candidatas a ser medidas. Con los resultados de la medición de los procesos es posible disponer de una información objetiva que permita planificar, identificar y llevar a cabo de una manera eficiente las acciones de mejora necesarias.

Modelado de los procesos software

Uno de los aspectos básicos y fundamentales para la tecnología de soporte a los procesos software es disponer de modelos de procesos que representen fielmente la forma de hacer las cosas de las organizaciones. En una empresa o en un dominio de aplicación, los procesos de diferentes proyectos tienden a seguir patrones comunes, bien porque las "mejores prácticas" son reconocidas formalmente, bien por la existencia de estándares utilizados. Por lo tanto se hace necesario intentar capturar estos aspectos comunes en una representación de proceso, la cual describe estas características comunes y fomenta la homogeneidad y la unificación de criterios.



Por lo tanto, uno de los grandes objetivos de la tecnología de procesos es lograr que la representación de procesos pueda ser usada para gestionar los procesos actuales de desarrollo y mantenimiento del software. Como primer paso, la tecnología de procesos introduce la noción de modelo de procesos, que consiste en la descripción de un proceso expresándolo en un lenguaje de modelado de procesos adecuado.

Un modelo de procesos puede ser analizado, validado y simulado, si es ejecutable. En los modelos de procesos se puede describir de una forma precisa los diferentes aspectos relacionados con los procesos software, de forma que con diferentes modelos se puedan expresar las diferentes vistas de un proceso.

Los objetivos y beneficios que motivan la introducción de modelos de procesos son varios, destacando los siguientes:

- **Facilidad de entendimiento y comunicación**, lo que requiere que un modelo de procesos contenga suficiente información para su representación. Un modelo, como representación del proceso que es, puede ser usado para la formación del personal.
- **Soporte y control de la gestión del proceso.**



Provisión para la automatización orientada al rendimiento del proceso que requiere un entorno de desarrollo efectivo del software, proporcionando orientaciones, instrucciones y material de referencia al usuario.

Provisión para el soporte automático a la ejecución, para lo cual es necesario automatizar ciertas partes del proceso, dar soporte al trabajo en grupo, compilación de métricas y aseguramiento de la integridad del proceso.

- **Soporte a la mejora del proceso.**

En la literatura se pueden encontrar diversos lenguajes y formalismos de modelado, conocidos como "*Lenguajes de Modelado de Procesos*" (LMP), que tienen como objetivo representar de una forma precisa y no ambigua, los diferentes elementos relacionados con un proceso software. A continuación se describen los diferentes elementos relacionados con el modelado de procesos, para lo cual se abordan en primer lugar los diferentes conceptos comunes relacionados con el proceso software.

En el siguiente apartado se presentan diversos lenguajes o meta modelos para la definición y representación de modelos de procesos.

Elementos del proceso software

En general, se puede identificar una serie de conceptos básicos relacionados con los procesos software y que son comunes a los diferentes modelos de procesos.

- **Actividad.** Las actividades se encargan de generar o modificar un conjunto dado de artefactos; incorporan e implementan procedimientos, reglas y políticas. Además, una actividad es un concepto con un componente funcional fuerte ya que acarrea entradas, salidas, y resultados intermedios.
- **Producto.** El conjunto de artefactos a ser desarrollados, entregados y mantenidos en un proyecto es lo que se denomina producto.

- **Recurso.** Un recurso es un activo que una actividad necesita para llevarse a cabo. En este campo, hay dos recursos de principal importancia: por un lado los desarrolladores (los agentes humanos en el proceso), y por otro, las herramientas de desarrollo (los agentes computarizados que tradicionalmente han sido usados en desarrollo del software como editores especializados y herramientas para la gestión, compiladores, etc.) y las herramientas de propósito general (como hojas de cálculo, editores de diagramas, etc. Que pueden ser usados para manejar el proceso).



- **Roles y Directivas.** Normalmente, las herramientas están fuertemente unidas a las actividades en las que son usadas, mientras que los desarrolladores se relacionan indirectamente a una actividad por medio de sus roles, es decir, el conjunto de responsabilidades, obligaciones y tareas (por ejemplo diseñadores, jefes de proyecto, revisores, etc.). El carácter de la organización impacta en el proceso indirectamente por medio de roles, y también directamente por medio de directivas (políticas, reglas, y procedimientos) que gobiernan las actividades. Las directivas normalmente vienen en manuales, y por lo tanto deberían ser estructuradas.

ELEMENTOS BÁSICOS DE UN MODELO DE PROCESO

Clasificación de los Lenguajes de Modelado de Procesos (LMP) Existen diferentes criterios para la clasificación de los lenguajes de modelado de procesos. Los procesos pueden ser modelados en diferentes niveles de abstracción y con diferentes objetivos.

La información de un modelo de procesos se puede estructurar bajo diferentes puntos de vista:

- ✓ **Funcional:** que representa que elementos del proceso se están implementando y que flujos de información son importantes para los elementos básicos del proceso.
- ✓ **Comportamental:** que representa cuando y bajo qué condiciones se implementan los elementos del proceso.
- ✓ **Organizacional:** que representa donde y por qué persona de la organización son implementados los elementos del proceso.
- ✓ **Informativo:** que representa las entidades de información de salida o manipuladas por un proceso, incluyendo su estructura y sus relaciones. Los diferentes lenguajes de modelado de procesos, proporcionan la notación necesaria para representar los procesos software y dicha representación puede incluir las clases de información comentadas anteriormente.



Modelado de Procesos Software

TEMA 2



Competencia:

Aplicar diversos métodos en el control de procesos de software.





Tema 02: Modelado de Procesos Software



DIAGRAMAS DE GANTT

Los diagramas de Gantt fueron creados por Hemy Gantt en el año 1917. Representan las diferentes actividades de un proceso como barras sobre un calendario aportando una representación visual de las actividades, su duración y su planificación.

Ejemplo de un diagrama de Gantt

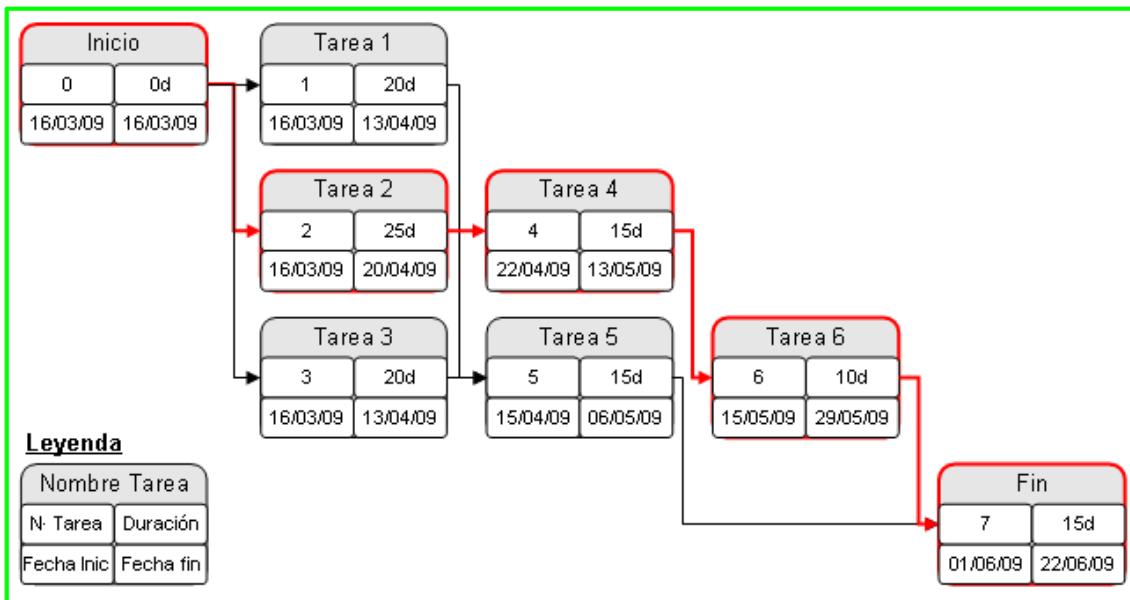
ACTIVIDADES	TIEMPO DE DURACION.											
	ABRIL				MAYO				JUNIO			
	1	2	3	4	1	2	3	4	1	2	3	4
Programar jornadas de alfabetización tecnológica a usuarios.	■	■										
Verificar el estado de los equipos informáticos.		■	■									
Gestionar recursos para el mantenimiento y reparación de las P.C.				■								
Realizar mantenimiento a las P.C.					■	■	■					
Facilitar talleres a usuarios tecnológicos de la Institución.								■	■			
Jornada de cierre de proyecto en la Institución.										■	■	

DIAGRAMAS PERT

Los diagramas PERT (Program Evaluation and Review Technique) representan gráficamente los procesos mediante un grafo dirigido en el que se incluyen las tareas, su duración y sus relaciones de precedencia. Son más difíciles de leer que un diagrama de Gantt, pero a su vez permiten un análisis más complejo del proceso, como la identificación de caminos críticos.



Ejemplo de un diagrama de Pert



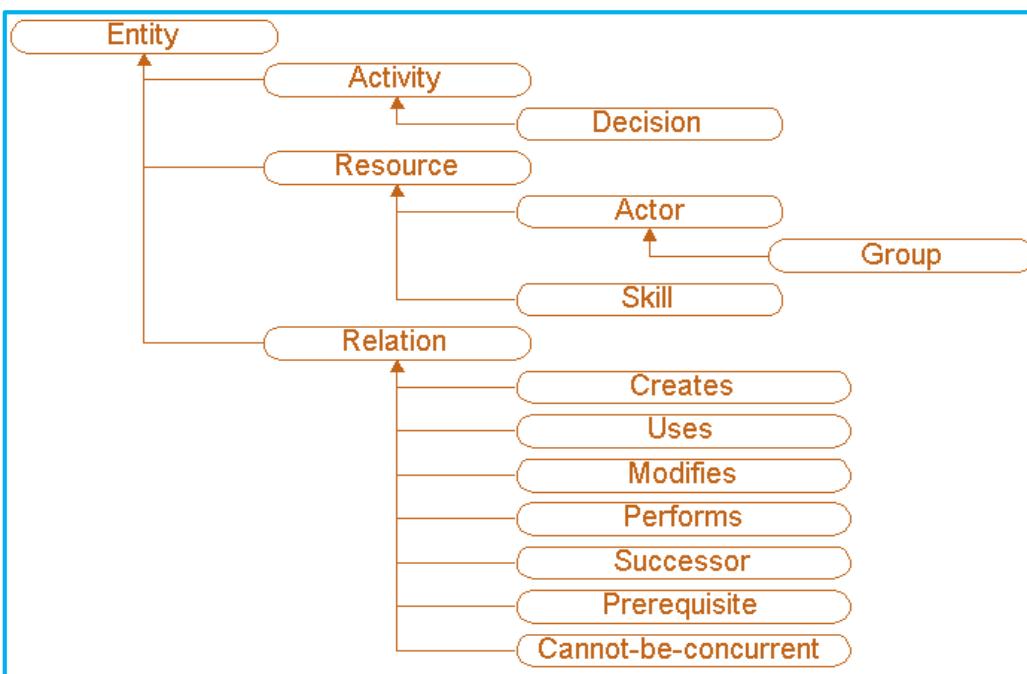
FORMATO DE INTERCAMBIO DE PROCESOS

El formato de intercambio de procesos (Process Interchange Format, PIF) (Lee et al., 1998), surge debido a la necesidad de diversas organizaciones de compartir sus modelos de procesos. El concepto de proceso en PIF es "un conjunto de actividades con ciertas relaciones entre si y con objetos determinados Instantes de Tiempo".



Los principales elementos. De acuerdo al metamodelo PIF son: la entidad **Actividad** (*Activity*) que se define como "cualquier cosa que ocurre en el tiempo", como por ejemplo un proceso, una tarea, o incluso un evento; la entidad **Relación** (*Relation*), que representa la relación entre dos entidades y puede ser de varios tipos; los **instantes de tiempo** (*timepoints*) que pueden ser una hora precisa, o un punto del tiempo en el que puede ocurrir un evento.

Y finalmente la entidad Objeto (Object), que representa todas aquellas entidades implicadas en un proceso, como artefactos, herramientas y agentes.



LENGUAJE DE ESPECIFICACIÓN DE PROCESOS (PSL)

El lenguaje de especificación de procesos (*Process Specification Language, PSL*) (Schlenoff et al., 1998) PSL define un proceso como "un conjunto de actividades en las que participan algunos objetos en un instante de tiempo determinado".



En PSL los tres constructores principales, a partir de los cuales se derivan la mayoría de los elementos de los modelos de procesos, son: actividad (*activity*), objeto (*object*) e instante de tiempo (*timepoint*).

En el núcleo del metamodelo PSL se define la base mínima necesaria para la representación de modelos de procesos. Esta base mínima es extendida mediante módulos de extensión, como por ejemplo extensiones para el ordenamiento temporal sobre instantes de tiempo.

El lenguaje PSL define de una forma muy precisa y no ambigua modelos de procesos mediante axiomas o definiciones usando KIF (Knowledge Interchange Format). Algunas de estas especificaciones se pueden expresar en el metamodelo, pero otras necesitan de mecanismos más potentes como OCL (Object Constraint Language).

Modelo del proceso unificado

El modelo del proceso unificado (*Unified Process Model*, UPM) (Kruchten, 1999) es una propuesta conjunta de organizaciones como IBM, Rational, Unisys, etc. Este metamodelo de procesos se ha usado para definir el "Proceso Unificado de Rational", un modelo de procesos de ingeniería del software comercializado por Rational Software.

El metamodelo UPM incluye los siguientes paquetes:

- **Nombres** (*names*): En el que se definen los mecanismos de nombrado
- **Elementos Basicos** (*Basic Elements*): define los elementos básicos, que son refinados en otros paquetes.
- **Estructura del Proceso** (*Process Structure*), define los principales conceptos del proceso, como artefactos (*artifacts*), roles (*roles*), o productos de trabajo (*work items*).
- **Guia** (*Guidance*), define como debería documentarse cada componente del proceso.
- **Componentes del Proceso** (*Process Components*), define mecanismos de empaquetamiento.

Core Plan Representation (CPR) (Pease, 1998), es un metamodelo patrocinado por la agenda DARPA (*Defense Advanced Research Project Agency*) y se concentra en la planificación (especificación de un conjunto de acciones para dar soporte a un conjunto de metas u objetivos) y la planificación (especificación de las cantidades de recursos usadas a lo largo del tiempo y el tiempo en que dichas acciones tendrán lugar).

La base del metamodelo CPR es la definición de planes.

Un plan (*plan*) es un conjunto de acciones (*actions*) necesarias para satisfacer determinados objetivos. En la realización de una acción un actor puede utilizar ciertos recursos. El actor de una acción puede ser el recurso de otra acción.

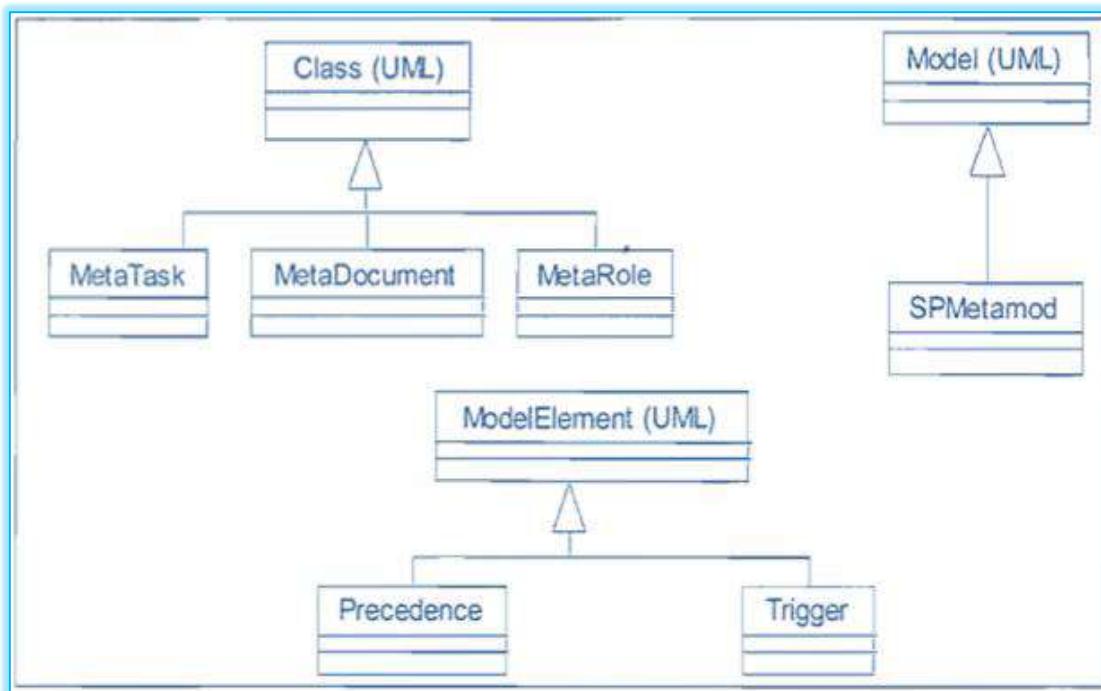


A partir del metamodelo CPR es posible modelar el diseño de un plan, pero no es posible almacenar la información sobre como este plan está siendo llevado a cabo en la realidad. Para dar soporte a la ejecución del proceso se ha añadido el concepto de **Modelo del Mundo** (WorldModel). Un plan de ejecución se estructura como un plan de diseño pero se usa para registrar los resultados de la ejecución del plan de diseño.

Promenade

PROMENADE (Franch y RibO, 1999; 2003) es un lenguaje para la modelización de procesos software que utiliza UML para describir sus constructores, mediante la generación de un profile.

Elementos básicos de PROMENADE



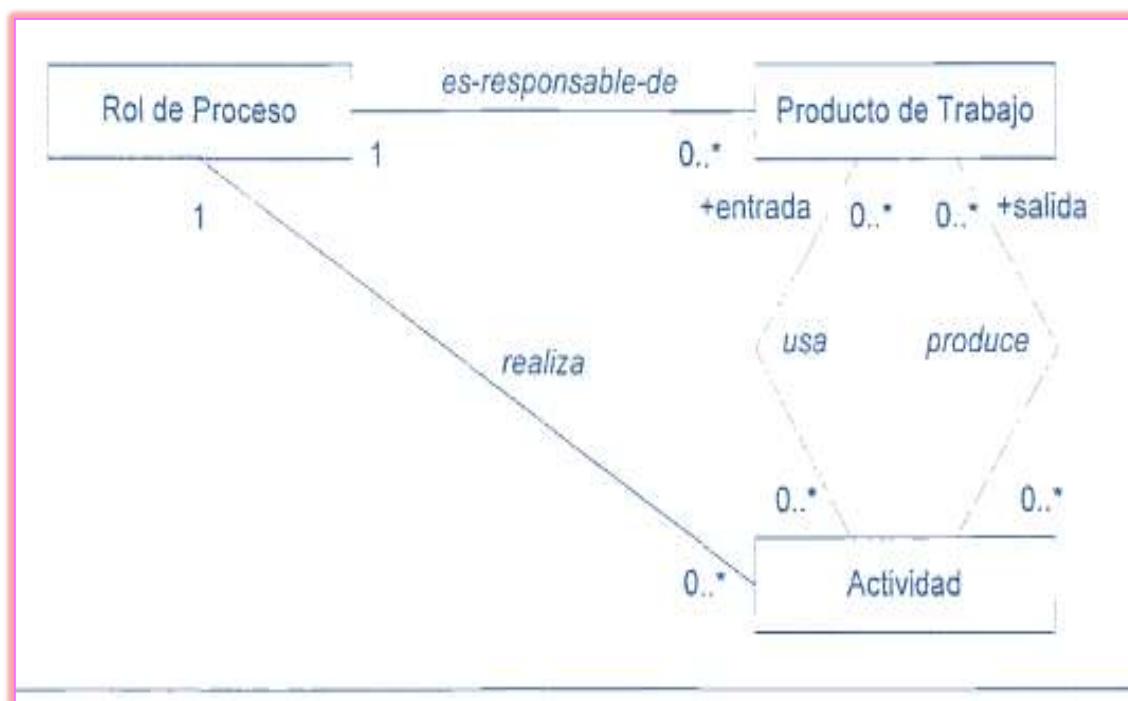
Las clases que componen el núcleo de PROMENADE son: meta Documento que son consideradas como especializaciones del constructor Clase de UML. Estas tres clases son los constructores cuyas instancias caracterizan un modelo de procesos software.

Spem

SPEM (*Software Process Engineering Metamodel*) es una especificación de OMG (2002). SPEM describe un metamodelo genérico para la descripción de procesos software concreto. Está basado en MOF y utiliza UML como notación de modelado. Por tanto, se basa en los principios de orientación a objetos. En esta propuesta no se da soporte a la ejecución (*enactment*) de los procesos, es decir, la planificación y ejecución de proyectos usando un modelo de proceso descrito con SPEM.



Modelo conceptual básico de SPEM



SPEM especifica el conjunto mínimo de elementos necesarios para describir cualquier proceso software concreto, sin incluir constructores para áreas o disciplinas específicas; de forma que en SPEM se describe un metamodelo genérico.

SMSDM

El metamodelo SMSDM (*Standard Metamodel for Software Development Methodologies*) (Henderson-Sellers y Gonzalez-Perez, 2004; Standards Australia, 2004) establece un marco de trabajo para la definición y extensión de metodologías de desarrollo de software. Incluyendo sus tres aspectos principales: el proceso a seguir, los productos utilizados y generados y las personas implicadas.



TEMA 3

Entorno PSEE



Competencia:

Comprender los diferentes entornos de ingeniería del software orientados al proceso.





Tema 03: Entorno PSEE

ENTORNOS PSEE



Los Entornos de Ingeniería del Software Orientados al Proceso (PSEE), dan soporte a los procesos de ingeniería, usados para concebir, diseñar, desarrollar y mantener un producto software.

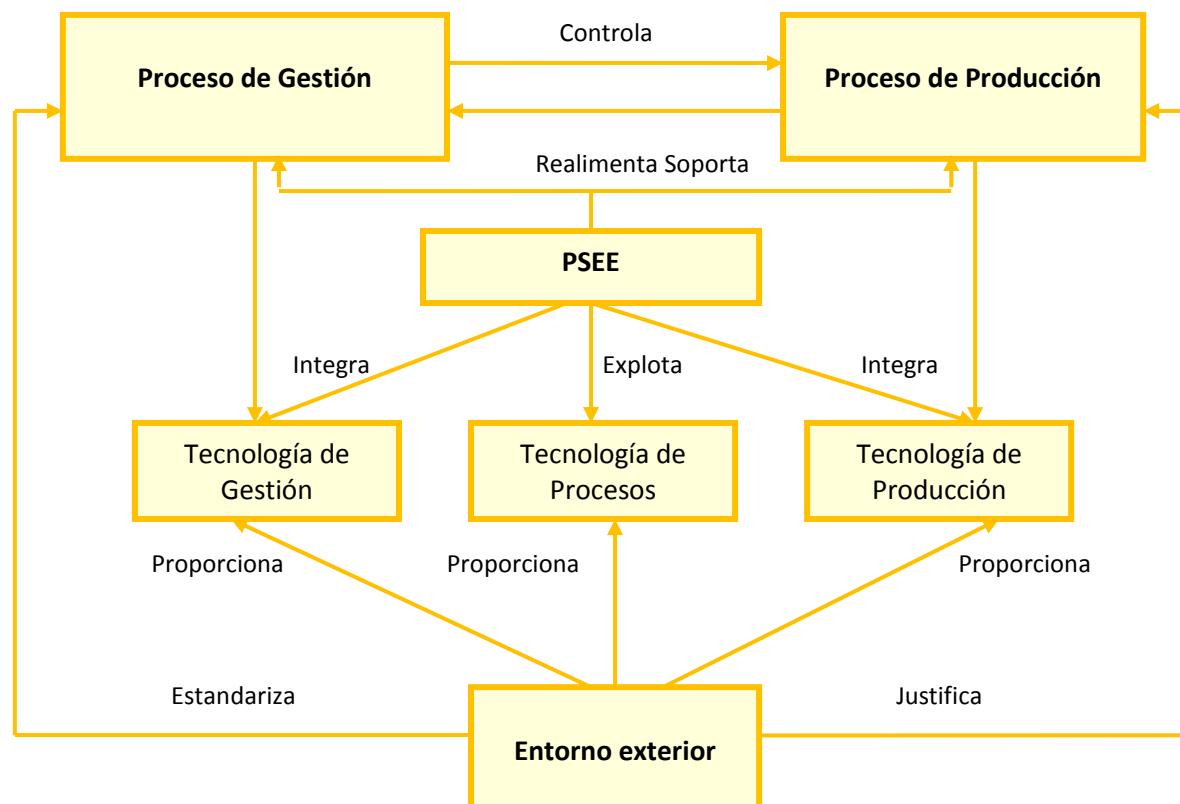
Los modelos asociados a un PSEE especifican como las personas deben interactuar, trabajar, y también cómo y cuándo las herramientas utilizadas en el proceso deben ser utilizadas y/o activadas automáticamente.

Un elemento clave del entorno constituye el **motor del proceso** (*process engine*) que es el encargado de guiar y ayudar a las personas a la hora de llevar a cabo las distintas actividades del proceso, y automatiza la ejecución de las actividades que no requieren intervención humana. El motor de procesos está constituido por los siguientes elementos:

- Un **Intérprete** del **Modelo de Procesos**, ejecuta el modelo controlando las herramientas usadas durante el proceso, guiando a las personas participantes y verificando que se satisfacen las restricciones especificadas en el modelo (como por ejemplo el orden de ejecución de ciertas actividades).
- Un **Entorno de Interacción del Usuario**, constituido por las herramientas que utilizan los usuarios, como pueden ser editores, compiladores, agendas, herramientas de gestión de proyectos, etc. Estas herramientas son controladas por el intérprete, que las utiliza para recibir realimentación de los usuarios y darles soporte durante el proceso.

- Un **Repositorio**, gestiona la información que es persistente en el entorno. Almacena los artefactos producidos durante el proceso y que son gestionados por el entorno, como pueden ser archivos de código fuente, documentación, ejecutables, casos de prueba, informes, etc. También se incluye toda la información del estado actual del proceso que está siendo ejecutado.

El Entorno PSEE Entorno de ingeniería del software orientado al proceso



Basado en los elementos anteriores se ha desarrollado un modelo de referencia y una propuesta arquitectal para entornos PSEE en general. De acuerdo a este modelo de referencia, un PSEE está controlado por un **motor de procesos**, cuyo objetivo es controlar el flujo de información entre los desarrolladores de acuerdo al modelo de procesos. El modelo es almacenado en un **repositorio**, junto con la definición del producto e información relevante sobre el estado del proceso. Además del repositorio, existen otro nivel de memoria importante formado por los **espacios de trabajo**, que son conjuntos de recursos informáticos que los desarrolladores utilizan cuando desempeñan un determinado rol en cierta actividad o tarea.

Un PSEE también tiene que tener la capacidad para compartir datos con el exterior mediante canales de importación y exportación, que permitan el intercambio de productos y modelos en un formato de comunicación reconocible.

La línea discontinua desde el motor de procesos a la capa de comunicación indica que el motor de procesos controla el PSEE esencialmente controlando el flujo de información entre el repositorio y los espacios de trabajo, entre unos espacios de trabajo y otros, y entre los usuarios y sus espacios de trabajos.

Clasificación de los PSEE

En primer lugar cabe destacar que todo PSEE está caracterizado por el lenguaje de modelado de procesos (LMP) que utiliza. Los LMPs utilizados pueden adoptar alguno de los siguientes cuatro enfoques: Lenguaje Basado en la Programación, Basados en Reglas, Basados en Autómatas Extendidos y Multiparadigma, que combinan dos o más paradigmas para describir los distintos aspectos de un proceso software.

SPADE (Bandinelli et al., 1993) es un ejemplo de este tipo, en el que su Estructura Principal está basada en Redes de Petri, proporciona un modelo de datos orientado a objetos para describir artefactos y utiliza un lenguaje operacional para describir las acciones asociadas a las sanciones definidas.

Otro de los aspectos clave de los PSEE es el tipo de soporte que ofrecen a los usuarios, distinguiéndose entre cuatro posibles tipos (Bandinelli et al., 1996):

- **Rol Pasivo.** El usuario guía el proceso y el PSEE opera en respuesta a las peticiones del usuario.
- **Guía Activa.** El PSEE guía el proceso y pregunta al usuario cuando es necesario, recordándoles en todo momento qué actividades deberán realizar. Los usuarios son libres para decidir y realizar las acciones sugeridas por el entorno.
- **Obligación.** El PSEE fuerza a los usuarios a actuar tal y como se ha especificado en el modelo de procesos.
- **Automatización.** El PSEE ejecuta las actividades sin intervención de los usuarios.



Un mismo PSEE puede adoptar distintas formas de soporte al usuario, como por ejemplo adoptar el enfoque de automatización para actividades que no requieren la intervención de los usuarios y el de obligación para el resto.



También es posible clasificar los PSEE en función de la forma de controlar y guiar el proceso. En este caso se distingue entre PSEE **Proactivos**, en los que el entorno inicia y controla las operaciones realizadas por las personas y **Reactivos** en los que el entorno queda subordinado a los usuarios.

A pesar de los grandes avances en la investigación de los PSEE, la gran mayoría no ha tenido la aceptación industrial esperada. Una de las causas más significativas ha sido el énfasis que los PSEE han dado a la descripción de modelos de procesos como modelos normativos cuyo seguimiento debía ser estricto. Por otro lado, muchas de las propuestas incluyen LMP muy complejos y poco intuitivos que ha dificultado su uso por los profesionales que prefieren lenguajes más intuitivos y que les facilite su comunicación y entendimiento del proceso (Fugetta, 2000).

Todo ello ha originado una importante reflexión en la comunidad investigadora siendo algunos requisitos y retos importantes para el futuro los siguientes:

- El PSEE debe dar **soporte dinámico a la ordenación de actividades**. Si la ordenación de las actividades puede ser elaborada y modificada dinámicamente, el motor de reificación del PSEE debe ser capaz de continuar apoyando y asistiendo durante la realización del proceso. Un aspecto clave son las interacciones de los humanos con el PSEE. El PML y el PSEE (como soporte del PML) deberían permitir cierta flexibilidad que les permita ser útiles dentro de la estrategia adoptada por una compañía, que puede ir desde un estricto y disciplinado proceso "dirigido por el plan" hasta un proceso completamente libre.
- El PSEE debe dar **soporte a la distribución de procesos software**, lo cual comprende la modularidad, heterogeneidad, interoperabilidad y componibilidad de procesos y la federación de fragmentos de proceso.

- También implica que el PSEE debe ser capaz de dar soporte a la comunicación, coordinación, cooperación y negociación entre los usuarios realizadores con sus diferentes roles.
- El PSEE debe dar soporte a la **evolución de procesos software**: tanto evolución "off-line" como "on-line". En este caso deben tenerse en cuenta las consecuencias en los procesos que están en curso y en los que ya han sobrepasado el punto de cambio en el modelo. Los PSEE también deben dar soporte a la evolución privada: el cambio será local a la instancia de modelo de proceso que se está ejecutando. Las desviaciones del proceso respecto del modelo deben ser soportadas y negociadas y su impacto debe ser gestionado.

Ejemplos de PSEE

En este apartado se ilustran las características de los PSEE mediante la presentación de algunos ejemplos representativos de la bibliografía.

Spade

El entorno SPADE (Bandinelli et al., 1993; 1995: 1996) es un PSEE diseñado en la Universidad Politécnica de Milán que proporciona soporte al análisis, diseño y ejecución de los procesos software. Para el modelado de los procesos utiliza el formalismo SLANG (SPADE Language), que es un LMP basado en una extensión de Redes de Petri a alto nivel. En SLANG un proceso se describe como una jerarquía de actividades.



Las actividades y estados de actividad en SLANG son representados como redes de Petri, mientras que los datos del proceso se representan como *tokens*. Un modelo de procesos en SLANG está compuesto básicamente por los siguientes elementos:

- **Process Types**, que es un conjunto de definiciones de tipos organizadas de forma jerárquica de acuerdo a un estilo OO. En SLANG todos los datos del proceso son caracterizados como tipos y organizados en jerarquías de herencia. El elemento raíz es el tipo Process Data, del que heredan el resto de tipos.

- **Process Activities**, es un conjunto de definiciones de actividad. Una definición de actividad es una Red de Petri a alto nivel donde se incluyen lugares (places), arcos (arcs) y transiciones (transitions). Cada lugar tiene un nombre y un tipo y se comporta como un repositorio que únicamente contiene tokens de su tipo o cualquiera de sus subtipos. Los lugares pueden cambiar sus contenidos como consecuencia de transiciones de disparo (fire transitions). Un caso particular son los lugares de interfaz de usuario, que se representan mediante círculos dobles y son lugares que pueden cambiar sus contenidos como consecuencia de la intervención humana.



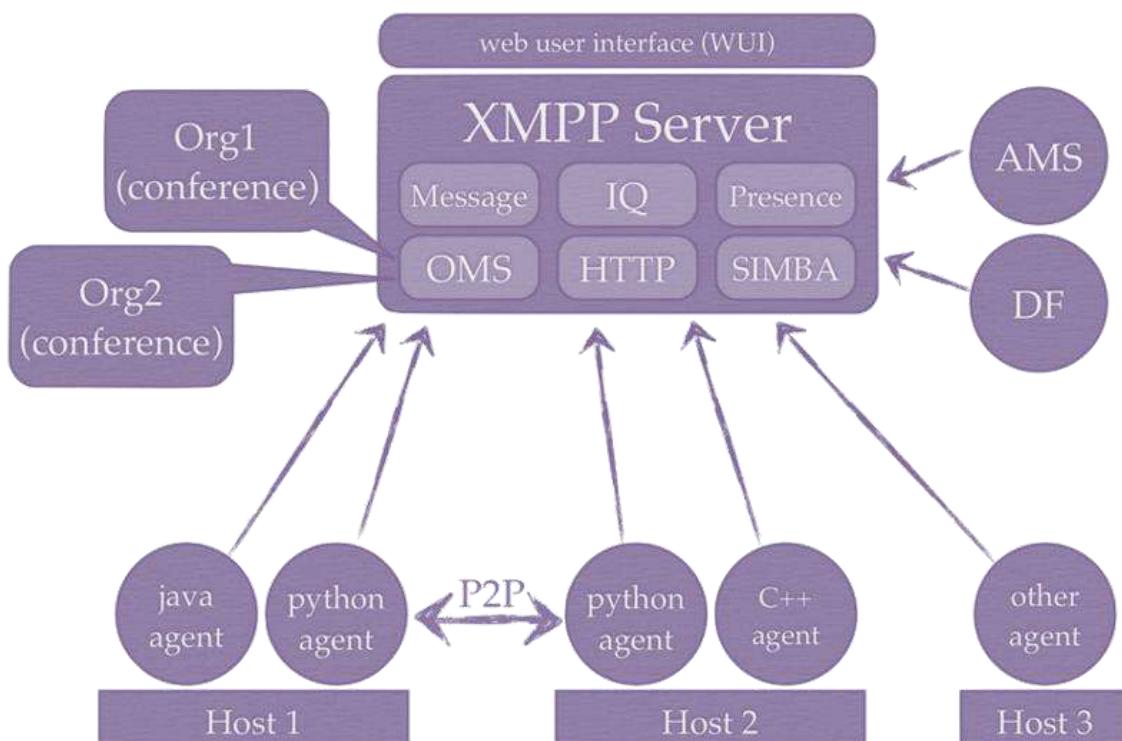
Apel

APEL (Dami et al., 1998; Estublier et al., 1998: 2003) es un PSEE desarrollado en el Laboratoire Logiciels, Systemes, Reseaux, en Francia. Los objetivos fundamentales que persigue se basan en dar soporte a la:

1. **Interoperabilidad** entre PSEE heterogéneos, permitiendo al desafiador del proceso construir una "federación" de PSEE capaces de gestionar procesos complejos y distribuidos.
2. **Evolución del Proceso**, con el fin de hacer frente a situaciones imprevistas durante la ejecución.

A/R	I/C	O/E	S/A	A/P	FR	G/L	P/O	C/R	CTL
Orders (Enter)									
Order #:	588	Customer:	1		Friendly Frame Shop			Part ship? Y	
Type:	INVOICE				Sal: 115.51			Back ords? Y	
					CR lim: 1,000				
1. Seq #	10								
2. Item #	MLDG-1				Sample moulding sample.				
					CHOP 16 1/2 X 20 3/4				
					EXACT CUT:				
3. Qty ordered	2				FRAMES			On-hand:	409.326
4. Qty to ship	2				FRAMES			Commitd:	17.600
								On B/O:	0
								Available:	391.726
								On order:	0
(Backorderable)									
5. Unit price	3.00				per FOOT			GROSS	TO-SHIP
6. Discount %	.00								
7. Ship date	ASAP								
8. Pft ctr	(Not applicable)				Line \$:	48.00		48.00	
9. Rtr to inv	(Not applicable)				Disc \$:	.00		.00	
10. Rtr to inv	(Not applicable)				Extnd \$:	48.00		48.00	
0 lines Run. qty:	2				Tot ord \$:	48.00			
Field number to change ?									

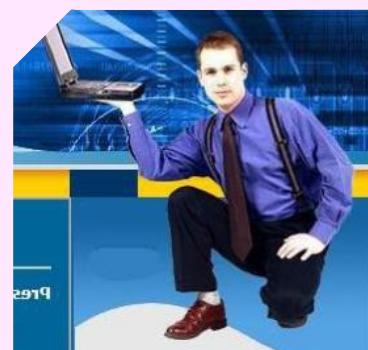
Arquitectura SPADE



APEL tiene dos formas de representación del proceso: significa, destinada a usuarios finales del proceso y para descripciones del proceso a alto nivel y textual, para usuarios avanzados y herramientas.

La arquitectura consiste básicamente en:

- Un **editor gráfico** para capturar y modelar procesos.
- Un **traductor** de la representación gráfica en textual.
- Un **compilador** de la representación textual a un formalismo ejecutable.
- Un **estado común**, que mantiene el estado actual del proceso en ejecución y de las entidades creadas durante la ejecución.
- Un **conjunto de servicios** que varían desde servicios de interfaz de usuario a servicios de control.



Para dar soporte a la interoperabilidad entre distintos PSEE la arquitectura básica de APEL ha evolucionado para incorporar los siguientes elementos adicionales:

- **Metamodelo Común**, para facilitar el intercambio de información y entendimiento de los distintos PSEE de la federación.
- **Servidor del Proceso**, que en tiempo de ejecución contiene el modelo de procesos y la reificación de todas las entidades creadas por el proceso en ejecución. Su interfaz permite a los componentes crear cualquier entidad y cambiar de forma arbitraria el proceso actual, así como el modelo. Este es el aspecto básico que da soporte a la evolución del proceso en APEL.

- **Servidor de Eventos**, que captura y gestiona todos los eventos (tal y como se han definido en el modelo de procesos).
- **Motor Común del Proceso**, que en función de los eventos que recibe del servidor de eventos se encarga de la ejecución del modelo de procesos común asegurando que se cumple la semántica del proceso del servidor del proceso.
- **Modelo de Interoperabilidad**, que recibe peticiones de los motores de proceso en forma de evento y transforma si es necesario, esos eventos en peticiones a otros servidores de proceso. También se encarga de garantizar la consistencia en la federación.



Los conceptos básicos del formalismo APEL para el modelado de los aspectos estáticos son: actividad, producto y agente. Estos elementos pueden tener estados, que varían como consecuencia de transiciones que se disparan debido a la ocurrencia de eventos. La noción de estado y evento constituyen el núcleo de los aspectos dinámicos del metamodelo APEL.

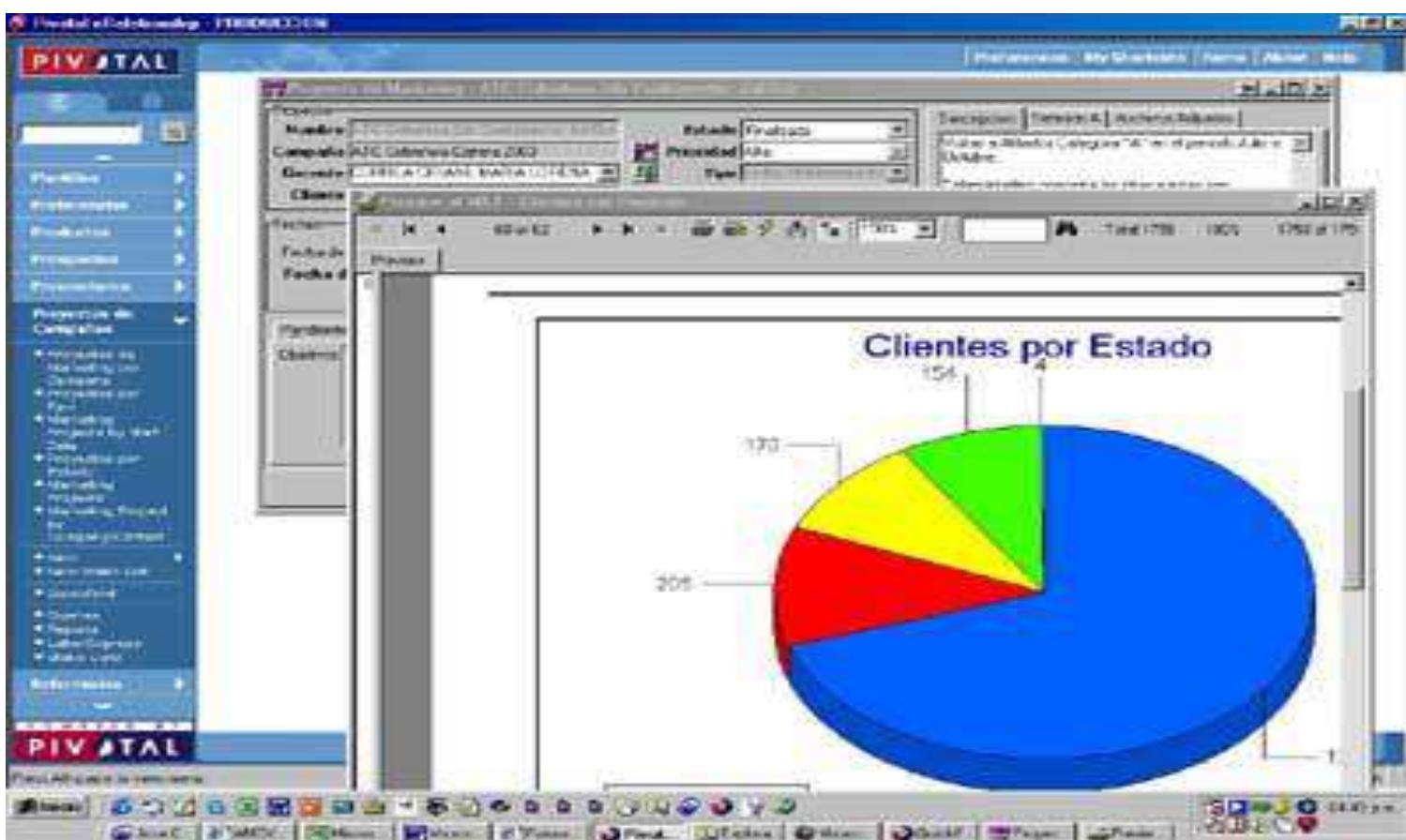
TEMA 4

Ciclo de Vida



Competencia:

Reconocer los diversos procesos en el ciclo de vida del software.





Tema 04: Ciclo de Vida

¿QUÉ ES EL CICLO DE VIDA?



Uno de los problemas más importantes en cualquier departamento de sistemas de información es definir un marco de referencia común que pueda ser empleado por todas las personas que participan en el desarrollo de los sistemas, y en el que se definen los procesos, las actividades y las tareas a desarrollar.

A lo largo de la historia se han propuesto diferentes paradigmas o ciclos de vida para el software: desde el ciclo en cascada, pasando por el modelo en espiral de Boehm, hasta los más recientes ciclos de vida orientados al objeto, como el ciclo de vida fuente.

Las organizaciones profesionales y los organismos internacionales se han venido ocupando del ciclo de vida del software, tanto IEEE como ISO/IEC han publicado normas tituladas, respectivamente, "IEEE Standard for Developing Software Life Cycle Processes" (Estandar IEEE para el Desarrollo de Procesos del Ciclo de Vida del Sofware) (IEEE, 1995, 1998c), e "Information Technology - Software life-cycle processes" (Proceso del ciclo de vida software) (ISO, 1995a; ISO, 2002a, 2004e).

La norma ISO 12207 entiende por modelo de ciclo de vida "Un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso". Por otro lado, la norma ISO 15288 (ISO, 2003) define ciclo de vida de los sistemas como "la evolución en el tiempo de un sistema de interés desde su concepción hasta su retirada", destacando que un modelo de ciclo de vida es "un marco de procesos y actividades relativas al ciclo de vida que actúa también como una referencia para la comunicación y el entendimiento".



El ciclo de vida abarca, por tanto, toda la vida del sistema, comenzando con su concepción y finalizando cuando ya no se utiliza. A veces también se habla de "*ciclo de desarrollo*", que es un subconjunto del anterior y que empieza en el análisis y finaliza con la entrega del sistema al usuario.



Procesos del ciclo de vida software

En la norma ISO 12207, las actividades que se pueden realizar durante el ciclo de vida del software se agrupan en procesos principales, procesos de soporte y procesos generales (de la organización), hay que destacar que la norma no fomenta o especifica ningún modelo concreto de ciclo de vida, gestión del software o método de ingeniería, ni prescribe como realizar ninguna de las actividades.

Procesos principales

Los procesos principales son aquellos que son útiles a las personas que inician o realizan el desarrollo, la explotación o el mantenimiento del software durante su ciclo de vida. Los procesos principales son:

- **Proceso de adquisición.** El propósito de este proceso es obtener el producto o servicio que satisface la necesidad expresada por el cliente. Este proceso consta de cuatro subprocessos: preparación de la adquisición, selección de proveedor, supervisión del proveedor y aceptación del cliente.
- **Proceso de suministro.** Este proceso proporciona un producto o servicio al cliente que satisface los requisitos acordados.



- **Proceso de desarrollo.** El propósito de este proceso es transformar un conjunto de requisitos en un producto o sistema basado en software que satisface las necesidades planteadas por el cliente.

Debido al interés que tiene este proceso, se resumen a continuación sus principales subprocessos:



- **Elicitación de requisitos**, cuyo objetivo es recopilar, procesar y seguir la traza de las necesidades y requisitos del cliente a lo largo del ciclo de vida del producto o servicio, así como establecer una línea de configuración (*baseline*) que sirva como base para definir los productos de trabajo necesarios.
- **Análisis de requisitos del sistema**, cuyo objetivo es transformar los requisitos definidos por los participantes o implicados (*stakeholders*) en un conjunto de requisitos técnicos del sistema deseado que guiarán el diseño del sistema.

- **Diseño arquitectónico del sistema**, cuyo objetivo es identificar qué requisitos del sistema que deben ser ubicados en los elementos del mismo.
- **Análisis de los requisitos del software**, cuyo objetivo es establecer los requisitos de los elementos de software del sistema.
- **Diseño del software**, cuyo objetivo es proporcionar un diseño para el software que implemente los requisitos y pueda ser verificado respecto a los mismos.
- **Construcción del software**, cuyo objetivo es producir unidades de software ejecutable que reflejen apropiadamente el diseño del software.

- **Integración del software**, cuyo objetivo es combinar las unidades de software produciendo elementos de software integrados, consistentes con el diseño software, que demuestra que se satisfacen los requisitos funcionales y no funcionales sobre una plataforma equivalente o completa.
- **Prueba del software**, cuyo objetivo es confirmar que el producto software integrado satisface los requisitos definidos.
- **Integración del sistema**, cuyo objetivo es integrar los elementos del sistema (incluyendo elementos software, elementos hardware, operaciones manuales, y otros sistemas) para producir un sistema completo que satisfaga el diseño del sistema y las expectativas de los clientes expresadas en los requisitos del sistema.



- **Prueba del sistema**, cuyo objetivo es asegurar que la implementación de todos los requisitos del sistema se prueba para la conformidad y que el sistema está listo para entregar.
- **Instalación del software**, cuyo objetivo es instalar el producto software que satisface los requisitos acordados en el entorno objetivo.
- **Proceso de operación**, este proceso incluye la operación del producto software en su entorno final y proporcionar soporte a los clientes del mismo. Consta de dos subprocessos: uso operacional y soporte al cliente.

- **Proceso de mantenimiento**, este proceso incluye la modificación de un sistema o producto software después de la entrega para corregir los fallos, mejorar el rendimiento u otros atributos, o adaptarlo a un entorno modificado. Esta modificación o la retirada de los productos existentes deben hacerse preservando la integridad de las operaciones organizacionales.

Procesos del ciclo de vida de sistemas

De la misma forma que la norma ISO 12207 para el software, en la norma ISO 15288 se presentan los principales procesos del ciclo de vida de los sistemas agrupados en cuatro categorías:

- **Procesos de acuerdo**, que incluyen los procesos de adquisición y suministro.
- **Procesos empresariales**, que incluyen: el proceso de gestión del entorno empresarial (cuyo objetivo es definir y mantener las políticas y procedimientos necesarios para las actividades de la organización), gestión de la inversión (cuyo objetivo es iniciar y mantener suficientes y adecuados proyectos con el fin de conseguir los objetivos de la organización), gestión de los procesos del ciclo de vida de sistemas (cuyo objetivo es asegurar que se encuentran disponibles para ser utilizados por la organización procesos efectivos de ciclo de vida del sistema), gestión de recursos (para proporcionar recursos a los proyectos), gestión de la calidad (la norma establece que el propósito del proceso de gestión de calidad es "asegurar que los productos, servicios e implementaciones de los procesos del ciclo de vida cumplen los objetivos de calidad de la empresa") logran la satisfacción del cliente.

- **Procesos de proyecto**, que se utilizan para establecer y hacer evolucionar planes de proyecto, valorar los logros actuales y el progreso respecto a los planes y controlar la ejecución del proyecto hasta su culminación.
- **Procesos técnicos**, que incluyen el proceso de definición de requisitos de las partes implicadas en el producto, análisis de requisitos, diseño arquitectónico, implementación, integración, verificación, transición, validación, operación, mantenimiento y retirada.

Lecturas Recomendadas

❖ PROCESO DE SOFTWARE

http://www.congresoson.gob.mx/ISO/normas/normatividad_conceptos.pdf

❖ MANTENIMIENTO DEL SOFTWARE

<http://alarcos.inf-cr.uclm.es/per/fruiz/cur/mso/comple/ISO14764.pdf>

Actividades y Ejercicios



1. En un documento en Word indique normas relacionadas con el estándar para los procesos (P.) de ciclo (C.) de vida (V.) del software (S.) y de una breve explicación sobre que trata cada una.

Envíalo a través de "**P. C V. S.**".

2. En un documento en Word redacte cuales de los procesos de ciclo de vida software que aparecen en la norma ISO 12207, además indique si son más aplicables para pequeñas y medianas empresas al momento de desarrollar un software.

Envíalo a través de "**ISO 12207**".

Autoevaluación

- 1) De acuerdo a la norma ISO 1995, un proceso se define como un conjunto de _____ que se transforman en entradas y en salidas.**
- a. Proceso software.
 - b. Procedimientos lógicos.
 - c. Actividades interrelacionadas.
 - d. Algoritmos de diseño.
 - e. Situaciones problemáticas.
- 2) De acuerdo a la norma ISO 1998, el _____ es un conjunto de procesos usados por las organizaciones para planificar, gestionar, ejecutar, monitorizar, controlar y mejorar sus actividades software relacionadas.**
- a. Situaciones problemáticas.
 - b. Procedimientos lógicos.
 - c. Actividades interrelacionadas.
 - d. Proceso de diseño.
 - e. Proceso software.
- 3) Los procesos software constan de dos subprocessos interrelacionados:**
- a. Proceso de producción y proceso de gestión.
 - b. Proceso de producción y proceso de software.
 - c. Proceso de productividad y proceso de calidad.
 - d. Proceso de gestión y proceso de calidad.
 - e. Proceso de gestión y proceso de software.
- 4) Es un conjunto de actividades con ciertas relaciones entre si y con objetos determinados instantes de tiempo:**
- a. El proceso de gestión.
 - b. El formato de proceso de producción.
 - c. El proceso de calidad.
 - d. El formato de intercambio de procesos.
 - e. El proceso de software.
- 5) En PSL los tres constructores principales, a partir de los cuales se derivan la mayoría de los elementos de los modelos de procesos, son:**
- a. Nombres, elementos básicos y guía.
 - b. Actividad, objeto e instante de tiempo.
 - c. Rol, producto y actividad.
 - d. Actividad, guía y producto.
 - e. Nombres, objeto y clase.

- 6) Los entornos de ingeniería del software orientados al proceso (PSEE), dan soporte a los procesos de ingeniería, usados por ejemplo para:**
- a. Diseñar un producto.
 - b. Mantener un producto software.
 - c. Desarrollar procesos de gestión.
 - d. Desarrollar metamodelos.
 - e. Mantener un metamodelo de un proceso.
- 7) El motor de procesos está constituido por alguno de los siguientes elementos:**
- a. Un respuestario.
 - b. Entorno de programación.
 - c. Interacción del metamodelo.
 - d. Intérprete del modelo de procesos.
 - e. Encendido y apagado.
- 8) Es un ejemplo de PSEE.**
- a. SPADE.
 - b. ESPAD.
 - c. APPLE.
 - d. APEEL.
 - e. SLANG.
- 9) La norma ISO 12207 está relacionado con:**
- a. El motor de proceso.
 - b. El proceso de software.
 - c. El PSEE.
 - d. El motor de gestión.
 - e. El ciclo de vida.
- 10) Los primeros procesos principales de un ciclo de vida son de:**
- a. Inicio, desarrollo, desactualización.
 - b. Nacimiento, desarrollo y muerte.
 - c. Adquisición, suministro y desarrollo.
 - d. Inicio, prueba, desactualización.
 - e. Nacimiento, prueba, muerte.

Resumen

UNIDAD DE APRENDIZAJE III: CALIDAD DEL PROCESO SOFTWARE

Un proceso se define como un conjunto de actividades interrelacionadas que se transforman en entradas y en salidas. El proceso software es un proceso o conjunto de procesos usados por una organización o proyecto, para planificar, gestionar, ejecutar, monitorizar, controlar y mejorar sus actividades software relacionadas. Los procesos pueden ser modelados en diferentes niveles de abstracción y con diferentes objetivos.

Los diagramas de Gantt y los diagramas PERT son representaciones gráficas de los procesos en el que se incluyen las tareas, su duración y sus relaciones de precedencia. PROMENADE es un lenguaje para la modelización de procesos software que utiliza UML. SPEM describe un metamodelo genérico para la descripción de procesos software concreto. El metamodelo SMSDM, establece un marco de trabajo para la definición y extensión de metodologías de desarrollo de software.

Los Entornos de Ingeniería del Software Orientados al Proceso (PSEE), dan soporte a los procesos de ingeniería, usados para concebir, diseñar, desarrollar y mantener un producto software. Un elemento clave del entorno constituye el motor del proceso. Todo PSEE está caracterizado por el lenguaje de modelado de procesos (LMP) que utiliza.

El modelo de ciclo de vida es un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso. Las actividades que se pueden realizar durante el ciclo de vida del software se agrupan en procesos principales, procesos de soporte y procesos generales.



UNIDAD 4



Evaluación y mejora de Procesos

Introducción

a) Presentación y contextualización

Los temas que se desarrollan en la presente unidad tienen por finalidad de que el alumno conozca que hoy en día la calidad del software no puede garantizarse únicamente centrando los programas de calidad en el producto, dado que, tal y como se ha comentado la calidad final del producto software está directamente relacionado con la forma en que se desarrolla y mantiene, es decir, con el proceso. Todo ello ha motivado en gran medida que las organizaciones dedicadas al desarrollo y mantenimiento del software se preocupen cada vez más de la mejora de sus procesos.

b) Competencia

Identifica los diferentes modelos de mejora y evaluación de procesos del producto software.

c) Capacidades

1. Conoce la Medición de Sistemas de Información proporcionadas a las organizaciones.
2. Analiza el modelo ideal y el proceso de software personal para la mejora de procesos.
3. Describe el proceso de software de equipo y el modelo CMM que se implantan en las empresas.
4. Aplica la norma estándar ISO/IEC 15504 en la evaluación de software.

d) Actitudes

- ✓ Respeta los estándares o normas ISO 90003 en el mejoramiento y evaluación de procesos.
- ✓ Indaga más información acerca del modelo ideal y el proceso de software personal.

e) Presentación de Ideas básicas y contenido esenciales de la Unidad:

La Unidad de Aprendizaje 04: Evaluación y mejora de procesos, comprende el desarrollo de los siguientes temas:

TEMA 01: Medición de sistemas de información.

TEMA 02: El modelo ideal y el proceso de software personal.

TEMA 03: Proceso de software de equipo y el modelo CMM.

TEMA 04: El estándar ISO/IEC 15504.

Medición de Sistemas de Información

TEMA 1



Competencia:

Conocer la Medición de Sistemas de Información proporcionadas a las organizaciones.



Desarrollo de los Temas



Tema 01: Medición de Sistemas de Información

LA NORMA ISO 90003

La norma ISO/IEC 90003 (ISO, 2004f) proporciona la guía necesaria en las organizaciones para la aplicación de la ISO 9001 (ISO, 2000b) a la adquisición, suministro, desarrollo, operación y mantenimiento de software y sus servicios relacionados.



Identifica todos los aspectos que deberían ser tratados y es independiente de la tecnología, modelos de ciclo de vida, procesos de desarrollo y estructuras organizacionales. La norma ISO 9001, especifica los requisitos para un sistema de gestión de la calidad cuando una organización necesita demostrar su capacidad de proporcionar de forma coherente productos que satisfagan los requisitos del cliente y aspira a aumentar su satisfacción a través de la aplicación eficaz del sistema, incluyendo los procesos para la mejora continua del sistema, y el aseguramiento de la conformidad con los requisitos y de acuerdo a las reglamentaciones existentes.

El estándar ISO 90003 surge debido a que la gestión de la calidad que propone ISO 9001 aun siendo un buen marco de partida, es excesivamente general y se queda corta para abordar proyectos de diseño e implantación de sistemas de gestión de la calidad más especializados. Las directrices proporcionadas en esta norma internacional no están enfocadas a ser utilizadas como criterios de evaluación en la certificación y registro del sistema de gestión de la calidad.

Según la norma ISO 90003 la organización debe establecer, documentar, implementar y mantener un sistema de gestión de la calidad software y mejorar continuamente su eficacia de acuerdo con los siguientes requisitos generales:

- Identificar los procesos necesarios para el sistema de gestión de la calidad y su aplicación a través de la organización. En este punto la organización debería identificar también los procesos de desarrollo, operación y mantenimiento de software.



- Determinar la secuencia e interacción de estos procesos. La organización debería también definir la secuencia e interacción de los procesos en: los modelos de ciclos de vida del desarrollo de software, por ejemplo, modelos en cascada, incremental y en espiral (evolutivos); la planificación de la calidad y el desarrollo, que debería basarse en un modelo de ciclo de vida.
- Determinar los criterios y métodos necesarios para asegurarse de que tanto la operación como el control de estos procesos sean eficaces.



- Asegurarse de la disponibilidad de recursos e información necesarios para apoyar la operación y el seguimiento de estos procesos.
- Realizar el seguimiento, la medición y el análisis de estos procesos.
- Implementar las acciones necesarias para alcanzar los resultados planificados y la mejora continua de estos procesos.

EL MODELO DE MADUREZ DE LA CAPACIDAD (CMM)

CMM

Desde la década de los años 1980, el Instituto de Ingeniería del Software (SEI, *Software Engineering Institute*) de la Universidad de Carnegie Mellon se ha centrado en proporcionar la base necesaria para mejorar el desarrollo del software considerando a las tareas de desarrollo del software como una serie de procesos que se pueden definir, medir y controlar. Como resultado se han obtenido modelos de referencia de la capacidad de los procesos y modelos de evaluación de dicha capacidad.

CMM (SEI, 1995) es el modelo propuesto por el SEI como referencia para determinar la capacidad de los procesos software de una organización. CMM proporciona a las



organizaciones de software el modelo de referencia necesario como soporte para el control de sus procesos de desarrollo y mantenimiento y para facilitar su evolución hacia una cultura de la Ingeniería del Software y de excelencia en la gestión. Es un

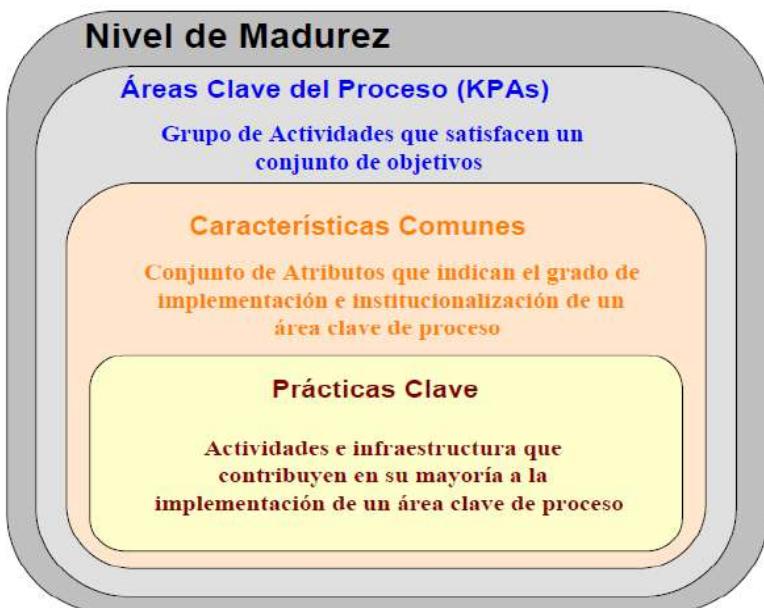
modelo con la finalidad de:

- Evaluar la madurez de los procesos de desarrollo de software dentro de una organización.
- Proponer un plan de mejora de los procesos de desarrollo de software de acuerdo a una serie de niveles.



A la hora de establecer la madurez de los procesos de una organización en CMM se establecen cinco niveles de capacidad, que definen una escala ordinal para representar la evolución del proceso software desde un nivel inicial caótico (procesos ad hoc cuyos resultados no son predecibles) hasta un estado de mejora continua (maduro). El modelo de referencia CMM establece una serie de áreas clave (hasta un total de dieciocho) agrupadas en los distintos niveles de madurez. Para que una organización pueda estar en un determinado nivel de madurez debe satisfacer los criterios de evaluación asociados con las áreas clave que pertenecen a ese nivel y a los niveles anteriores. Cada área clave de proceso o *KPA* (*Key Process Área*) se describe en función de una serie de prácticas clave (*KPs, Key Practices*), que a su vez se organizan en una serie de características comunes (*Comun features*).

Estructura de CMM como modelo de referencia para la evaluación



Niveles de Capacidad de CMM

Nivel	Características	Resultados
Inicial	<ul style="list-style-type: none"> - Ausencia de gestión de proyectos. - El proceso de software es cambiante e irregular. - Los planes, estimaciones y calidad son impredecibles. - El rendimiento depende de la capacidad individual de los miembros del grupo. - Se establecen programas de formación del personal de desarrollo y mantenimiento. 	Productividad y calidad escasa. Riesgo máximo
Repetible	<ul style="list-style-type: none"> - Los procesos de software son estables y repetibles. - La organización establece políticas de gerencia de proyectos y procesos. - La planificación se basa en proyectos similares. - Existen estándares definidos y exigidos. - El proceso se enmarca en un sistema de gerencia de proyectos basado en experiencias pasadas. 	Productividad y calidad baja. Riesgo alto.
Definido	<ul style="list-style-type: none"> - Los procesos son definidos: estandarizados, documentados e institucionalizados. - Los procesos de ingeniería y gerencia son estables y se integran en uno sólo. - Existe un entendimiento común de los procesos, funciones y responsabilidades. - La organización mantiene un grupo dedicado a la definición, mejoramiento y difusión del proceso de Ingeniería de Software. 	Productividad y calidad media. Riesgo medio.
Gestionado	<ul style="list-style-type: none"> - Los procesos son medibles o cuantificables - La productividad y la calidad se miden y registran para cada proyecto de la organización. - Se fijan metas cuantitativas de la calidad del software. - Mediante el uso de métricas de software, se crea una base cuantitativa para la evaluación y estimación en proyectos futuros. 	Productividad y calidad alta. Riesgo mínimo.
Optimizado	<ul style="list-style-type: none"> - Los procesos se mejoran continuamente. - La organización busca lograr el nivel máximo de capacidad. - Se incorporan nuevas tecnologías y métodos para mejorar los procesos. 	Productividad y calidad total. Riesgo nulo.

Cada nivel de madurez se define en base a:

- Áreas clave del proceso. Cada nivel de madurez, excepto el nivel inicial se descompone en diferentes áreas clave del proceso. Ejemplos de áreas clave son la gestión de configuración y planificación del proyecto del segundo nivel de madurez, o la prevención de defectos y gestión de cambio del proceso. correspondientes al nivel quinto de madurez. Cada área clave contiene un conjunto de objetivos o metas, que describen de forma general que se debe hacer para dar soporte a un área clave de proceso. Las metas se usan para comprobar si efectivamente se implementa adecuadamente un área clave de proceso determinada.



- Características Comunes. Cada área clave de proceso se organiza en una serie de características comunes que representan los atributos que debe tener el proceso. Mediante la evaluación de las características comunes se puede averiguar si la implementación de un área clave de proceso se ha realizado de forma que sea efectiva, repetible y duradera.
- Prácticas Clave. Constituyen los ejemplos de que se debe hacer para satisfacer los objetivos de un área clave de proceso sin entrar en detalle de cómo hacerlo.



Para poder conocer el nivel de madurez de una organización es necesario realizar la evaluación de sus procesos software. Por este motivo y con el fin de proporcionar el medio necesario para realizar evaluaciones basadas en CMM y para poder comparar los resultados de evaluación se creó el marco de trabajo CAF (*CMM Appraisal Framework*), que identifica los requisitos y características necesarias en un método de evaluación basado en CMM para mejorar la consistencia y la fiabilidad de los diferentes métodos de evaluación y sus resultados.

- Los dos principales métodos de evaluación basados en CMM son SCE (*Software Capability Evaluation*) y CBA-IPI (*CMM - Based Appraisal for Internal Process Improvement*). Por otra parte, el marco de mejora de procesos del SEI, basado en CMM, lo constituye el modelo IDEAL. A continuación se resumen todos ellos.

MÉTODOS MÁS REPRESENTATIVOS DE EVALUACIÓN



SCE (Software Capability Evaluation)

SCE (Bymes y Philips, 1996) es el método desarrollado para evaluar los procesos software de una organización con el objetivo de determinar su capacidad. La capacidad de un proceso se refiere al rango de los resultados esperados que se pueden obtener al llevar a cabo un proceso determinado.

Las principales áreas de aplicación de SCE son: la selección del suministrador, la monitorización del proceso y la evaluación interna. SCE usa el modelo de madurez de capacidad (CMM) como modelo de referencia. El objetivo de la evaluación de SCE es el proceso software, y en particular, se centra en conjuntos de procesos (o lo que en CMM se considera como áreas clave de procesos) que se pueden agrupar en tres categorías:

- Procesos organizacionales. que contienen un conjunto de áreas clave que se centran sobre la gestión organizacional de los procesos software: los procesos de gestión de proyectos. centrados en aspectos de gestión de proyectos como su planificación y seguimiento; y los procesos de ingeniería. que incluyen aspectos de desarrollo del producto como la gestión de requisitos, ingeniería del producto, revisiones por pares, etc.

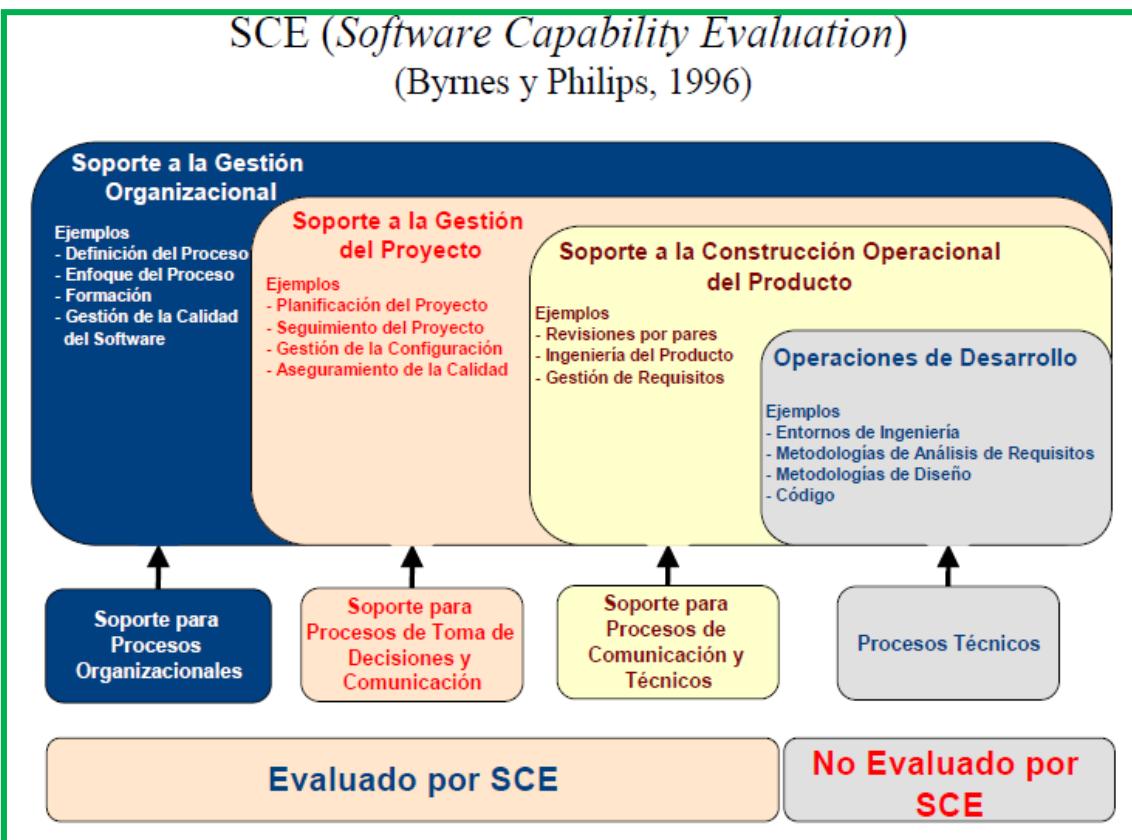
- Aunque en el modelo CMM se consideran los procesos de producción técnica, estos no se incluyen en el alcance de la evaluación proporcionada por SCE.

- El proceso de evaluación definido en SCE está compuesto básicamente por las siguientes actividades: planificar y preparar la evaluación. Llevar a cabo la evaluación e informar sobre los resultados de la evaluación.

- El equipo de evaluación SCE lleva a cabo una planificación en la que básicamente identifica las áreas de proceso a evaluar para realizar un proceso de evaluación basado en rigurosas revisiones de documentación y entrevistas en el que, mediante un proceso de análisis, se establecen las debilidades y fortalezas de la evaluación para finalmente realizar los informes adecuados en función de los resultados obtenidos.



Procesos Evaluados por SeE



CBA-IPI (CMM-Based Appraisal for Internal Process Improvement)

CBA-IPI (Dunaway y Masters, 2001) es un método que facilita a una organización conocer la capacidad de sus procesos software mediante la identificación de las fortalezas y debilidades y la relación de estas fortalezas y debilidades en base al modelo CMM, con el fin de establecer y dar prioridad a planes de mejora software y para facilitar que la organización se centre en la mejora de los aspectos que le resulten más beneficiosos en función de su nivel de madurez y sus objetivos de negocio.



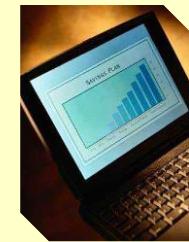
El método consiste en la evaluación de la capacidad del proceso software de una organización a través de un grupo de profesionales adecuadamente entrenados que trabajan como un equipo para averiguar y valorar las distintas áreas clave del proceso de CMM que se encuentran en el alcance de la evaluación. Los datos necesarios para la valoración se obtienen a partir de datos obtenidos de cuestionarios, revisiones de documento, presentaciones y entrevistas con gestores, jefes de proyecto y agentes software.

Fases y Actividades de la evaluación SCE

Fase SCE v 3.0	Actividades y Resultados
	<p>La Organización Patrocinadora:</p> <ul style="list-style-type: none"> • Determina los atributos deseados del producto • Determina la capacidad del proceso más apropiada para alcanzar los objetivos de negocio (la capacidad objetivo del proceso) • Selecciona y forma al equipo de la evaluación (SCE) <p>Resultado: Se definen los objetivos y los requisitos de la evaluación</p>
Planificar y Realizar la preparación para la Evaluación	<p>El Equipo SCE:</p> <ul style="list-style-type: none"> • Identifica las áreas en las que la organización carece de experiencia (indicando un riesgo potencial) • Define el alcance de la evaluación. <p>Resultado: Se define el alcance de la evaluación definido y se completan las preparaciones a alto nivel para evaluar a la organización de desarrollo.</p> <p>El Equipo SCE:</p> <ul style="list-style-type: none"> • Selecciona los proyectos a evaluar. • Prepara los temas específicos para la evaluación. • Analizar los datos <p>Resultado: Se completan las preparaciones detalladas para evaluar un</p>
Conducir la Evaluación	<p>El Equipo SCE:</p> <ul style="list-style-type: none"> • Investiga cada tema planificado en el sitio de desarrollo. • Conduce actividades de recogida de datos mediante la realización de entrevistas, revisiones de documentos y presentaciones. • Consolida la información recogida y valida las observaciones. • Determina los puntos fuertes, débiles y las actividades de mejora. <p>Resultado: Datos del Proceso consolidados y se determinan los resultados.</p>
Informar los Resultados de la Evaluación	<p>El Equipo SCE:</p> <ul style="list-style-type: none"> • Presenta y entrega los resultados al patrocinador y a la organización. • Produce un informe final para el patrocinador. • Realiza recomendaciones para el uso de los resultados. <p>Resultado: Se determinan y documentan los resultados de la evaluación Datos del Proceso consolidados y se determinan las búsquedas.</p>

Los dos principales objetivos de CBA IPI son:

- Dar soporte, habilitar y animar a una organización a la mejora del proceso software.
- Proporcionar una visión exacta de las fortalezas y debilidades de los procesos software actual de la organización, usando CMM como modelo de referencia y para identificar las áreas clave del proceso que es necesario mejorar.



Las actividades y alcance del proceso de evaluación del método CBA-IPI son básicamente los mismos que en el método SCE (*planificación, conducción y generación de informes*). En realidad, CBA-IPI es muy similar a SCE con la diferencia fundamental de que CBA IPI es una evaluación centrada en la mejora de procesos, mientras que SCE suele orientarse más a la selección de suministradores, aunque también se puede usar para la evaluación interna de procesos.

De acuerdo a la tecnología usada en los modelos de evaluación basados en CMM, se considera que CBA IPI es un método para la valoración (assessment) de la capacidad para mejora de procesos, mientras que SCE es un método de evaluación (emillation) con el fin de seleccionar suministradores o para medir el progreso de las mejoras. La diferencia fundamental entre la valoración y la evaluación es que la primera consiste en un proceso que una organización hace para sí misma, mientras la segunda es un proceso en el que un grupo externo llega a una organización y examina la capacidad de sus procesos para tomar decisiones respecto de posibles negocios o tratos futuros.



El alcance de una valoración es relativo a las necesidades de la organización y objetivos de negocio del patrocinador, que es usualmente el gestor senior de la organización evaluada. En contraste, el alcance de una evaluación es relativo a las necesidades del patrocinador, que es la persona o grupo de personas responsables de decidir si se hace la evaluación de la organización con la que se pretende hacer negocios. Los resultados de la evaluación de los métodos comentados anteriormente se pueden utilizar en el contexto de la mejora de procesos software, ya sea para la mejora de los procesos de la propia organización evaluadora (CBA-IPI) o para mejora en la organización evaluada (SCE).

Diferencias entre Valoración y Evaluación

	SCE	CBA IPI
	Valoración (<i>Assessment</i>)	Evaluación (<i>Evaluation</i>)
Objetivo	Mejora de Procesos	Selección y monitorización de suministradores Los desarrolladores lo usan para medir el progreso de la mejora
Comunicación de Resultados	Se usan en la organización evaluada	Los resultados se hacen saber al patrocinador (<i>sponsor</i>)
Equipo	Colaborativo. Los miembros de la organización deben formar un equipo	Los miembros de la organización podrían no estar en un equipo
Alcance	Se aplica a la organización, no a proyectos individuales o contratos	Se aplica a necesidades específicas del patrocinador
Uso de los resultados	Como entradas al plan de mejora	Como entrada a la selección de suministrador, gestión de riesgos y mediciones de la mejora interna



El Modelo Ideal y el Proceso de Software Personal

TEMA 2



Competencia:

Analizar el modelo ideal y el proceso de software personal para la mejora de procesos.





Tema 02: El Modelo ideal y el Proceso de Software Personal

EL MODELO IDEAL

El marco de mejora de procesos del SEI lo constituye el modelo IDEAL en el que se define un marco de ciclo de vida para la mejora de procesos. Este modelo fue concebido originalmente como un ciclo de vida para la mejora de procesos software basado en el modelo CMM y posteriormente el modelo IDEAL fue revisado en la versión 1.1 para proporcionarle un alcance más amplio. IDEAL constituye un enfoque usable y entendible para la mejora continua estableciendo los pasos necesarios que se deben seguir para llevar a cabo un programa de mejora y proporcionando un enfoque ingenieril y disciplinado.



El modelo IDEAL está compuesto por cinco fases, cada una de las cuales esta formada por una serie de actividades:

- Iniciación, que constituye el punto de partida, en el cual se establece la infraestructura, los roles y responsabilidades que hay que asumir y se asignan los recursos necesarios. En esta fase se elabora el plan de mejora de procesos que proporciona la guía necesaria para completar el inicio y llevar a cabo las fases de diagnóstico y establecimiento. Además, se decide la aprobación del programa de mejora, se establecen los recursos necesarios y se establecen los objetivos generales de la mejora a partir de las necesidades de negocio de la organización. Estos objetivos son refinados en la fase posterior de establecimiento.



Respecto a la infraestructura, se establecen dos componentes fundamentales: un grupo directivo de la gestión (Management Steering Group, MSG) y un grupo de procesos de ingeniería del software (software engineering process group, SEPG). Durante la fase de inicio también se realizan planes para comunicar el comienzo de la iniciativa de mejora y se sugiere la necesidad de realizar evaluaciones para determinar la preparación de la organización a la hora de llevar a cabo una iniciativa de mejora de procesos.

- Diagnóstico, en la que se lleva a cabo el trabajo preliminar necesario para realizar las fases posteriores. En esta fase se inicia el plan de acción de la mejora de acuerdo con la visión de la organización, el plan de negocio estratégico, las lecciones aprendidas de esfuerzos de mejora realizados en el pasado, aspectos clave a los que se enfrenta la organización y los objetivos a largo plazo. Se realizan actividades de valoración para establecer una línea base del estado actual de la organización, entregándose sus resultados y recomendaciones en las acciones del plan de mejora.

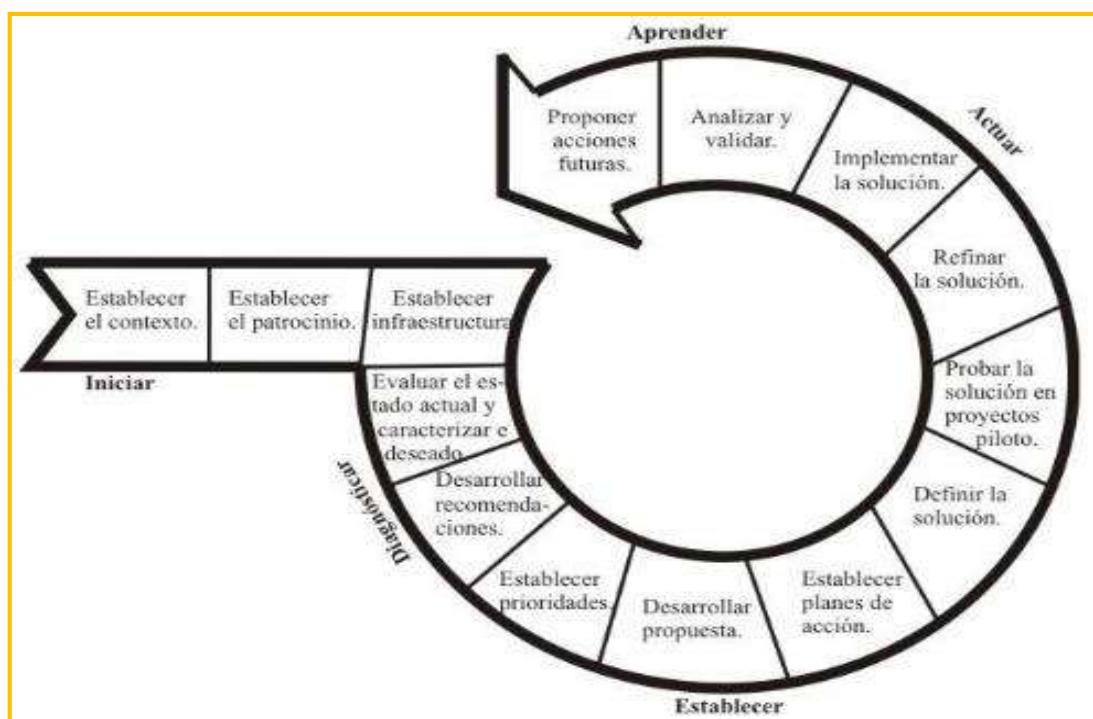
- Establecimiento, durante la cual se priorizan los aspectos que la organización ha decidido mejorar, se desarrollan las estrategias necesarias para obtener las soluciones de mejora y se completa el borrador del plan de mejora definido en las fases anteriores. En esta fase también se desarrollan objetivos medibles a partir de los objetivos generales fijados en la fase de inicio y que son incluidos en el plan de mejora. Ello conlleva además la definición de las técnicas necesarias para el control del progreso y se preparan los recursos y se proporciona la formación necesaria a los grupos de trabajo técnico (*Technical Working Groups, IFVG*). El plan de acción desarrollado debe guiar las actividades de mejora de acuerdo a los aspectos detectados para la mejora ordenados según su prioridad y según las recomendaciones de la fase de diagnóstico. También durante esta fase se crean plantillas de acciones tácticas que los grupos de trabajo técnico deben completar y llevar a cabo.

- Actuación, en la que se crean y se llevan a cabo las acciones destinadas a mejorar las áreas identificadas en las fases previas. Se desarrollan planes para ejecutar las acciones de mejora y para evaluar o probar los procesos nuevos o mejorados. Una vez completada exitosamente la prueba de nuevos procesos y tras determinar su adecuación para ser adoptados en la organización. Se desarrollan y ejecutan los planes necesarios para su implantación.
- Aprendizaje, cuyo objetivo es tratar de hacer más efectiva la siguiente iteración por el modelo IDEAL cuando sea necesaria. Una vez alcanzada esta fase, se han desarrollado las soluciones, se han aprendido importantes lecciones del proceso y se han tomado mediciones sobre el rendimiento y la consecución de los objetivos marcados.



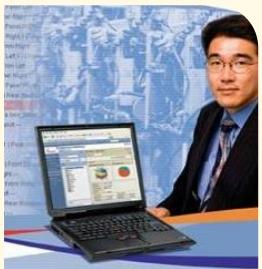
Estos artefactos son añadidos a la base de datos del proceso, que constituye una fuente de información muy relevante para el personal implicado en la próxima iteración por las fases del modelo. La información reunida permite realizar una evaluación sobre la estrategia, los métodos y la infraestructura utilizada en el programa de mejora, lo que permite su conexión y ajuste de cara a futuras mejoras. Es necesario plantear algunas preguntas, como por ejemplo sobre el rendimiento de la infraestructura (equipos de trabajo MSG, SEPG, TWG, etc.) y los métodos empleados por los TWG en sus actividades de desarrollo de la solución. Una respuesta adecuada a cada una de estas preguntas es fundamental para plantear el siguiente ciclo del modelo IDEAL.

El modelo IDEAL (McFeeley, 1996)



PSP (PERSONAL SOFTWARE PROCESS)

En el contexto del modelo CMM y a la hora de facilitar la aplicación de los procesos de evaluación y mejora en una organización, es necesario implantar buenas prácticas en el desarrollo software. Con tal fin se desarrolló el método PSP (Humphrey, 1997). El Proceso de Software Personal (PSP) apoya a las empresas que están llevando a cabo o tienen planeado implementar un plan de mejora de procesos basados en un modelo como CMM, ayudando a crear personal capacitado y disciplinado en su trabajo.



Esta principalmente basado en CMM y permite implementar las prácticas de ingeniería del software descritas en dicho modelo a nivel individual, incorporando de forma efectiva, eficaz y a bajo costo aspectos tales como la planificación y seguimiento de proyectos, las revisiones e inspecciones, el proceso de ingeniería del producto, el enfoque y la medición cuantitativa del proceso, la prevención de defectos, la evaluación de calidad, etc.

Al igual que en CMM, PSP se basa sobre los principios de mejora del proceso, sin embargo, mientras que CMM se centra en mejorar la capacidad de la organización, PSP se centra en la mejora de los ingenieros software aplicando la gestión y control del proceso a nivel individual. Con PSP los ingenieros desarrollan software usando un enfoque disciplinado y estructurado, siguiendo un proceso definido y planificando, midiendo, realizando un seguimiento de su trabajo, gestionando la calidad del producto y aplicando la realimentación obtenida para mejorar sus procesos de trabajo individuales.

Entre los beneficios que PSP ofrece a los ingenieros de software destacan los siguientes:

- Proporciona una serie de principios al ingeniero para llevar a cabo un proceso personal disciplinado.
- Asiste a los ingenieros en la realización de planes precisos.
- Determina los pasos que los ingenieros deben seguir para mejorar la calidad del producto.
- Establece bancos de pruebas para medir la mejora del proceso personal.
- Determina el impacto que los cambios del proceso tienen sobre el rendimiento del ingeniero.



Niveles de Proceso de PSP



Estos resultados son obtenidos haciendo que los participantes recopilan datos específicos relacionados con el proceso y el producto y estableciendo la línea base que proporcione a los ingenieros con un contexto para mejorar el proceso.

Para alcanzar un nivel se deben cumplir los requisitos establecidos en los niveles previos, más los nuevos impuestos en dicho nivel.



Estos niveles son:

La Línea Base del Proceso Personal (PSPO y PSPO.1), que proporciona una introducción al PSP y establece la base inicial a partir del histórico de datos de tamaño, tiempos y defectos. A este nivel los ingenieros escriben programas y

se les permite usar sus métodos actuales, pero dentro del marco de trabajo compuesto por los siguientes seis pasos representados en la siguiente tabla.

PASO	FASE	DESCRIPCION
1	planificar	planificar el trabajo
2	diseñar	diseñar el programa
3	codificar	implementar el diseño
4	compilar	compilar el programa y corregir y registrar los defectos
5	probar	realizar las pruebas del programa y corregir
		y registrar los defectos encontrados
6	postmorten	registrar los tiempos reales de tamaño, tiempo y
		defectos en el plan

Las tres medidas base de PSP son: tiempo de desarrollo, defectos y tamaño. Todas las demás medidas en PSP son derivadas de las anteriores. El proceso de medición en PSP se introduce desde las tres primeras asignaciones del proceso en los niveles PSPO y PSPO.1. En la siguiente tabla se muestran ejemplos de los formularios que se utilizan para el registro de tiempos y defectos:

Formulario de Registro de Tiempos

FORMULARIO DE REGISTROS DE TIEMPOS						
FECHA	COMIENZO	FIN	Tiempo. Interrup	T. DELTA	FASE	COMENTARIOS
13/05/2005	7:58	8:45	3	44	planificar	llamada telefónica
	8:47	10:29	2	100	diseñar	crear y revisar diseño
	7:49	8:59		70	codificar	codificar todas las funciones
	9:15	9:46		31	compilar	compilar y enlazar
	9:47	10:10		23	probar	ejecutar pruebas A B y C
	4:33	4:51		16	postmortem	

Gestión Personal del Proyecto (PSP1 Y PSP 1.1), se centra en las técnicas para la gestión del proyecto a nivel individual. Se introducen métodos para la estimación del esfuerzo y planificación y seguimiento de calendario. Las estimaciones de tamaño y esfuerzo se realizan usando el método PROBE (*Proxy-Based Estimating*), con el que los ingenieros usan el tamaño relativo del Proxy, como por ejemplo objetos, puntos función, procedimientos, y los transforman a líneas de código (LOC).



Gestión Personal de la Calidad (PSP.2 y PSP2.1), añade métodos de gestión de la calidad a PSP tales como: revisiones personales de diseño y código, una notación para el diseño, plantillas de diseño, técnicas de verificación y métricas para gestionar la calidad del proceso y del producto. El objetivo es encontrar y eliminar todos los defectos antes de llegar a la primera compilación, para lo cual se define una métrica de rendimiento definida como el porcentaje de defectos introducidos que fueron eliminados antes de la compilación. Los nuevos pasos del proceso "revisar el diseño" y "revisar el código" son incluidos en PSP2 para ayudar a los ingenieros a obtener un 100% en la métrica de rendimiento. Las revisiones son realizadas por el ingeniero sobre su propio diseño y código, y son revisiones estructuradas, dirigidas por datos y son guiadas por listas de comprobación derivadas de los datos históricos de los defectos introducidos por el ingeniero.



Formulario de Registro de Defectos

FORMULARIO DE REGISTRO DE DEFECTOS						
FECHA	Nº	TIPO	INTRODUCIDO EN	ELIMINADO EN	TIEMPO CORRECCIÓN	DEFECTO CORREGIDO
13/5/05	1	20	Código	COMPIL	22	
Descripción		Error de sintaxis en sentencia scanf				
FECHA	Nº	TIPO	INTRODUCIDO EN	ELIMINADO EN	TIEMPO CORRECCIÓN	DEFECTO CORREGIDO
13/5/05	2	20	Código	COMPIL	18	
Descripción		Error de declaración de función				
FECHA	Nº	TIPO	INTRODUCIDO EN	ELIMINADO EN	TIEMPO CORRECCIÓN	DEFECTO CORREGIDO
13/5/05	3-6	20	Código	COMPIL	1	
Descripción		Faltaba:				

Proceso Personal Cílico (PSP3), que resuelve la necesidad de escalar PSP de manera eficiente a proyectos de mayor tamaño sin sacrificar la calidad o la productividad. En este nivel los ingenieros deben aprender a alcanzar la productividad más alta en un determinado rango de tamaño. Por debajo de este rango la



productividad tiende a disminuir debido a costes generales. Por encima de este rango la productividad también tiende a disminuir porque se ha alcanzado el límite de escalabilidad del proceso. PSP3 soluciona este límite introduciendo una estrategia de desarrollo cílico en la que los programas grandes se descomponen en partes que luego son integradas.



Esta estrategia asegura que los ingenieros trabajan en sus niveles máximos de productividad y calidad con solo costes incrementales, no exponenciales, en grandes proyectos. Respecto a los niveles anteriores se introducen dos nuevos formularios: un resumen de ciclo para resumir el tamaño, tiempo de desarrollo y defectos en cada ciclo; y un registro de seguimiento de problemas, para documentar los aspectos que pueden afectar a los ciclos futuros o completados. Usando PSP3 los ingenieros descomponen su proyecto en series de ciclos PSP2.1, y luego integran y prueban las salidas de cada ciclo. Teniendo en cuenta que los programas que se producen con PSP2.1 son de alta calidad, los costes de integración y pruebas se minimizan.



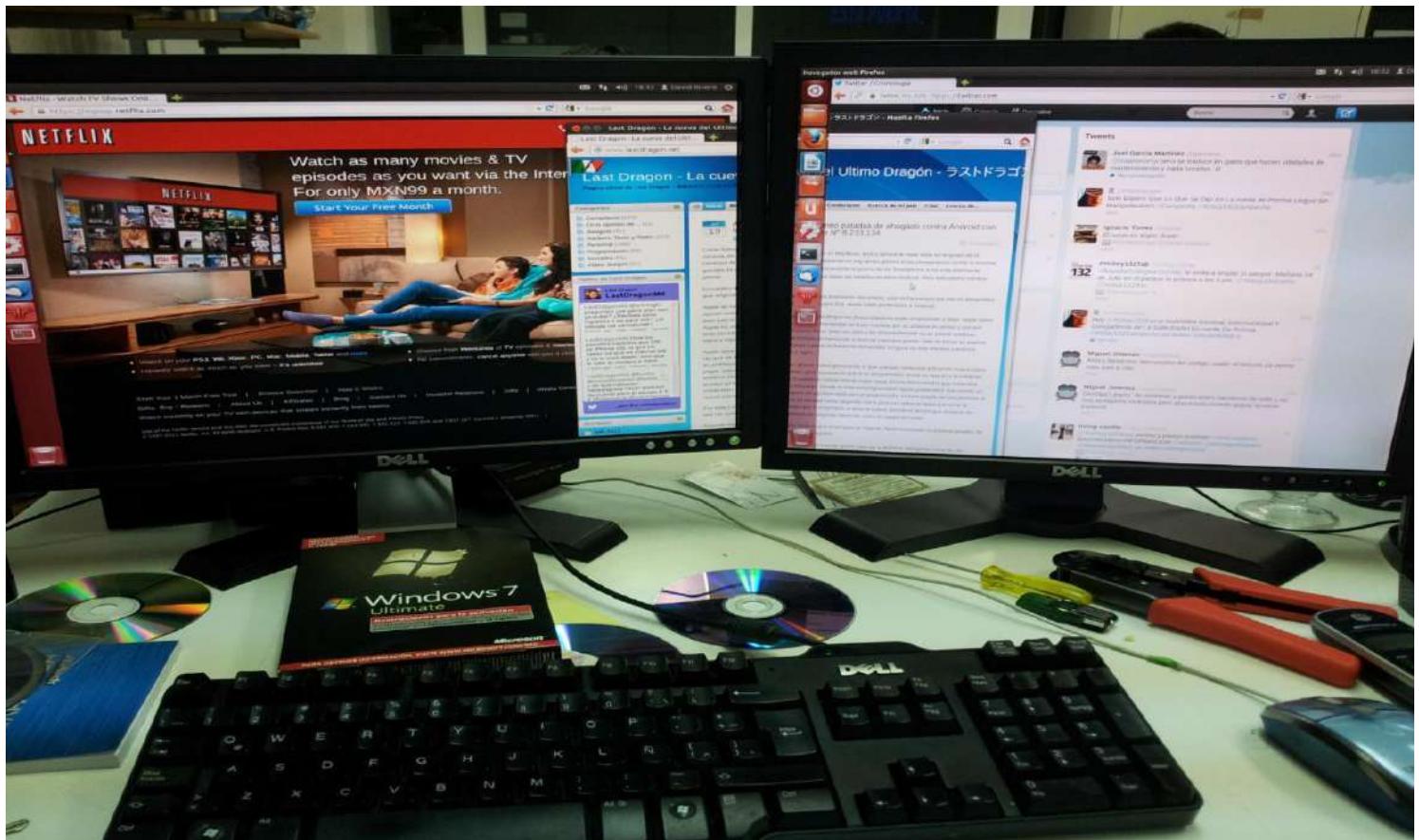
Proceso de Software de Equipo y el Modelo CMM

TEMA 3



Competencia:

Describir el proceso de software de equipo y el modelo CMM que se implantan en las empresas.





Tema 03: Proceso de Software de Equipo y el Modelo CMM

TSP (TEAM SOFTWARE PROCESS)

El Proceso de Software de Equipo (TSP), ayuda a conformar equipos para el desarrollo de software de calidad. TSP proporciona un marco de trabajo, que se construye sobre la base de PSP, con fases de desarrollo bien definidas, en las que los productos de software se generan en varios ciclos.



Además, se establecen medidas estándares para la calidad del producto y para el desempeño de los equipos y de los desarrolladores y se aplican evaluaciones por rol y del equipo, fomentando una disciplina en el proceso y proporcionando una guía para resolver los problemas del trabajo en equipo. TSP es un proceso que los equipos utilizan para planificar su trabajo, ejecutar sus planes y mejorar de forma continua sus procesos de desarrollo software. El proceso TSP se define a través de una serie de guiones en los que se describen todos los aspectos de planificación de proyectos y desarrollo de productos. En este proceso se incluyen las definiciones de los roles del equipo, las métricas definidas, y el proceso postmortem. TSP se considera como una instancia del nivel 5 de CMM, definida para equipos.



El origen de TSP se debe a las limitaciones que PSP tenía en el ámbito industrial. PSP ha tenido un gran éxito en entornos académicos y de hecho los datos obtenidos de los alumnos que han aplicado PSP han sido muy consistentes. Este hecho creó una evidencia muy significativa sobre los beneficios que los ingenieros obtendrían al usar PSP: les permitiría tener el control de su proceso personal mediante la mejora de sus habilidades de estimación y la reducción de los defectos introducidos en los productos sin afectar a su productividad. Sin embargo no quedaba claro como los ingenieros podrían aplicar estas habilidades en la práctica dentro de las empresas. Como resultado no se aplicó PSP de forma efectiva en la industria salvo en algunas empresas.

Además, se descubrió en empresas formadas en PSP, que los gestores no proporcionaban el entorno necesario e incluso volvían a repetir las mismas prácticas caóticas que utilizaban antes de aprender PSP. Como consecuencia, Humphrey desarrollo el Team Software Process como una respuesta operacional al problema de implementar PSP en equipos dentro de las organizaciones. PSP cubre solo las fases de desarrollo de software desde el diseño a las pruebas unitarias. Era necesario un proceso más práctico que cubriera también los requisitos, las pruebas de integración, la documentación y otras actividades típicas en todo proyecto de desarrollo. Por otro lado, TSP debía incluir otros aspectos importantes como los roles de equipo, interrelaciones dentro de la organización y la definición de un proceso de equipo para ser utilizado dentro de los procesos existentes en la organización.

TSP proporciona un proceso operacional definido para guiar a los ingenieros y gestores sobre los pasos necesarios en la construcción de equipos. Los procesos operacionales son procesos que definen de forma precisa el trabajo a realizar y se consideran como guiones más que como las descripciones textuales muy extensas que aparecen en los libros de definición de los procesos de la organización. Estos guiones son diseñados para facilitar su uso por los miembros de equipo cuando están realizando su trabajo. El proceso especifica los pasos necesarios para establecer un entorno efectivo de trabajo en equipo. Sin una guía específica los ingenieros tienden a saltarse pasos, o realizarlos en un orden poco productivo, o gastando tiempo pensando qué hacer después. TSP les proporciona los procesos operacionales necesarios para formar equipos de ingeniería, establecer un entorno de equipo efectivo y guiar a los equipos a la hora de realizar su trabajo.

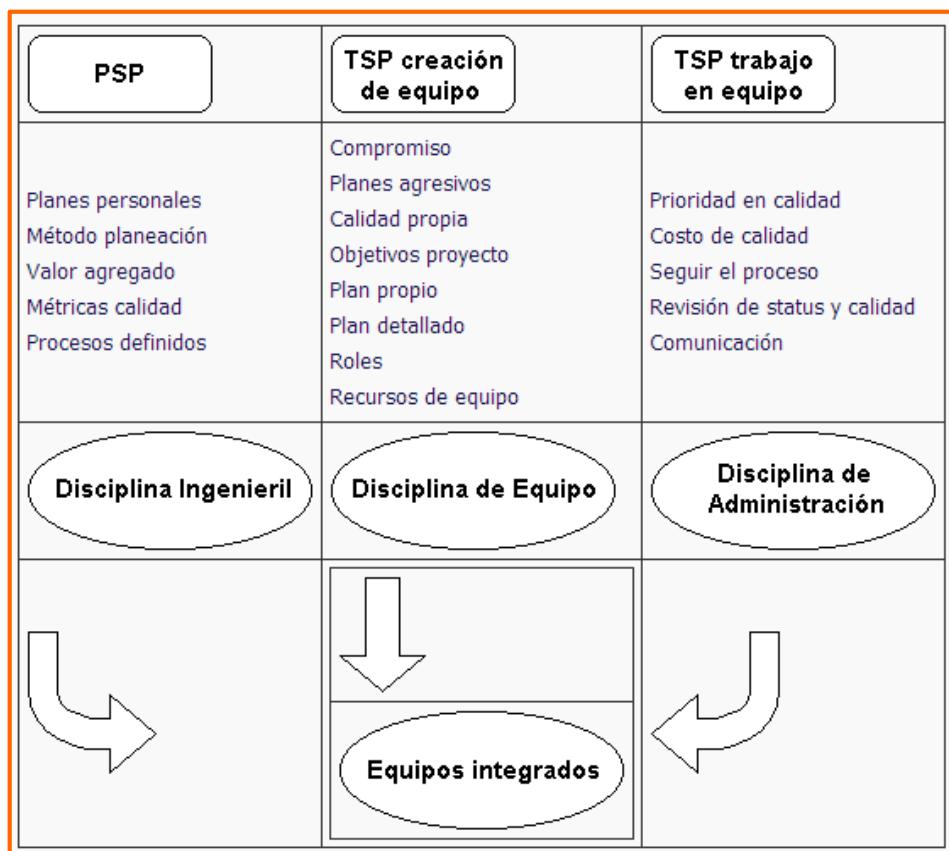


Antes de que los miembros puedan participar en un equipo TSP, deben conocer cómo realizar un trabajo disciplinado. Tal y como se muestra en la figura de abajo, es necesario que los ingenieros que usan TSP estén formados en PSP. La formación en PSP incluye el aprendizaje necesario para: realizar planes detallados, reunir y usar datos del proceso, desarrollar planes, usar los valores obtenidos para realizar un seguimiento del proyecto, medir y gestionar la calidad del producto y definir y usar procesos operacionales. En TSP, la tarea de construir el equipo es un proceso de planificación de cuatro días denominado lanzamiento del equipo (team launch).



En este proceso, todos los miembros del equipo desarrollan la estrategia, el proceso y el plan para hacer su proyecto. El lanzamiento del equipo está compuesto por nueve reuniones, cada una de las cuales tiene un guión con las actividades a seguir descritas en detalle y que el entrenador utiliza para guiar al equipo. Como resultado se obtiene un plan, en los que todos los miembros del equipo deberían haber participado y con el que deben estar comprometidos. Una vez completado este proceso inicial, el equipo sigue su proceso definido para hacer el trabajo necesario.

Formación de Equipos en TSP



De acuerdo a TSP, los equipos son relanzados periódicamente. Ello se debe a que TSP sigue una estrategia de desarrollo iterativa y evolutiva, lo que hace que los relanzamientos periódicos sean necesarios de forma que cada fase o ciclo pueda ser planificado de acuerdo al conocimiento obtenido en los ciclos previos. El relanzamiento también es necesario para actualizar los planes detallados de los ingenieros, que normalmente son precisos solo para unos pocos meses.

En cada relanzamiento los equipos hacen un plan global y un plan detallado de los tres meses o cuatro meses posteriores. Durante cada lanzamiento del equipo también se elabora el plan de calidad. Para gestionar la calidad los equipos establecen métricas y objetivos de calidad así como planes para alcanzar dichos objetivos y los medios para conocer el progreso y llevar a cabo acciones colectivas cuando no se satisfacen los objetivos. TSP enseña a los equipos como deben realizar este proceso de gestión de calidad mediante guiones en los que se definen las métricas a usar como parte del proceso.

Las métricas pueden ser de tamaño (por ejemplo en miles de líneas de código, KLOC), tiempo (en minutos y horas), calidad (en forma de defectos), rendimiento del proceso (% de defectos eliminados antes de una fase determinada) y densidad de defectos (defectos KLOC) de los productos obtenidos. En TSP se establece como estas métricas son definidas, estimadas, recopiladas, presentadas y analizadas. También se hace uso en el proceso de datos históricos de los equipos, y de líneas guía sobre calidad y planificación.



PEOPLE CAPABILITY MATURITY MODEL (PEOPLE-CMM)

El modelo de madurez de capacidad de las personas (People Capability Maturity Model. People-CMM), es un marco de trabajo que ayuda a las organizaciones a resolver de forma exitosa los aspectos críticos relacionados con sus recursos humanos. Está basado en las mejores prácticas en campos como los recursos humanos, la gestión del conocimiento y el desarrollo organizacional para guiar a las organizaciones a la hora de mejorar sus procesos de gestión y desarrollo de sus empleados.

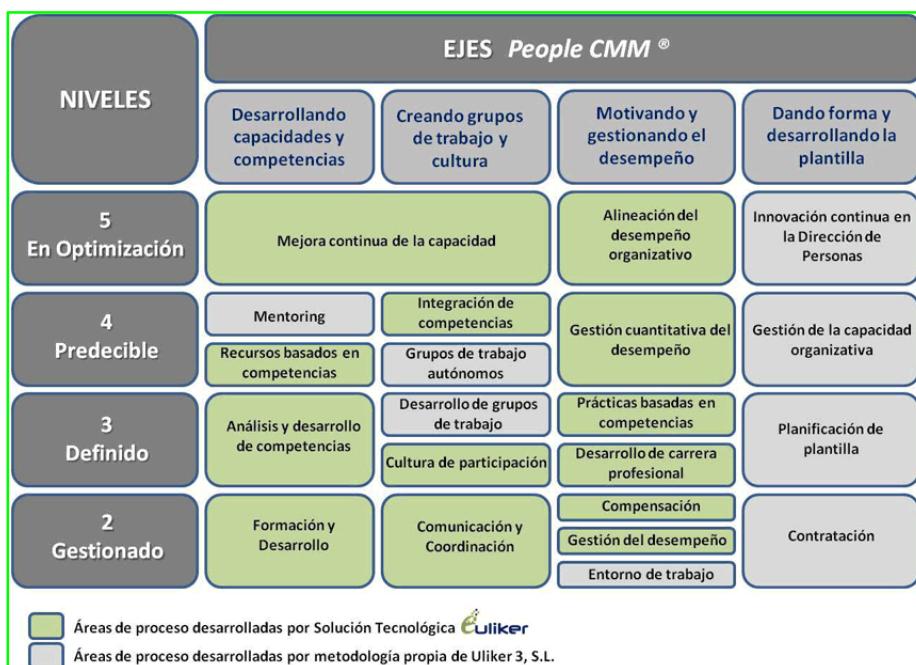
People CMM proporciona un programa de desarrollo continuo de los empleados, establece prioridades para acciones de mejora, integra el desarrollo de los empleados con la mejora de procesos y establece una cultura de excelencia. El Modelo People CMM está diseñado sobre la premisa de que las prácticas de mejoras de los empleados no tendrán éxito al menos que el comportamiento de la organización cambie para darles soporte.

Es un modelo basado en el proceso que asume que las prácticas de los empleados son procesos estándares de la organización que pueden ser mejorados de forma continua mediante los mismos métodos que se utilizan para otros procesos de negocio. People CMM consiste en cinco niveles de madurez a través de los cuales las prácticas y procesos de las fuerzas de trabajo van evolucionando. Estos niveles establecen las bases necesarias para la mejora continua de las competencias individuales, el desarrollo de equipos efectivos, la motivación en la mejora del rendimiento y el establecimiento de las fuerzas de trabajo que la organización necesita para llevar a cabo sus planes de negocio.



Desde la perspectiva de People CMIM, la madurez de la organización se deriva de las prácticas de fuerzas de los empleados que son realizadas de forma rutinaria y el punto hasta el cual estas prácticas han sido integradas dentro de un proceso institucionalizado para mejorar su capacidad. En una organización madura, los individuos responsables realizan prácticas repetibles como requisitos ordinarios y esperados de acuerdo a su puesto de trabajo. A medida que una organización es más madura, mayor capacidad tiene para atraer, desarrollar y retener el talento que necesita para ejecutar sus negocios.

Niveles de Madurez de People-CMM



EI

Estándar

ISO/IEC 15504

TEMA 4



Competencia:

Aplicar la norma estándar ISO/IEC 15504 en la evaluación de software.





Tema 04: El Estándar ISO/IEC 15504

ESTÁNDAR ISO/IEC 15504

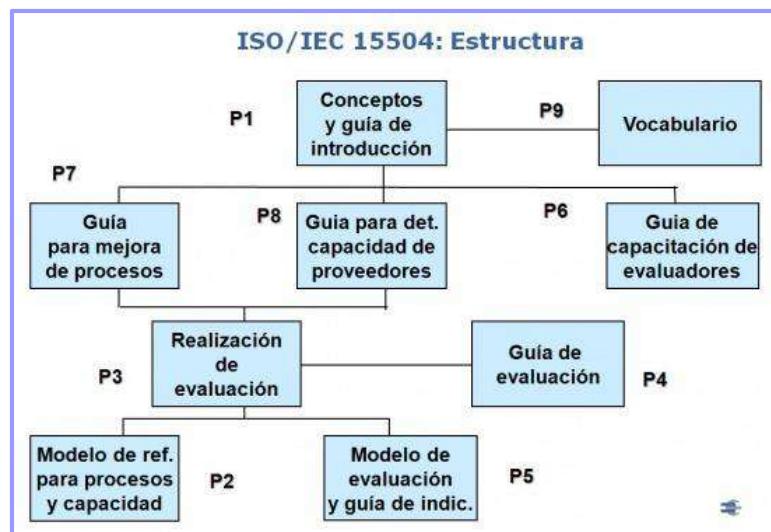


El estándar ISO/IEC 15504 (ISO, 2004a; 2004b; 2004c; 2004d: 2006) proporciona un marco de trabajo para la evaluación de procesos software y establece los requisitos mínimos para realizar una evaluación que asegure la repetibilidad y consistencia de las valoraciones obtenidas,

La evaluación del proceso es aplicable en el contexto de una organización que actúa en su nombre o representando otra organización para: entender el estado de sus propios procesos con el fin de mejorarlos; determinar la capacidad de los procesos de otra organización a través de un contrato; determinar la capacidad de sus propios procesos ante un requisito o clase de requisitos en particular.

La parte informativa del estándar proporciona la guía necesaria sobre cómo utilizar un proceso de evaluación dentro de un programa de mejora o dentro de un tipo de proceso para la determinación de la capacidad.

El estándar está compuesto por cinco partes (que sustituyen las nueve partes de la versión anterior de 1998), y de las cuales la quinta se encuentra actualmente en preparación.



El objetivo de la evaluación del proceso es conocer la capacidad de los procesos de una organización. Como resultado de una exitosa implementación de la evaluación de los procesos se determina la información que caracteriza los procesos evaluados y el punto hasta el cual los procesos realizan su propósito.

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



NIVEL DE MADUREZ 0

La organización NO tiene una
implementación efectiva de los procesos

NIVEL DE MADUREZ 1

Se alcanza el propósito de los procesos en términos generales.
Los procesos se realizan cuando es necesario, pero no se hace
de una forma planificada ni se realiza ningún seguimiento.

NIVEL DE MADUREZ 2

Se obtienen los productos del proceso, pero esta vez, de acuerdo
con una planificación y realizándose un seguimiento. Estos
productos se ajustan a unos estándares y a unas especificaciones
prefijadas. También se tienen definidos plazos y recursos.

NIVEL DE MADUREZ 3

Los procesos se realizan y se gestionan utilizando procedimientos
definidos. Cada implementación de un proceso se hace utilizando
procedimientos creados según un estándar y documentados.

NIVEL DE MADUREZ 4

Se recogen medidas detalladas del Nivel de realización de los
procesos y se analizan. Esto permite mantener el proceso dentro
de unos límites predefinidos, así como disponer de una mejor
posición para poder cuantificar la capacidad del proceso y
predecir su comportamiento.

NIVEL DE MADUREZ 5

La realización de los procesos se optimiza de forma
continuada, de cara a su contribución a alcanzar los objetivos
de negocio de la organización. Se establecen objetivos
cuantitativos de eficacia y eficiencia en la realización de los
procesos, basados en los objetivos de negocio de la
organización. Se lleva a cabo una monitorización continua de
los procesos y se analizan los datos obtenidos. Esto permite
que los procesos estándar definidos dentro de la
organización cambien dinámicamente, para adaptarse de
forma efectiva a los actuales y futuros objetivos de la
empresa.

IT360.es
IT QUALITY MANAGEMENT

Toni Martín Ávila, junio 2010

CMMI Y SCAMPI

El éxito y amplia aceptación de CMM propicio la aparición de modelos similares **ISO / IEC TR 15504** incluso en otras disciplinas además del software. Esta proliferación de modelos ha facilitado la aparición de conflictos en los objetivos y técnicas de la mejora de procesos, debido al considerable incremento en el entrenamiento requerido, y a la confusión por parte de los que los aplican de cuál de los modelos usar según sus necesidades específicas. CMMI constituye un solo modelo que cubre múltiples disciplinas y se creó con el objetivo de eliminar esas desventajas.

El proyecto CMMI persigue objetivos tanto a corto como a largo plazo. Los objetivos iniciales (los cuales se llevaron a cabo en el 2000 con la publicación de la versión 1.0 de los modelos CMMI-SE/SW y CMMI-SE/SW/IPPD) consistían en integrar tres modelos de mejora de procesos específicos: software, ingeniería de sistemas y desarrollo de procesos y productos integrados. CMMI-SE/SW especifica el modelo CMMI que contiene las disciplinas de ingeniería de sistemas y software. CMMI-SE/SW/IPPD indica el modelo que añade material para la integración de procesos y desarrollos de procesos en CMMI-SE/SW.

Esta integración fue propuesta para reducir el coste de la mejora de procesos basados en modelos e implementados mediante varias disciplinas de la siguiente manera:

- ✚ Eliminando inconsistencias.
- ✚ Reduciendo duplicaciones.
- ✚ Incrementando la claridad y comprensión.
- ✚ Proporcionando terminología común.
- ✚ Proporcionando estilos consistentes.
- ✚ Estableciendo reglas de construcción uniformes.
- ✚ Manteniendo componentes comunes.
- ✚ Asegurando la consistencia con ISO 15504.
- ✚ Siendo susceptible a la inferencia de esfuerzos legales.



Los objetivos a largo plazo consisten en establecer la base necesaria para la posterior inclusión de otras disciplinas (tales como adquisición y seguridad). Para facilitar ambos modelos de integración actuales y futuros, el equipo de desarrollo de CMMI creó un marco de trabajo automatizado y extensible y definió reglas para la posible inclusión de más disciplinas dentro de este marco de trabajo.

Modelos de CMMI

DISCIPLINA DEL MODELO	MODELO FUENTE	DESCRIPCIÓN MODELO FUENTE
Software	El CMM para software (SW-CMM)	Modelo que describe los principios y prácticas fundamentales de la madurez de procesos software. El CMM está organizado para ayudar a las organizaciones de software a mejorar mediante una trayectoria evolutiva, creciendo con fines específicos, desde un ambiente caótico hacia unos maduros y disciplinados procesos de software
Ingeniería de Sistemas	Modelo de Capacidad de Ingeniería de Sistemas (EIA/IS 731)	Integración de todas las disciplinas de sistemas para que conozcan las necesidades técnicas y de negocio de la forma más efectiva
Proceso integrado de desarrollo de productos	Desarrollo integrado de producto CMM (IPD CMM)	Enfoque sistemático para el desarrollo del producto que incrementa la satisfacción del cliente mediante una colaboración oportuna de las disciplinas necesarias a lo largo del ciclo de vida del producto



Un concepto fundamental de todos los modelos CMMI es el área de procesos. No todo lo relacionado con procesos y mejora de procesos está incluido en un modelo de mejora de procesos. Como sus predecesores, CMMI selecciona solo los aspectos más importantes de la mejora de procesos y entonces agrupa estos aspectos dentro de "áreas".

Desde el punto de vista de los contenidos del modelo CMMI, hay que indicar que en este modelo se hace una distinción de los mismos en tres categorías principales, que por orden de importancia son: requerido, esperado e informativo.

- **Material requerido:** Es esencial para el modelo y para la comprensión de que es necesario para la mejora de procesos y para hacer demostraciones de conformidad con el modelo.

- **Material esperado:** No es un material esencial para el modelo, y en algunos casos, podría no estar presente en una organización que use el modelo CMMI con éxito. Sin embargo, el material esperado se considera que juega un papel fundamental en la mejora de procesos.
- **Material Informativo:** Es el más extenso y constituye la mayoría del modelo. Proporciona una guía útil para mejorar los procesos, y ofrecer una mayor claridad para la comprensión de los componentes requeridos y esperados.



Otro aspecto importante y muy novedoso en el modelo CMMI, es que usa dos tipos de representaciones de sus modelos, incluyendo de esta forma la representación de CMM y la representación de ISO 15504: **representación por etapas y continuo**.

Representación por etapas

Un modelo por etapas proporciona un marco predefinido para la mejora organizacional basada en el agrupamiento y ordenación de procesos y en las relaciones organizacionales asociadas. El término "por etapas" viene de la forma en la que el modelo describe este marco como una serie de "etapas", denominadas "niveles de madurez". Cada nivel de madurez tiene un conjunto de áreas de procesos que indican en qué aspectos debiera centrarse una organización para la mejora sus procesos. Cada área de procesos está descrita en términos de prácticas que contribuyen a satisfacer sus objetivos. Las prácticas describen la infraestructura y actividades que más contribuyen en la implementación e institucionalización efectiva de las áreas de procesos. El progreso ocurre satisfaciendo los objetivos de todas las áreas de proceso en un nivel de madurez determinado. El CMM para software es el ejemplo primordial de modelo por etapas.



Como se puede observar en la figura la representación por etapas se estructura en torno a niveles de madurez, que se van alcanzando en la medida que se cumplen en la organización las áreas clave de proceso asociadas a cada nivel de madurez. Esta representación permite evaluar los procesos de una organización para establecer la madurez global de la misma.

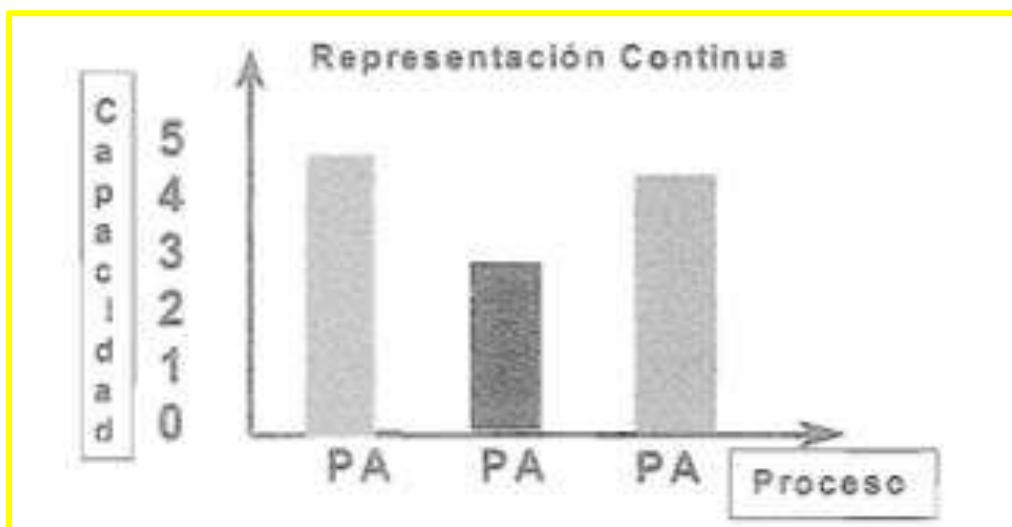
Representación continua



Los modelos continuos proporcionan una guía menos específica con respecto al orden en el cual debería realizarse el proceso de mejora. Se denominan continuos porque ninguna etapa discreta está asociada con la madurez de la organización. Como los modelos por etapas, los modelos continuos tienen áreas de procesos que contienen prácticas. A diferencia de los modelos por etapas, las prácticas de un área de procesos en un modelo continuo están organizadas de forma que dan soporte a la mejora y al crecimiento de procesos individuales. La mayoría de las prácticas asociadas con la mejora de procesos son genéricas; son externas a las áreas de procesos individuales y son aplicables a todas las áreas de procesos. Las prácticas genéricas están agrupadas bajo niveles de capacidad, cada una de las cuales tiene una definición que es casi equivalente a la definición de niveles de madurez en los modelos por etapas.

La representación continua no se estructura en torno a niveles de madurez, sino que lo que se facilita es la evaluación de procesos individuales, permitiendo que una organización pueda seleccionar un conjunto de sus procesos individuales para evaluarlos y conocer la madurez concreta de dichos procesos.

Representación Continua de CMMI



Lecturas Recomendadas

- ❖ **MODELO IDEAL Y EL PROCESO DE SOFTWARE PERSONAL**
http://www.reocities.com/SiliconValley/lab/3629/IDEAL_ciclo.pdf
- ❖ **PROCESO DE SOFTWARE DE EQUIPO Y EL MODELO CMM**
<http://www.globales.es/imagen/internet/Informaci%C3%B3n%20General%20CMMI.pdf>
- ❖ **EL ESTÁNDAR ISO/IEC 15504**
<http://www.kybeleconsulting.com/recursos/articulos/implantacion-iso15504-con-scrum/>

Actividades y Ejercicios



1. En un documento en Word realice un cuadro comparativo sobre las generalidades, campo de aplicación y alcance de las normas ISO 9000-3, ISO 9001, ISO 2000b.
Envíalo a través de "**ISO's**".
2. Aplique el proceso de TSP para el desarrollo de dos pequeños proyectos software rellenando la información de tiempos y defectos. Realice una comparativa de los dos proyectos destacando los progresos obtenidos a nivel personal.
Envíalo a través de "**Proceso de TSP**".

Autoevaluación

- 1) CMM, estas siglas pertenecen a:**
 - a. Modelo de madurez de la capacidad.
 - b. Modelo de madurez de la calidad.
 - c. Modelo de mejora continua.
 - d. Calidad de modelo de madurez.
 - e. Calidad de modelo mejorado.

- 2) Cada área clave de proceso se organiza en una serie de _____ que representan los atributos que debe tener el proceso.**
 - a. Áreas claves del proceso.
 - b. Prácticas clave.
 - c. Características diferenciadas.
 - d. Calidad diferenciada.
 - e. Características comunes.

- 3) ¿Qué especifica la norma ISO 9001?**
 - a. Características diferenciadas de los sistemas.
 - b. Prácticas claves del software.
 - c. Requisitos para un sistema de gestión de la calidad.
 - d. Características comunes del proyecto.
 - e. Mejora continua del sistema.

- 4) ¿En qué fase del modelo IDEAL se inicia el plan de acción de la mejora de acuerdo con la visión de la organización, el plan de negocio estratégico?**
 - a. Diagnóstico.
 - b. Iniciación.
 - c. Revisión.
 - d. Establecimiento.
 - e. Actuación.

- 5) PSP son las siglas de:**
 - a. Proceso simple programado.
 - b. Programa de software profesional.
 - c. Proyecto simple personal.
 - d. Proceso de Software Personal.
 - e. Programa simple principal.

6) TSP está relacionado con:

- a. Proceso de software de equipo.
- b. Gráficos de control y proceso de decisiones.
- c. Diagrama de árbol y estudio de índice de capacidad.
- d. Diagrama de matriz y histogramas.
- e. Gráficos de control y diagrama de árbol.

7) PEOPLE-CMM se refiere al :

- a. Modelo de planificación de la producción.
- b. Modelo de madurez de capacidad de las personas.
- c. Modelo de planificación de madurez.
- d. Modelo de la capacidad de planificar de las personas.
- e. Modelo de planificación de las personas.

8) El objetivo de la evaluación del proceso es _____ de una organización.

- a. Conocer la capacidad de recopilar datos.
- b. Analizar la capacidad de los procesos.
- c. Conocer la capacidad de los procesos.
- d. Planificar los procesos y la capacidad.
- e. Analizar la recopilación de datos.

9) Es el más extenso y constituye la mayoría del modelo. Proporciona una guía útil para mejorar los procesos, y ofrecer una mayor claridad para la comprensión de los componentes requeridos y esperados.

- a. Material esperado.
- b. Material requerido.
- c. Material de análisis.
- d. Material de diseños.
- e. Material informativo.

10) ¿En qué fase del modelo IDEAL se constituye el punto de partida, en el cual se establece la infraestructura, los roles y responsabilidades que hay que asumir y se asignan los recursos necesarios?

- a. Iniciación.
- b. Almacenamiento.
- c. Diagnóstico.
- d. Actuación.
- e. Revisión.

Resumen

UNIDAD DE APRENDIZAJE IV: EVALUACIÓN Y MEJORA DE PROCESOS

La norma ISO/IEC 90003 (ISO, 2004f) proporciona la guía necesaria en las organizaciones para la aplicación de la ISO 9001 (ISO, 2000b) a la adquisición, suministro, desarrollo, operación y mantenimiento de software y sus servicios relacionados. Identifica todos los aspectos que deberían ser tratados y es independiente de la tecnología, modelos de ciclo de vida, procesos de desarrollo y estructuras organizacionales.

El modelo IDEAL constituye un enfoque usable y entendible para la mejora continua estableciéndolos pasos necesarios que se deben seguir para llevar a cabo un programa de mejora y proporcionando un enfoque ingenieril y disciplinado. El modelo IDEAL está compuesto por cinco fases, cada una de las cuales está formada por una serie de actividades: Iniciación, Diagnóstico, Establecimiento, Actuación y Aprendizaje.

El Proceso de Software de Equipo (TSP), ayuda a conformar equipos para el desarrollo de software de calidad. TSP proporciona un marco de trabajo, que se construye sobre la base de PSP, con fases de desarrollo bien definidas, en las que los productos de software se generan en varios ciclos. El origen de TSP se debe a las limitaciones que PSP tenía en el ámbito industrial. PSP ha tenido un gran éxito en entornos académicos y de hecho los datos obtenidos de los alumnos que han aplicado PSP han sido muy consistentes.

El estándar ISO/IEC 15504 (ISO, 2004a; 2004b; 2004c; 2004d; 2006) proporciona un marco de trabajo para la evaluación de procesos software y establece los requisitos mínimos para realizar una evaluación que asegure la receptibilidad y consistencia de las valoraciones obtenidas. El objetivo de la evaluación del proceso es conocer la capacidad de los procesos de una organización. Como resultado de una exitosa implementación de la evaluación de los procesos se determina la información que caracteriza los procesos evaluados y el punto hasta el cual los procesos realizan su propósito.

Glosario

- ❖ **ADAPTABILIDAD:** Sub característica de portabilidad, que indica las características del software que influyen en las posibilidades de adaptación a diferentes entornos.
- ❖ **CALIDAD:** Conjunto de propiedades y de características de un producto o servicio, que le confieren su aptitud para satisfacer unas necesidades explícitas e implícitas.
- ❖ **CALIDAD DE SOFTWARE:** Es la concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo.
- ❖ **CMMI:** Modelo para la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software.
- ❖ **ISO:** La Organización Internacional de Normalización, conformada por una red de los institutos de normas nacionales de 164 países, con una Secretaría Central en Ginebra (Suiza) que coordina el sistema.
- ❖ **MADUREZ:** Subcaracterísticas de fiabilidad, que indica la frecuencia con que ocurren los fallos.
- ❖ **METAMODELO:** Es un modelo que define el lenguaje para poder expresar un modelo.
- ❖ **MODELO:** Es una abstracción semánticamente consistente de un sistema.
- ❖ **PORTABILIDAD:** Es el esfuerzo requerido para transferir el programa desde un hardware y/o un entorno de sistema de software a otro.
- ❖ **REPOSITORIO:** Es aquel que consta de las tablas y vistas que se utilizan como interfaz con los datos y el código procedural que las maneja. Almacena los detalles del sistema que se está desarrollando

Fuentes de Información

BIBLIOGRÁFICAS:

- ❖ **Piattini Velthuis, Mario G.** Calidad de sistemas informáticos Ed. Ra-Ma, 2010
- ❖ **Coral Calero Muñoz** Calidad del producto y proceso Software Ed. Ra-Ma, 2009.
- ❖ **Rene Ventura B.** Planificación y Evaluación de Proyectos Informáticos. Ed. EUNED, 2010
- ❖ **Roger Pressman.** Ingeniería de software. Ed. Mc Graw Hil, 2009.

ELECTRÓNICAS:

- ❖ **Guía para la aplicación de Estándares de Ingeniería de Software**
<http://www.ie.inf.uc3m.es/grupo/docencia/reglada/ls1y2/ESA/BSSC962-ES.PDF>
- ❖ **Compendio de la ingeniería de software**
http://www.navegapolis.net/files/cis/CIS_1_05.pdf
- ❖ **Calidad de software**
http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad_software.PDF
- ❖ **Gestión de la calidad**
http://www.uhu.es/eyda.marin/apuntes/gesempre/Tema5_1IGE.pdf

Solucionario

UNIDAD DE APRENDIZAJE 1

1. B
2. C
3. A
4. B
5. D
6. A
7. A
8. C
9. A
10. C

UNIDAD DE APRENDIZAJE 2:

1. A
2. D
3. E
4. A
5. B
6. C
7. D
8. E
9. B
10. C

UNIDAD DE APRENDIZAJE 3:

1. C
2. E
3. A
4. D
5. B
6. B
7. D
8. A
9. E
10. C

UNIDAD DE APRENDIZAJE 4:

1. A
2. E
3. C
4. A
5. D
6. A
7. B
8. C
9. E
10. A