

Administración de Proyectos de Software

Calidad en Ingeniería de Software

E. Estévez - P. Fillottrani

Depto. Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur

segundo semestre 2018



Calidad en Ingeniería de Software

Calidad en el Software

Desarrollo Cero Defectos

Normas ISO

CMM y CMMI



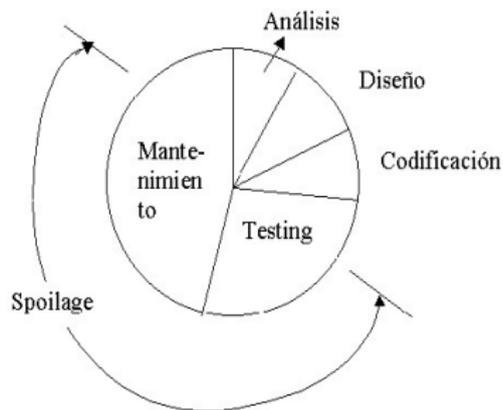
Defectos

- ▶ problema: los errores se aceptan. Esto no es productivo en el tratamiento del problema
- ▶ **defecto**: desviación entre el resultado esperado y el resultado logrado
- ▶ **bug**: algo que se arrastra por propia voluntad en el código y produce problemas. No atribuible a una persona. Involuntario
- ▶ los bugs se pueden aceptar, los defectos no
- ▶ el primer paso para lograr calidad e identificar los defectos como lo que son: **fallas individuales**



Spoilage

- ▶ **spoilage** es el esfuerzo dedicado al diagnóstico y eliminación de fallas que fueron introducidas durante el proceso de desarrollo
- ▶ representa el 55% del costo total del sistema



Spoilage

- ▶ muchos incluyen todo el testing como spoilage
- ▶ ¿por qué incluir tanto de la etapa de mantenimiento como spoilage?
- ▶ muchas de las mejoras de mantenimiento son requerimientos que siempre estuvieron
- ▶ las mejoras se agregan en una razón de productividad que es una o dos veces menor, en magnitud, que la de desarrollo. En la mayoría de los casos esto se debe a errores de diseño
- ▶ **spoilage** es el costo de errores humanos durante el proceso de desarrollo. Errores de codificación, análisis, diseño o testeo



Calidad de Software

- ▶ la calidad del software está dada por la ausencia de spoilage
- ▶ diferentes defectos tienen diferentes costos
- ▶

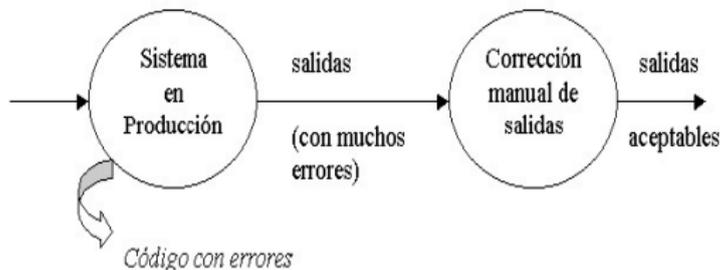
$$\text{calidad} = \frac{\sum \text{costo de diagnóstico y corrección de defectos}}{\text{volumen}}$$

- ▶ valores pequeños implican alta calidad
- ▶ se evita la recolección de datos sobre errores

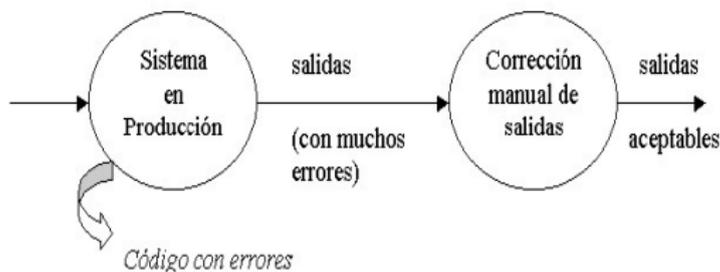


Control de Calidad en el Software

- ▶ los defectos no son un problema tecnológico, son un problema sociológico
- ▶ las medidas y las acciones para mejorar pueden afectar la organización de proyectos, objetivos,...
- ▶ muchas veces los defectos no se corrigen en el lugar adecuado



Control de Calidad en el Software



- ▶ es muy común en la Ingeniería de Software la corrección de defectos en el código, pero no se eliminan del proceso de construcción de sistemas
- ▶ la clave de la promoción de la calidad es remover los defectos del proceso en lugar de removerlos del producto



Control de Calidad en el Software

- ▶ la alternativa de remover defectos es abstenerse de los mismos
- ▶ es el objetivo del **desarrollo con cero defectos**
- ▶ el costo de los defectos constituye de un 35% a un 55% del costo de desarrollo
- ▶ en la mayoría de los proyectos hay personas que introducen más spoilage que el valor de su producción
- ▶ tener una persona con producción pobre en el equipo, es a veces mejor que tener uno más productivo



Control de Calidad en el Software

- ▶ motivos:
 - ▶ las personas que comenten muchos defectos no aparentan ser malos desarrolladores
 - ▶ son muy buenos expertos en remover los errores
- ▶ las personas que luego de medir, resultan que introducen muchos errores, deben reasignarse a otro tipo de tareas
- ▶ la respuesta racional a encontrar amplia variación de aptitudes es especializar



Control de Calidad en el Software

| | Empleado A | Empleado B | A+B |
|------------------------|------------|------------|-------------------|
| costo de codificación | \$ 5000.- | \$ 5000.- | \$ 10000.- |
| costo de testing | \$ 5000.- | \$ 5000.- | \$ 10000.- |
| inserción de defectos | 0.8 d/KLOC | 7 d/KLOC | |
| defectos esperados | 4.8 | 42 | (asum. 6 KLOC) |
| detección de defectos | 0.7 | 0.85 | |
| defectos detectados | 3.36 | 35.7 | |
| costo defectos | \$ 1200.- | \$ 12500.- | (asum. \$ 350.-) |
| spoilage real proyecto | \$ 2400.- | \$ 5000.- | |
| defectos residuales | 1.44 | 6.3 | (asum. \$ 2800.-) |
| spoilage producto | \$ 4032.- | \$ 17640.- | \$ 21672.- |
| spoilage total | \$ 6432.- | \$ 22640.- | \$ 29072.- |



Control de Calidad en el Software

| | Empleado A | Empleado B | A+B |
|------------------------|------------|------------|-------------------|
| costo de codificación | \$ 10000.- | — | \$ 10000.- |
| costo de testing | — | \$ 10000.- | \$ 10000.- |
| inserción de defectos | 0.8 d/KLOC | — | |
| defectos esperados | 9.6 | — | (asum. 6 KLOC) |
| detección de defectos | — | 0.85 | |
| defectos detectados | 8.16 | — | |
| costo defectos | \$ 2850- | — | (asum. \$ 350.-) |
| spoilage real proyecto | \$ 2850.- | — | \$ 2850.- |
| defectos residuales | 1.44 | 6.3 | (asum. \$ 2800.-) |
| spoilage producto | \$ 4032.- | — | \$ 4032.- |
| spoilage total | \$ 6882.- | — | \$ 6882.- |



Control de Calidad en el Software

- ▶ las personas somos diferentes
- ▶ ignorar las diferencias y asignarles tarea de manera homogénea es una de las faltas más importantes del gerenciamiento
- ▶ la medición de defectos es la parte más productiva para la toma de conciencia del trabajo basado en cero-defecto
- ▶ cuestiones relativas:
 - ▶ ¿quién recolecta los datos?
 - ▶ ¿qué datos se deben recolectar y cómo?
 - ▶ ¿cómo se determina la responsabilidad del defecto?



Cuestiones relativas

- ▶ ¿quién recolecta los datos?
 - ▶ personas sin intereses en los resultados: independientes, competentes y confiables
 - ▶ responsabilidad del **grupo de métricas**
- ▶ ¿qué protocolo debe gobernar el uso de los datos de defectos?
 - ▶ debe ser público, garantizando el correcto uso de los mismos
 - ▶ los datos recolectados en bases individuales, sólo pueden ser usados para beneficios del individuo. Motivar la propia auto-estima
 - ▶ los únicos datos que se pueden compartir, son datos globales



Cuestiones relativas

- ▶ ¿qué datos deben ser recolectados?
 - ▶ comienzan con la detección del defecto. Se registra y se le asigna un número
 - ▶ defecto-Nro + detección-Fecha + defecto-Síntoma + detector-Nombre + (Horas + Fecha + Empleado-Nro + Tarea-Tipo) + eliminación-Fecha + costo-Total + defecto-Causa + costo-Responsabilidad + responsable
- ▶ ¿cómo asignar la responsabilidad?
 - ▶ no debe ser ubicada, debe aceptarse
 - ▶ defecto de diseño: diseñador + programador



Cuestiones relativas

- ▶ ¿cómo deben recolectarse los datos?
 - ▶ es una tarea independiente. Un defecto es considerado como tal si es detectado por una persona fuera del equipo de desarrollo
 - ▶ el testing debe realizarlo un grupo fuera del equipo de desarrollo
- ▶ hay que considerar el **factor humano**:
 - ▶ se solicitan controles de calidad
 - ▶ no se objeta la medición, se está en contra de las malas mediciones



Mejorando la Calidad del Software

- ▶ un producto de baja calidad es signo de un testing pobre
Resultado: Se invierte más en testing
- ▶ estadísticas:
 - ▶ proyectos por encima de la media: inversión de 20.6% en testing
 - ▶ proyectos por debajo de la media: inversión de 28.4% en testing
- ▶ los principales determinantes de la calidad están principalmente en el software antes de que comience el testeo



Mejorando la Calidad del Software

- ▶ no importa la cantidad de testing, sólo se podrá remover un **porcentaje** de los defectos presentes (alrededor del 50%)
- ▶ lo que varía es la incidencia de los errores en el código no testeado
- ▶ mejorar la calidad del código testeado es reducir la inserción de defectos en el código antes de su testeo



Mejorando la Calidad del Software

- ▶ los diferentes métodos usados en entornos con pocos defectos no son más importantes que incentivos diferentes
- ▶ los incentivos están fuertemente dirigidos a mejorar la calidad antes del testing
- ▶ **separar el testing**: se trata de lograr un código con cero-defecto antes de testearlo. Ultima posibilidad de acomodar la auto-estima. Se tienden a dedicar más tiempo a lo que más les gusta. Probablemente sea el testing. El problema es que no se separa del todo. En muchos casos sólo pruebas de usuario o de integración



Mejorando la Calidad del Software

- ▶ ¿por qué no se hace?
 - ▶ la calidad no es el interés de los proyectos de desarrollo
 - ▶ actitud política relacionada con costos
 - ▶ separar en parte el testing tiene efectos no deseados: no tiene incidencia en la proporción de inserción de defectos
 - ▶ a la mayoría de los programadores les gusta el diagnóstico del defecto
- ▶ separar los tres grupos con distintos objetivos: **desarrollo**, **testing**, **métrica**. Grupos adversarios para lograr mejores resultados



Mejorando la Calidad del Software

- ▶ el testing del **grupo negro**: misión no es el diagnóstico sino la detección
- ▶ herramientas para el grupo de desarrollo: Inspección formal del código
- ▶ siembra de defectos: motivación para el grupo de testing - documentación de los defectos plantados - estadísticas del tipo de defectos
- ▶ incentivos adversarios: pagar por reducción de spoilage



Mejorando la Calidad del Software

- ▶ desarrollando habilidades de calidad:
 - ▶ armar grupo de entrenamiento
 - ▶ tomar código depurado e insertar defectos
 - ▶ los programadores buscan defectos mediante inspección
- ▶ el juego de la **guerra de código**: juego organizado por Yourdon Inc
 - ▶ dos grupos a desarrollar
 - ▶ se intercambian desarrollos y se prueba lo del otro grupo
 - ▶ se calcula spoilage y densidad de defectos
 - ▶ se determina un vencedor



Desarrollo Cero Defectos

- ▶ no sólo aceptamos el error sino que nos anticipamos a él
- ▶ asumimos que las personas tenemos un factor de error interno, en lo laboral pero no en lo personal
- ▶ la mejor política es la prevención de defectos. Con la prevención no se necesita reparar, examinar, explicar,...
- ▶ el primer paso es tomar una actitud de prevención de defectos: **cero defecto**



Desarrollo Cero Defectos

- ▶ en la programación el objetivo es producir los módulos principales libres de defectos
- ▶ IMS (IBM): 425 módulos con más de 300 módulos libre de defectos. 7.3% de los módulos con el 57% de los defectos
- ▶ OS/360: 47% de una clase de defectos se encontraba en el 4% de los módulos
- ▶ la política es la eliminación anticipada de los módulos con defecto



Desarrollo Cero Defectos

- ▶ la probabilidad de que existan más errores en la sección de un programa es proporcional al número de errores ya encontrados en esa sección (Myers)



Desarrollo Cero Defectos

- ▶ se puede dibujar la curva en cualquier punto del proyecto. Ej: testing unitario, testing de integración, testing de aceptación. Siempre con la misma inclinación
- ▶ los defectos tempranos son un predictor fuerte de defectos posteriores
- ▶ no gastar “limpiando” código que ha demostrado ser susceptible de defectos. Liberarse de él
- ▶ el costo de codificar es irrelevante comparado con el costo de limpiar los defectos de un módulo



Desarrollo Cero Defectos

- ▶ fijar una cota de tolerancia de defectos. Cuando un módulo la supere, recodificarlo. Se debe asignar la re-codificación a otra persona
- ▶ se debe aplicar esto desde el proceso de compilación
- ▶ comenzar a medir los defectos:
 - ▶ defectos durante proyecto .vs. defectos luego de instalación
 - ▶ defectos tests unitarios .vs. defectos de tests de integración
 - ▶ defectos primeros 6 meses de spoilage .vs. defectos spoilage posterior
 - ▶ defectos de compilación .vs. el resto de defectos detectados



Programa de Desarrollo Cero Defectos

1. enunciar el objetivo de 0 defecto para productos terminados y subproductos
2. separar organizativamente: desarrollo, testing, métricas
3. el producto que se pasa de desarrollo a testing es no compilado
4. los errores de compilación se consideran defectos. Cada compilación no satisfactoria es 1 defecto
5. los defectos de compilación son un feedback para el proceso de inspección
6. repetidos fracasos en compilación es un indicador para rechazar el módulo



Programa de Desarrollo Cero Defectos

- ▶ motivar las inspecciones con feedback de defectos de compilación
- ▶ construir alta calidad en software es costoso, la baja calidad también
- ▶ el costo de construirlo correctamente por primera vez es menor que corregirlo
- ▶ aplicar programas de mejoramiento de calidad a sistemas en los cuales la calidad sea relevante
- ▶ se puede aplicar a parte del proyecto, no a la totalidad



ISO

- ▶ la **International Organization for Standardization (ISO)** tiene como objetivo proveer una forma clara de estructurar las operaciones para potenciar al personal (empower), reducir papelería y burocracia, satisfacer a los clientes y sobrevivir en el negocio a largo plazo
- ▶ la calidad es un estado de la mente y puede desarrollarse mejor por las personas motivadas que por las personas desalentadas
- ▶ se entiende por **calidad** a la totalidad de rasgos y características de un producto o servicio que se basan en su habilidad de satisfacer necesidades enunciadas o implícitas



Calidad Total

- ▶ calidad significa satisfacer al cliente
- ▶ está totalmente definida por las percepciones del cliente
- ▶ en la definición, no hay espacio para alta o baja calidad
- ▶ no importa lo que nosotros pensemos, lo único que importa es lo que piensa el cliente
- ▶ el desafío es que las percepciones del cliente cambian constantemente
- ▶ demanda permanentemente más y mejores productos y servicios. La organización debe ser lo suficientemente rápida y flexible para satisfacer estas demandas
- ▶ **Total Quality Management (TQM)** administración de calidad total



ISO 9000

- ▶ la **administración de calidad** (QM), especificada en ISO 9000, es sobre cómo administrar una organización con el objetivo de satisfacer las necesidades de los clientes
- ▶ los beneficios son financieros y personales
- ▶ bases:
 - ▶ mejoras continuas en la forma de trabajar
 - ▶ aprender de los errores
 - ▶ identificar necesidades de cambio
 - ▶ que los procesos sean proactivos en lugar de reactivos
 - ▶ lograr implementación rápida de los cambios
 - ▶ tener confianza en los cambios implementados
- ▶ las revisiones constructivas proveen información necesaria para que las personas mejoren



ISO 9000

- ▶ se respetan estándares y prácticas, se identifican debilidades y se corrigen rápidamente, se establece un entorno de mejora continua
- ▶ todos los procesos son visibles - los entregables son definidos y controlados, y siempre se conoce su estado
- ▶ los cambios son controlados
- ▶ la calidad de los productos y servicios está asegurada a través de procesos que los verifican con estándares predefinidos. Se validan con sus especificaciones que son conocidas anticipadamente por todos los involucrados



Normas ISO 900X

- ▶ la norma **ISO 9001** es un modelo para asegurar la calidad en diseño, desarrollo, producción, instalación y servicio
- ▶ ANSI/ASQC ISO 9001 en EEUU, ISO 9001 en Europa, BS/EN ISO 9001 en Inglaterra
- ▶ es el estándar en el cual una empresa puede certificar por el diseño, desarrollo, intalación y mantenimiento de productos y servicios tales como desarrollo de software, operación y soporte
- ▶ hay dos subconjuntos de estos estándar: ISO 9002 e ISO 9003
- ▶ aquellas organizaciones que no realizan sus propios diseños pueden aplicar ISO 9002



Normas ISO 900X

- ▶ **ISO 9002** es un modelo para asegurar la calidad en producción, instalación y servicio
- ▶ **ISO 9003** es un modelo para asegurar la calidad en inspecciones finales y testeo
- ▶ **ISO 9000.3** define administración de calidad y elementos del sistema de calidad. Guía para desarrollo, provisión y mantenimiento de software
- ▶ **ISO 9004.4** es una guía para mejoramiento de la calidad



Normas ISO 900X

- ▶ **ISO 9004.2** administración de calidad y elementos del sistema de calidad - Guía para servicios. Puede ser aplicada para servicios hospitalarios (hoteles, entretenimientos, servicios turísticos,...), comunicaciones, salud, mantenimientos (eléctricos, mecánicos, computación,...), técnicos (consultoría,...), científicos (investigación, desarrollo, ...)
- ▶ **ISO 10011**: guía para auditar sistemas de calidad
- ▶ **ISO 10013**: guía para desarrollar manuales de calidad



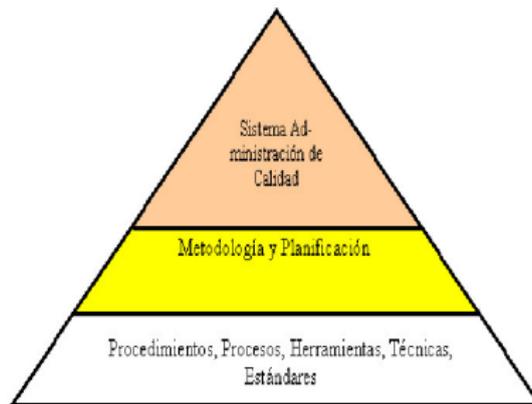
Objetivos de las normas ISO

- ▶ la forma de hacer negocios se basa en establecer buenas relaciones con proveedores y clientes
- ▶ permiten establecer interfaces de sistemas comunes para reducir conflictos y promover la aceptación de productos y servicios
- ▶ se plantea la necesidad de definir interfaces comunes y luego auditar cada uno los sistemas del otro para su aceptación
- ▶ las normas ISO 9000 permiten acordar una especificación de interfaces comunes. Asesoramiento independiente que garantiza la eficiencia de los sistemas de terceros con los cuales una organización debe relacionarse



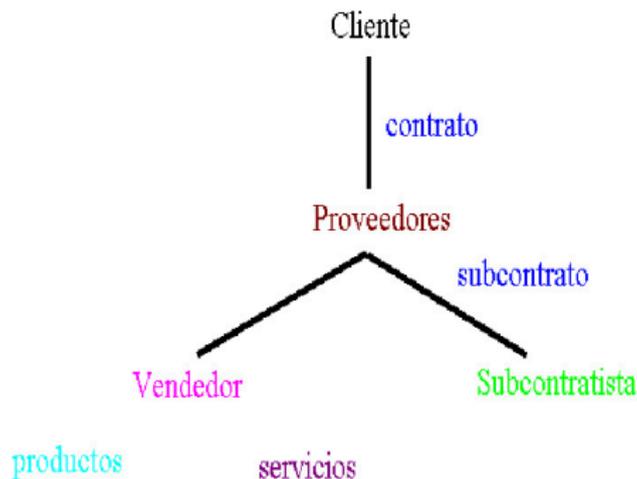
Sistemas de Administración de Calidad

- ▶ define cómo un QMS supervisa el desarrollo, las operaciones y el soporte día a día, para asegurar que las tareas se realizan correctamente haciendo uso de procesos, procedimientos, métodos, guías predefinidas



Sistemas de Administración de Calidad

- relaciones entre las diferentes partes involucradas en un sistema basado en ISO 9000



Sistemas de Administración de Calidad

- relaciones entre la documentación en un QMS



ISO 9001 en Software

- ▶ define un conjunto mínimo de requerimientos para administración de calidad. Incluye:
 - ▶ garantía de calidad
 - ▶ control de calidad
 - ▶ control de procesos
 - ▶ administración de cambios
 - ▶ reporte de problemas
 - ▶ revisiones formales
- ▶ **sistema de administración**: colección de requerimientos que deberían ser necesarios para implementar un sistema de administración de calidad para cualquier tipo de operación



Sistema de Administración de Calidad

Sistema de Administración
de Calidad

responsabilidad de la gerencia
auditoría de calidad
registro de control de calidad
control de datos y documentos
entrenamiento
acciones correctivas y preventivas



Responsabilidad de la Gerencia

- ▶ debe formalmente comprometerse con la calidad enunciando políticas de calidad
- ▶ es un documento que debe ser firmado por los gerentes involucrados. Debe incluir un enunciado de compromiso y los servicios y productos alcanzados
- ▶ debe haber una estructura organizativa que describa cómo la calidad será implementada en la organización
- ▶ se definen los roles y responsabilidades de todos los miembros
- ▶ los procedimientos de revisión de administración deben estar ubicados para asegurar el desarrollo adecuado y efectivo del sistema de administración de calidad



Sistema de Calidad

- ▶ los procesos usados para desarrollar y entregar productos y servicios deben estar documentados
- ▶ incluye describir manuales de calidad, procedimientos de calidad, métodos y estándares
- ▶ debe existir planes de calidad que describan cómo se logran los objetivos de calidad
- ▶ el plan de calidad puede ser “basado en proyectos” como en desarrollo de software o “basado en procesos” donde se usan procesos repetitivos



Control de Datos y Documentos

- ▶ es obligatorio controlar todos los documentos y datos que constituyen el QMS. También los documentos y datos que controlan el desarrollo del producto, incluyendo la especificación de requerimientos y la documentación de diseño
- ▶ la documentación de políticas y procesos debe estar autorizada, tener una lista de distribución y tener control de cambios formal
- ▶ la documentación del usuario (guías y documentación de operación) debe ser controlada usando el mismo procedimiento, pero se considera un producto



Auditoría Interna de Calidad

- ▶ el QMS debe tener procedimientos que provean revisiones periódicas y sistemáticas de todas sus operaciones en orden a garantizar su efectividad continua
- ▶ cuando se detectan discordancias con el QMS, se deben tomar acciones correctivas
- ▶ la revisión del QMS debiera indicar que todo el trabajo se realiza usando el sistema y que esto agrega valor en lugar de entorpecerlo



Acciones Correctivas y Preventivas

- ▶ los defectos, problemas o deficiencias pueden identificarse en el software, en la documentación, en el entorno de desarrollo, en el QMS, en la metodología adoptada
- ▶ deben revisarse periódicamente los procesos, procedimientos, entornos, métodos, estándares para identificar mejoras potenciales o debilidades
- ▶ se deben tomar acciones preventivas de tal manera que los problemas potenciales no se conviertan en problemas actuales
- ▶ se deben considerar encuestas de clientes, feedback del personal, y otras fuentes de información



Registro de Control de Calidad

- ▶ se requiere: identificar, juntar, catalogar, archivar, retener, mantener, y disponer todos los registros referentes a QMS incluyendo actividades de desarrollo, mantenimiento y soporte
- ▶ los registros deben administrarse adecuadamente de tal manera que sea fácilmente consultarlos
- ▶ los registros serán revisados durante auditorías internas y externas



Entrenamiento

- ▶ identificar las necesidades de capacitación de todo el personal.
- ▶ requiere que todo el personal esté capacitado en herramientas técnicas, y lenguajes usados para el desarrollo, tanto como en QMS
- ▶ los registros de capacitación proveen evidencias que el personal asignado a tareas específicas está calificado para realizar esas tareas



Sistema de Administración de Productos

- ▶ es una colección de requerimientos relacionados con el control y trazabilidad de los productos
- ▶ incluye:
 - ▶ control de productos desarrollados
 - ▶ control de provistos por el cliente
 - ▶ control de productos comprados
- ▶ también incluye la administración de productos defectuosos y el control de stocks
- ▶ varios de estos requerimientos no son aplicables a desarrolladores de software



Sistema de Administración de Productos

Sistema de Administración
de Productos

{ control de provistos por el cliente
control de productos que no satisfacen calidad
control de productos desarrollados in-house
control de comprados y subcontratados
administración de almacenaje/entrega



Administración de Productos

- ▶ los documentos entregables y programas deben estar identificados y administrados, de tal manera que todas las partes interesadas sepan que tienen el producto correcto y puedan ser advertidos de cualquier cambio y/o deficiencia
- ▶ las normas ISO tienen en cuenta los cambios a documentos y a componentes de software
- ▶ los documentos pueden ser controlados con los mismos procedimientos de QMS para documentos y datos de control. Ej: Identificación y proceso formal de control de cambios



Control de Productos Provistos por el Cliente

- ▶ es necesario asegurar que la información y material provisto por el cliente es adecuado para su uso y que el cliente acepta las responsabilidades por la exactitud de la información provista
- ▶ se especifican los procedimientos requeridos para asegurar que:
 - ▶ la información y material suministrado por el cliente es correcto
 - ▶ los elementos son protegidos mientras están en poder de la empresa
 - ▶ los elementos son mantenidos por la empresa adecuadamente



Compras y Manejo

- ▶ se debe tener un proceso de evaluación y selección de proveedores
- ▶ es necesario que todos los desarrollos subcontratados satisfagan los requerimientos del contrato entre la empresa y el cliente
- ▶ es un beneficio importante si el proveedor tiene implementado un sistema de administración de calidad

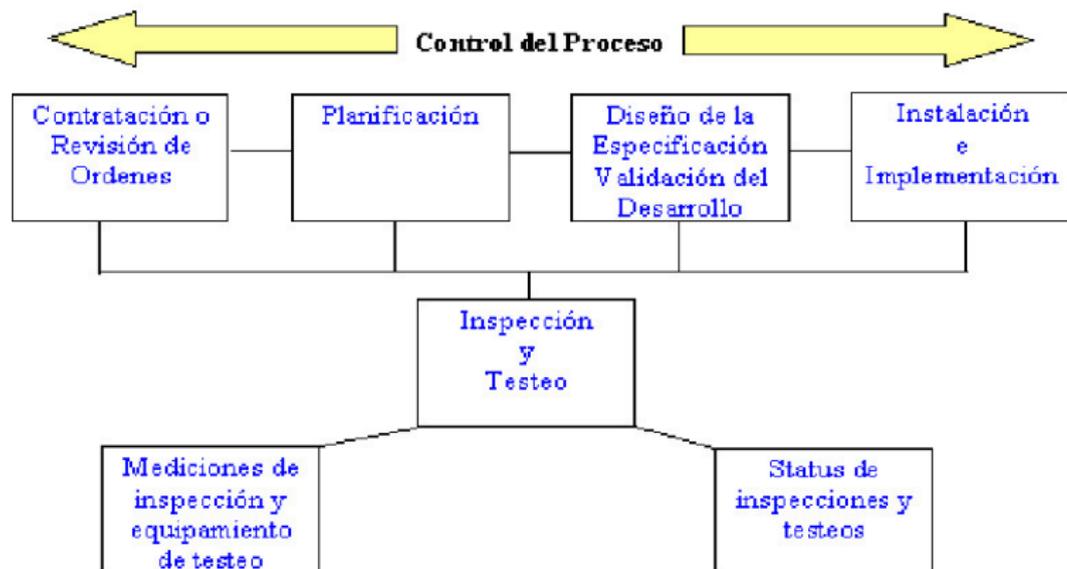


Almacenaje, Preservación y Entrega

- ▶ requerimientos para asegurar que los documentos y el software sean manejados y almacenados de tal manera que el cliente reciba el software y los documentos adecuados



Administración del Desarrollo



Administración de Desarrollo

- ▶ es un conjunto de requerimientos que incluye IEEE 1298 + ISO 9001
- ▶ estructurado para satisfacer los estándares de un proyecto de desarrollo de software
- ▶ incluye:
 - ▶ objetivos del ciclo de vida
 - ▶ inspección y testeo
 - ▶ administración del proyecto
 - ▶ no hay requerimientos de seguir un determinado ciclo de vida
 - ▶ los entregables son determinados por la metodología



Revisiones de Contratos

- ▶ las propuestas y presupuestos conducen a contratos y definen las restricciones de un proyecto
- ▶ la revisión de los mismos debe ser incluida en un QMS
- ▶ se revisan los contratos para asegurar que existen los perfiles, habilidades, recursos, y capacidad para satisfacer las necesidades del usuario



Control de Procesos y Planeamiento de Proyectos

- ▶ se debe crear un plan de calidad para cada proyecto. Debe incluir:
 - ▶ información de entregables
 - ▶ puntos de revisión
- ▶ un proyecto no debe proseguir hasta que no esté terminado su plan de calidad
- ▶ el proceso de producción de software está cubierto en ISO por un **proceso especial**, en el cual los defectos en los entregables del proceso no se detectan hasta que el producto está en uso



Control de Procesos y Planeamiento de Proyectos

- ▶ los requerimientos de la norma abarcan:
 - ▶ especificación de requerimientos
 - ▶ diseño
 - ▶ entorno de desarrollo: software de base, sistemas operativos, compiladores, editores, DBMS,...)
- ▶ los estándares debieran incluir estándares de programación, de diseño, codificación, testeo, preparación de documentación, capacitación



Administración de Requerimientos

- ▶ no se refiere específicamente a la especificación de requerimientos porque asume su definición en el contrato
- ▶ IEEE 1298 requiere una aceptación formal de la especificación de requerimientos



Control de Diseño y Desarrollo

- ▶ fuerza la revisión formal del diseño, la programación y los documentos del usuario
- ▶ es esencial que el input al diseño sea el adecuado, las metodologías y técnicas estén definidas, y se sigan los procedimientos



Inspección y Testeo

- ▶ obliga a las inspecciones formales y al testing de productos
- ▶ el testing debe incluir:
 - ▶ preparación y revisión del plan de test
 - ▶ preparación y revisión de los datos de test
 - ▶ revisión de resultados esperados
- ▶ se debe definir tests a los productos provistos por el cliente, productos comprados, herramientas y técnicas, inspecciones, mediciones y equipamiento de testing



Estado de Inspección y Test

- ▶ el estado de un producto de software bajo desarrollo, corresponde al estado de inspección y test de los productos manufacturados
- ▶ apunta a los medios para identificar el estado de los productos en desarrollo



Control de Inspección, Mediciones y Equipamiento de Test

- ▶ son los requerimientos de políticas y procedimientos para asegurar que el equipamiento de testing es adecuado para el testing del producto
- ▶ ejemplo: Programas especiales para generar datos de prueba, simuladores



Técnicas Estadísticas

- ▶ no se usan para probar que el software funciona
- ▶ si las estadísticas se usan entonces las técnicas deben ser estadísticamente correctas y el mecanismo de muestreo debe ser válido
- ▶ las métricas no son técnicas estadísticas a menos que se usen para aceptación de productos
- ▶ el uso de métricas debe ser considerado y documentado en este punto
- ▶ se debería consultar la ISO 9126 para términos definidos en métricas de software



Mantenimiento de Software y Servicio

- ▶ para asegurar que el mantenimiento del software es realizado correctamente se necesita definir políticas y procedimientos
- ▶ se considera **mantenimiento** a la rectificación de defectos y cambios funcionales menores
- ▶ el mantenimiento se debe realizar con los mismos controles de calidad que el desarrollo original
- ▶ como **servicio** se incluye mesa de ayuda, atención al cliente y servicios provistos por técnicos especializados



Objetivo de la Administración e ISO 9001

- ▶ las organizaciones se enfrentan a dilemas aparentemente conflictivos
- ▶ existe una demanda creciente por parte de los individuos y los grupos de reconocimiento, autonomía y poder de decisión
- ▶ la necesidad de satisfacer requerimientos de calidad a medida que las necesidades del cliente aumentan
- ▶ existe la necesidad de ser flexible, y tener la capacidad de dar respuestas rápidas en la producción de productos y servicios individualizados y personalizados para cada cliente



Objetivos de la Administración e ISO 9001

- ▶ ¿cómo se ven los **Sistemas de Administración Formalizados (SAF)** ?
 - ▶ justificación para una continua reglamentación, deshumanización y automatización de personas trabajando bajo procedimientos
 - ▶ liberación y habilitación de personas para que desarrolle lo mejor
- ▶ el resultado estará dictado por la actitud de la gerencia. Los beneficios también
- ▶ no hay conflictos entre SAF bien diseñados y trabajo creativo, con enfoques abiertos, movimientos rápido y mínimo trabajo en papel
- ▶ en desarrollo de software las personas trabajan en contacto directo con el cliente. Necesidad de enfocar procesos orientados al cliente



Recursos Humanos

- ▶ ¿cuál es el objetivo? ¿qué se está tratando de lograr? ¿qué estamos tratando de hacer entre todos?
- ▶ si las personas conocen las reglas, qué se espera de ellas, y la cultura de la organización, pueden contribuir a su desarrollo y mejoramiento
- ▶ si todas las personas están motivadas y trabajan en colaboración, se está logrando un recupero de la inversión en personal en lugar de pagarles para que atiendan nuestras premisas por 40 horas semanales
- ▶ **empowerment**: propuesta de productividad que disminuye gerencias medias permitiendo al personal trabajar sin dirección y control permanente



Recursos Humanos

- ▶ el empowerment se logra si las personas conocen muy bien cuál es su trabajo: ¿cómo se hace? ¿qué se espera de ellas?
- ▶ en el desarrollo de software se es **totalmente dependiente** de las personas que se contratan
- ▶ características: buenas relaciones interpersonales, motivadas, creativas, productivas, pensantes, pensantes, pensantes,
- ▶ las personas no alcanzan el límite de sus capacidades, alcanzan el límite de lo que se espera de ellas



Recursos Humanos

- ▶ se alienta a las personas a dar el máximo de su intelecto, no limitarse por preconceptos
- ▶ se usan las oportunidades para fomentar diversión, iniciar nuevas ideas, y mantener el diálogo
- ▶ las evidencias muestran que se logran buenos resultados con organizaciones de 2 a 3 personas trabajando en colaboración facilitando el surgimiento de nuevas ideas
- ▶ en esta línea se deben evitar ambientes institucionalizados. La creatividad se fomenta con entornos distendidos



Recursos Humanos

- ▶ para que las personas sepan cómo hacer su trabajo se necesita capacitación permanente
- ▶ se las debe alentar a la mejora permanente sobre cómo realizar su trabajo
- ▶ se debe alentar a todos los grupos con la responsabilidad de atender al cliente y con los derechos a exigir cambios cuando el sistema no cumpla con el servicio al cliente
- ▶ y colocar mas confianza en la auto-determinación y menos en el cumplimiento de reglas



Recursos Humanos

- ▶ se fomenta el principio que cuando se ve un problema se arregla, o se tiene la responsabilidad del seguimiento
- ▶ si el problema no se arregla, que no estén atemorizados por presentar el problema y los motivos de fracaso de la solución intentada
- ▶ se asegura que el sistema se mejora como resultado de la experiencia
- ▶ motivar a la gente para que haga cosas en lugar de que hable de cosas



Recursos Humanos

- ▶ se proveer acceso a la mayor cantidad de información de tal manera que puedan tomar decisiones con conocimiento
- ▶ los proveedores de servicio deben responder instantáneamente a cualquier situación. Se requiere tomar la decisión correcta, la primera vez, todo el tiempo
- ▶ se debe dar reconocimiento al personal
- ▶ no quebrar el grupo.
- ▶ proveer información cuando se descubre un punto débil: quién lo encontró, cómo se solucionó
- ▶ siempre con actitud positiva: un problema es la oportunidad de eliminar un defecto y hacer un mejor trabajo la próxima vez.



Recursos Humanos

- ▶ hay que ser cuidadoso con las evaluaciones anuales de desempeño y sistemas de medición de performance
- ▶ agregarle a las estadísticas las consideraciones de valor necesarias
- ▶ ser flexible en los arreglos laborales. Mantener la organización chata y mantener la burocracia afuera
- ▶ los grupos de trabajo debe ser auto-gerenciados. El grupo debe reconocer y atender muchos de sus problemas sin intervención
- ▶ debe realizar la mayoría de las tareas de gerenciamiento por que percibe la necesidad de hacerlo en orden a sobrevivir



Procesos

- ▶ **proceso**: la aplicación de métodos, procedimientos, instrucciones de trabajo y estándares para lograr un resultado
- ▶ **objetivo de un proceso**: construir calidad en un producto. No probar e inspeccionar la calidad en el producto terminado
- ▶ chequear el trabajo dentro del grupo unos con otros ni bien se termina. Se hace sin registros formales. Provee feedback rápido evitando que se propaguen errores. Luego pasa al control formal de calidad y finalmente a inspecciones formales de los entregables



Procesos

- ▶ los procesos deberían ser definidos por las personas que los deben realizar
- ▶ de esta manera se minimiza la burocracia y papeles y se fomenta la mejora continua
- ▶ el temor al fracaso es la mayor amenaza a la mejora continua. Si no se cometen errores, no se está progresando
- ▶ cuando se cambian procesos tener siempre en mente: ser cuidadoso de no rechazar las formas simples.
- ▶ lo complejo y grande es la antítesis de exitoso. Procesos simples y flexibles y sujetos a mejoras continuas (medidas correctivas y preventivas)



Procesos

- ▶ una de las barreras mas grande a la productividad y a la mejora es la burocracia
- ▶ para evitarla se debe crear una aversión a ella, especialmente a nivel de la gerencia media
- ▶ el entorno correcto para delegación incluye:
 - ▶ especificar roles y responsabilidades de las personas
 - ▶ especificar requerimientos de los procesos
 - ▶ especificar objetivos y estándares
- ▶ muchas veces el fracaso en la implementación de sistemas formalizados en ISO 9001 es la mala comprensión de los mismos y es un muy buen negocio para las grandes consultoras



Procesos

- ▶ los procesos documentados adecuadamente deberían reducir supervisión
- ▶ no escribir procedimientos que traten de pensar por las personas
- ▶ los procedimientos no se pagan por peso: se debe escribir corto, claro, amigable, fácil de seguir
- ▶ los procedimientos escritos deben servir para mejorar los procesos y no para defender posiciones y frenar cambios



Administración de Calidad Total y CMM

- ▶ el ciclo de revisión de gerencia, acciones correctivas y preventivas, y auditoría soportado por entrenamiento, registros de calidad y un QMS documentado garantiza un sistema de administración poderoso, auto-correctivo y con mejoras continuas
- ▶ **capability maturity code** (CMM) del Carnegie Mellon University's Software Engineering Institute es un modelo que provee una medida razonable repetible de la madurez del proceso de desarrollo de software en una organización
- ▶ ISO 9001 y CMM son complementarios. ISO 9001 es usado como la estructura y CMM define los requerimientos de los procesos en mas detalle



Planeamiento

- ▶ el planeamiento se ve muchas veces como una pérdida de tiempo
- ▶ muchos desarrolladores todo lo que hacen es una hoja con un Diagrama de Gantt
- ▶ es imperativo planificar de tal manera de saber cómo se van a satisfacer los requerimientos antes de comenzar con los detalles del trabajo
- ▶ tampoco se debe planificar eternamente....
- ▶ los planes también se hacen para soportar cambios (incluir planificación sobre su propia actualización). La falta de visibilidad no debe evitar la planificación



Planeamiento

- ▶ un plan debe ser modificado ni bien comienza a perder relevancia. Un proyecto se atrasa de a un día por vez
- ▶ los planes deben identificar los objetivos y cómo vamos a lograrlos
- ▶ habiendo planificado se deben cumplir los cronogramas y fechas límites y constantemente revisarlas
- ▶ la planificación también sirve para identificar oportunidades para optimizar tiempos e interacciones y solapamientos entre tareas
- ▶ ISO 9001 es claro en la necesidad de planificar. IEEE 1298 es específico e incluye qué se debe tener en cuenta en la planificación. ISO 9000.3 incluye listas de subplanes y objetivos de planificación



Producto

- ▶ los clientes demandan que los productos y servicios satisfagan exactamente sus necesidades
- ▶ si no se logra la velocidad de innovación necesaria, el cliente se mudará a un competidor
- ▶ para permanecer compitiendo se necesita ofrecer mas servicios con productos personalizados. Los mas exitosos proveen a sus clientes con resultados totalmente sintonizados con sus necesidades
- ▶ se necesita asegurar que existen en los procesos los mecanismos de feedback necesarios para permanecer en contacto con el cliente



Clientes

- ▶ política de satisfacción de clientes:
 - ▶ el cliente es el visitante mas importante en nuestras premisas
 - ▶ no depende de nosotros, nosotros dependemos de él
 - ▶ no es una interrupción en nuestro trabajo, es el propósito del mismo
 - ▶ no es un extrañ o en nuestro negocio, es parte de él
 - ▶ no le estamos haciendo un favor al servirlo, el nos hace un favor al darnos la oportunidad de hacerlo



Clientes

- ▶ el mensaje de servicio al cliente debe ser permanentemente comunicado al personal
- ▶ los procesos deben alentar la innovación y visión de servicio al cliente en lugar de favorecer productos estándares
- ▶ los programas de mejoramiento de calidad deben conducir a reducir costos, mejorar la atención a clientes y aumentar las ventas
- ▶ un cliente insatisfecho, son diez potenciales clientes perdidos
- ▶ el principal problema es que pocos comunican su insatisfacción



Clientes

- ▶ el cliente es el motivo de la existencia de las organizaciones
- ▶ en el modelo de servicios al cliente se debe repensar las organizaciones:
 - ▶ los directores existen para dar soporte a la gerencia
 - ▶ el rol de la gerencia es soportar a los equipos de trabajo
 - ▶ los equipos soportan a los clientes
- ▶ el soporte al cliente incluye entregar productos y servicios donde y cuando el cliente desee
- ▶ estar atento al crecimiento de tal manera que esto no ocasione disminución en la calidad del servicio



Mejoras

- ▶ es importante que las organizaciones crezcan y mejoren. El estancamiento significa retroceder
- ▶ el hacer algo siempre tendrá algún beneficio
- ▶ al comenzar a hacer, no intentar un enfoque big bang
- ▶ ir despacio con pequeños incrementos: reduce riesgos, acelera cambios, minimiza desorganización
- ▶ promover estilos consensuados
- ▶ liderar y mostrar el camino, pero involucrando al personal



Mejoras

- ▶ uno de los factores mas importantes en la administración de cambios es la administración del recurso más crítico: el tiempo
- ▶ esto también es aplicable a acuerdos y negociaciones
- ▶ el trabajo es librarse de las cosas que están entre la organización y el lugar donde se desea colocarla: prácticas, procedimientos y procesos obsoletos
- ▶ todo el personal en la organización debe conocer las ventajas competitivas con las que se opera
- ▶ delegar tareas: hacer lo que uno sabe que hace bien



Mejoras

- ▶ comenzar a gerenciar caminando. Analizar que es lo que nos ata al escritorio versus ir a ver que es lo que está pasando
- ▶ ser impaciente con reuniones y burocracia
- ▶ alentar el rechazo a demoras o pensamientos negativos
- ▶ descubrir los procesos que contribuyen a la organización
- ▶ descubrir por qué los gerentes cobran sus sueldosa



Mejoras

- ▶ los gerentes deben permanentemente focalizar las oportunidades para mejorar y esto se puede hacer conociendo y comprendiendo el fracaso
- ▶ asegurarse que los gerentes aceptan que cuando las cosas van mal es su responsabilidad
- ▶ se debe motivar a las personas que informan las debilidades y fracasos a los gerentes
- ▶ muchas oportunidades de mejora se pierden cuando los gerentes matan al mensajero



Sistema de Administración de Calidad

- ▶ el sistema de administración de calidad ISO 9000 es un sistema de administración formal que es parte de la cultura de la organización
- ▶ no es un agregado que puede conectarse o desconectarse
- ▶ la implementación de este sistema requiere compromisos, acciones y participación
- ▶ no es una actividad nocturna
- ▶ si se cumplen las normas ISO, se puede tener evidencia independiente para probar a clientes que el sistema de calidad opera eficientemente
- ▶ esta tarea la realizan agencias de certificación acreditadas



Sistema de Administración de Calidad

- ▶ la responsabilidad de la gerencia es fundamentalmente proveer la infraestructura dentro de la cual la organización pueda crecer y prosperar
- ▶ la auditoría de los sistemas de calidad es principalmente encontrar oportunidades para obtener mayores beneficios del sistema mejorando las cosas
- ▶ las acciones correctivas y preventivas buscan usar cada problema y problema potencial como una oportunidad para mejorar



ISO 9126

- ▶ la **ISO 9126** está basada en el modelo de Mc Call
- ▶ define un modelo que permite evaluar y comparar la calidad de productos de software sobre la misma base
- ▶ la calidad queda definida como un atributo de alto nivel, definida por:
 - ▶ funcionalidad
 - ▶ fiabilidad
 - ▶ usabilidad
 - ▶ portabilidad
 - ▶ mantenibilidad
 - ▶ eficiencia



Funcionalidad

- ▶ un conjunto de atributos que muestra la existencia de un conjunto de funciones y sus propiedades especificadas
- ▶ las funciones son aquellas que satisfacen necesidades enunciadas o implícitas
- ▶ está compuesta por:
 - ▶ adecuación
 - ▶ exactitud
 - ▶ interoperabilidad
 - ▶ seguridad de acceso
 - ▶ cumplimiento funcional



Funcionalidad

- ▶ **adecuación** es la capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados
- ▶ **exactitud** es la capacidad del producto software para proporcionar los resultados o efectos correctos o acordados, con el grado necesario de precisión
- ▶ **interoperabilidad** es la capacidad del producto software para interactuar con uno o más sistemas especificados



Funcionalidad

- ▶ **seguridad de acceso** es la 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** es la capacidad del producto software para adherirse a normas, convenciones o regulaciones en leyes y prescripciones similares relacionadas con funcionalidad



Fiabilidad

- ▶ un conjunto de atributos que muestran la capacidad del software de mantener su nivel de performance bajo condiciones enunciadas durante un período de tiempo enunciado
- ▶ está organizado en:
 - ▶ madurez
 - ▶ tolerancia a fallas
 - ▶ capacidad de recuperación
 - ▶ cumplimiento de la fiabilidad



Fiabilidad

- ▶ **madurez** es la capacidad del producto software para evitar fallar como resultado de fallos en el software
- ▶ **tolerancia a fallos** es la capacidad del software para mantener un nivel especificado de prestaciones en caso de fallos software o de infringir sus interfaces especificados.



Fiabilidad

- ▶ **capacidad de recuperación** es la capacidad del producto software para reestablecer 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 al fiabilidad



Usabilidad

- ▶ un conjunto de atributos que muestran los esfuerzos necesarios para usar, y la evaluación individual de tal uso, por parte de un conjunto de usuarios definidos o implícitos
- ▶ está compuesto por:
 - ▶ capacidad para ser entendido
 - ▶ capacidad para ser aprendido
 - ▶ capacidad para ser operado
 - ▶ capacidad de atracción
 - ▶ cumplimiento de la usabilidad



Usabilidad

- ▶ **capacidad para ser entendido** es la capacidad del producto software que permite al usuario entender si el software es adecuado y cómo puede ser usado para unas tareas o condiciones de uso particulares
- ▶ **capacidad para ser aprendido** es la capacidad del producto software que permite al usuario aprender sobre su aplicación



Usabilidad

- ▶ **capacidad para ser operado** es la capacidad del producto software que permite al usuario operarlo y controlarlo
- ▶ **capacidad de atracción** es la capacidad del producto software para ser atractivo al usuario
- ▶ **cumplimiento de la usabilidad** es la capacidad del producto software para adherirse a normas, convenciones, guías de estilo o regulaciones relacionadas con la usabilidad



Eficiencia

- ▶ un conjunto de atributos que muestran la relación entre el nivel de performance y la cantidad de recursos utilizados, bajo condiciones especificadas
- ▶ formada por:
 - ▶ comportamiento temporal
 - ▶ utilización de recursos
 - ▶ cumplimiento de la eficiencia



Eficiencia

- ▶ **comportamiento temporal** es la capacidad del producto software para proporcionar tiempos de respuesta, tiempos de proceso y potencia apropiados, bajo condiciones determinadas
- ▶ **utilización de recursos** es la 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** es la capacidad del producto software para adherirse a normas o convenciones relacionadas con la eficiencia



Mantenibilidad

- ▶ un conjunto de atributos que muestran los esfuerzos necesarios para implementar determinadas modificaciones
- ▶ compuesto por:
 - ▶ capacidad para ser analizado
 - ▶ capacidad para ser cambiado
 - ▶ estabilidad
 - ▶ capacidad para ser testeado
 - ▶ cumplimiento de la mantenibilidad



Mantenibilidad

- ▶ **capacidad para ser analizado** es la capacidad del producto software para serle diagnosticadas deficiencias o causas de los fallos en el software, o para identificar las partes que han de ser modificadas
- ▶ **capacidad para ser cambiado** es la capacidad del producto software que permite que una determinada modificación sea implementada



Mantenibilidad

- ▶ **estabilidad** es la capacidad del producto software para evitar efectos inesperados debidos a modificaciones del software
- ▶ **capacidad para ser testeado** es la capacidad del producto software que permite que el software modificado sea validado
- ▶ **cumplimiento de la mantenibilidad** es la capacidad del producto software para adherirse a normas o convenciones relacionadas con la mantenibilidad



Portabilidad

- ▶ un conjunto de atributos que muestran la habilidad del software para ser transferido de un entorno a otro
- ▶ está definida por:
 - ▶ adaptabilidad
 - ▶ instalabilidad
 - ▶ coexistencia
 - ▶ capacidad para reemplazar
 - ▶ cumplimiento de la portabilidad



Portabilidad

- ▶ **adaptabilidad** es la 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** es la capacidad del producto software para ser instalado en un entorno especificado



Portabilidad

- ▶ **coexistencia** es la capacidad del producto software para coexistir con otro software independiente, en un entorno común, compartiendo recursos comunes
- ▶ **capacidad para reemplazar** es la 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** es la capacidad del producto software para adherirse a normas o convenciones relacionadas con la portabilidad



ISO 9126 e ISO 14598

- ▶ la norma ISO 9126 no proporciona métricas ni métodos de medición
- ▶ se revisó la norma ISO 9126, y se concluyó en un nuevo modelo de calidad que distingue tres aproximaciones a la calidad de producto – **ISO 14598**:
 - ▶ **calidad interna**: se mide por las propiedades estáticas del código, utilizando técnicas de inspección
 - ▶ **calidad externa**: se mida por las propiedades dinámicas del código cuando éste se ejecuta
 - ▶ **calidad en uso**: se mide por el grado en que el software está realizado en función de las necesidades del usuario en el entorno de trabajo para el cual fue construido

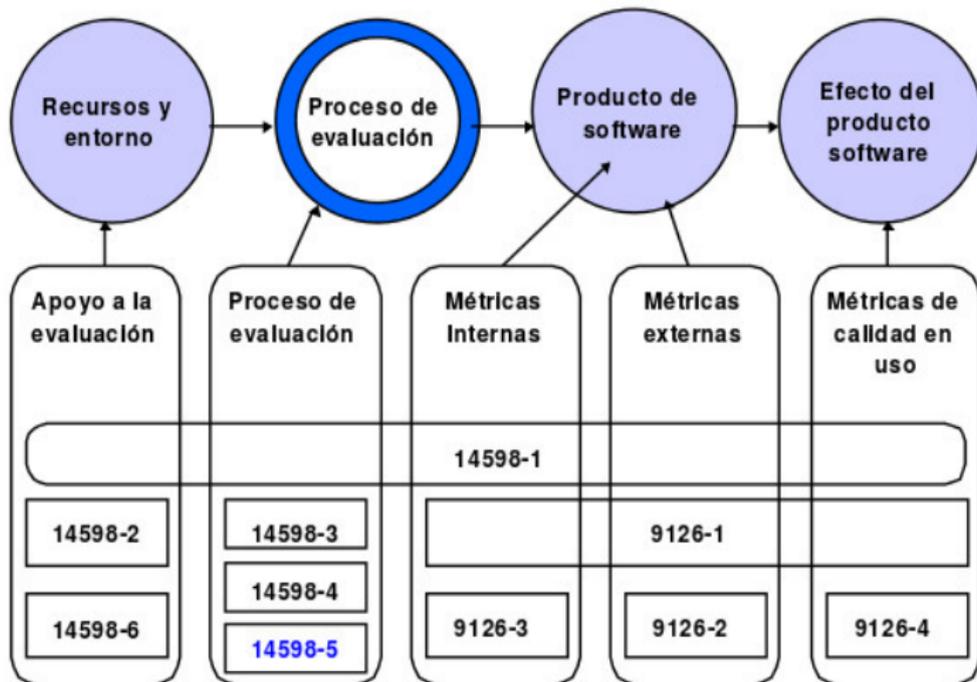


ISO 9126 e ISO 14598

- ▶ **ISO 14598** 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
- ▶ en particular, es utilizada para aplicar los conceptos descritos en la norma ISO / IEC 9126. Se definen y describen las actividades necesarias para analizar los requisitos de evaluación, para especificar, diseñar y realizar acciones de evaluación y para concluir la evaluación de cualquier tipo de producto de software



ISO 9126 e ISO 14598



ISO 25000

- ▶ la serie de estándares **ISO/IEC 25000**, también conocidos como **SQuaRE** (System and Software Quality Requirements and Evaluation), tiene como objetivo crear un framework para la evaluación de calidad en productos de software
- ▶ son el resultado de la evolución de varios otros estándares, en particular from ISO 9126, que define un modelo de calidad, y ISO 14598, que define un proceso para la evaluación de software



Familia de normas ISO 25000



Normas ISO 2500n

- ▶ definen todos los modelos, términos y definiciones comunes referenciados por todas las otras normas de la familia 25000
- ▶ actualmente esta división se encuentra formada por:
 - ▶ **ISO 25000 - Guide to SQuaRE**: contiene el modelo de la arquitectura de SQuaRE, la terminología de la familia, un resumen de las partes, los usuarios previstos y las partes asociadas, así como los modelos de referencia.
 - ▶ **ISO 25001 - Planning and Management**: establece los requisitos y orientaciones para gestionar la evaluación y especificación de los requisitos del producto software



Normas ISO 2501n: Modelo de Calidad

- ▶ presentan modelos de calidad detallados incluyendo características para calidad interna, externa y en uso del producto software
- ▶ actualmente esta división se encuentra formada por:
 - ▶ **ISO 25010 - System and software quality models**: describe el modelo de calidad para el producto software y para la calidad en uso. Esta Norma presenta las características y subcaracterísticas de calidad frente a las cuáles evaluar el producto software
 - ▶ **ISO 25012 - Data Quality model**: define un modelo general para la calidad de los datos, aplicable a aquellos datos que se encuentran almacenados de manera estructurada y forman parte de un Sistema de Información



Normas ISO 2502n: Mediciones de Calidad

- ▶ 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
- ▶ actualmente esta división se encuentra formada por:
 - ▶ **ISO 25020 - Measurement reference model and guide**: presenta un modelo de referencia común para los elementos de medición de la calidad y una guía para seleccionar o desarrollar y aplicar medidas
 - ▶ **ISO 25021 - Quality measure elements**: conjunto recomendado de métricas básicas y derivadas para cualquier etapa del desarrollo
 - ▶ **ISO 25022 - Measurement of quality in use**: define métricas para realizar la medición de la calidad en uso del producto
 - ▶ **ISO 25023 - Measurement of system and software product quality**: define métricas para realizar la medición de la calidad de producto
 - ▶ **ISO 25024 - Measurement of data quality**: define métricas para realizar la medición de la calidad de datos



Normas ISO 2503n: Requisitos de Calidad

- ▶ 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
- ▶ actualmente está compuesta por:
 - ▶ **ISO 25030 - Quality requirements**: provee de un conjunto de recomendaciones para realizar la especificación de los requisitos de calidad del producto software



Normas ISO 2504n: Evaluación de Calidad

- ▶ proporcionan requisitos, recomendaciones y guías para llevar a cabo el proceso de evaluación del producto software
- ▶ se encuentra formada por:
 - ▶ **ISO 25040 - Evaluation reference model and guide**: propone un modelo de referencia para la evaluación, cuáles son las entradas al proceso de evaluación, las restricciones y los recursos necesarios
 - ▶ **ISO 25041 - Evaluation guide for developers, acquirers and independent evaluators**: describe los requisitos y recomendaciones para la implementación práctica de la evaluación del producto desde distintos puntos de vista
 - ▶ **ISO 25042 - Evaluation modules**: define lo que se considera un módulo de evaluación y la documentación, estructura y contenido que se debe utilizar a la hora de definir uno de estos módulos
 - ▶ **ISO 25045 - Evaluation module for recoverability**: define un módulo para la evaluación de la Recuperabilidad



Normas ISO 25050 a 25099

- ▶ la división de extensión de SQuaRE (ISO 25050 a ISO 25099) se reserva para normas o informes técnicos que aborden dominios de aplicación específicos o que puedan ser utilizados para complementar otras normas de la familia SQuaRE
- ▶ hoy en día existen:
 - ▶ ISO 25051 - Requirements for quality of Commercial Off-The-Shelf (COTS) software product and instructions for testing
 - ▶ ISO 25060 - Common Industry Format (CIF) for Usability – General Framework for Usability-related Information (en proceso de aprobación)
 - ▶ ISO 25062 - Common Industry Format (CIF) for usability test reports



Modelo de Madurez CMM

- ▶ es un marco que describe los elementos claves de un proceso de software efectivo
- ▶ describe un camino de mejora evolutivo desde un proceso ad hoc inmaduro a un proceso maduro y disciplinado
- ▶ CMM describe prácticas para:
 - ▶ planificación
 - ▶ ingeniería
 - ▶ administración del desarrollo
 - ▶ mantenimiento de software
- ▶ cuando las organizaciones siguen estas prácticas se mejora la habilidad para cumplir las metas de costos, de planificación, de funcionalidad y calidad del producto

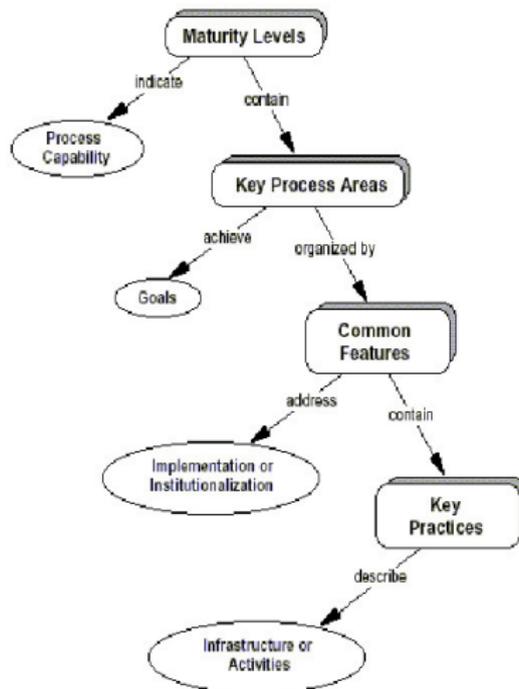


Modelo de Madurez CMM

- ▶ CMM está compuesto por cinco **niveles de madurez**
- ▶ con excepción del nivel 1, cada nivel de madurez está compuesto por **áreas de proceso clave**
- ▶ cada área de proceso clave está organizada en cinco secciones, llamadas **características comunes**
- ▶ las características comunes especifican las prácticas claves, que cuando se implementan colectivamente satisfacen los objetivos de las áreas de proceso claves



Modelo de Madurez CMM



Modelo de Madurez CMM

- ▶ **niveles de madurez (NM)**: es una plataforma evolutiva bien definida con el propósito de lograr un proceso de software maduro
- ▶ las **capacidades del proceso (CP)** describen el rango de resultados esperados que pueden ser logrados siguiendo un proceso de software
- ▶ las CP de una organización proveen una forma de predecir fielmente las salidas esperadas del próximo proyecto de software encarado por la organización



Modelo de Madurez CMM

- ▶ las **áreas de proceso clave** (APC) identifican una serie de actividades relacionadas, que cuando se realizan colectivamente, logran un conjunto de objetivos considerados importantes para establecer las capacidades del proceso (CP) en ese nivel de madurez
- ▶ las APC se diseñaron para que residan en un sólo Nivel de Madurez
- ▶ ejemplo: una de las APC para el nivel 2 es Planificación de Proyectos de Software



Modelo de Madurez CMM

- ▶ los **objetivos** son prácticas claves de un Área de Proceso Clave
- ▶ se usan para determinar si una organización o proyecto ha implementado efectivamente un APC
- ▶ representan el alcance, los límites, y la intención de cada APC
- ▶ ejemplo: un objetivo del APC Planificación de Proyectos de Software es “las estimaciones del software son documentadas para su uso en la planificación y seguimiento del proyecto de software”



Modelo de Madurez CMM

- ▶ las **características comunes** (CC) constituyen prácticas claves en cada APC
- ▶ están divididas en cinco secciones de CC:
 - ▶ requisitos a realizar
 - ▶ habilidad para realizar
 - ▶ actividades realizadas
 - ▶ mediciones y análisis
 - ▶ verificación de la implementación
- ▶ las CC son atributos que indican si la implementación y la institucionalización de un APC es efectiva y repetible
- ▶ las CC de Actividades Realizadas describen las actividades de implementación
- ▶ las otras cuatro CC describen los factores de institucionalización, que hacen que un proceso sea parte de la cultura de la organización

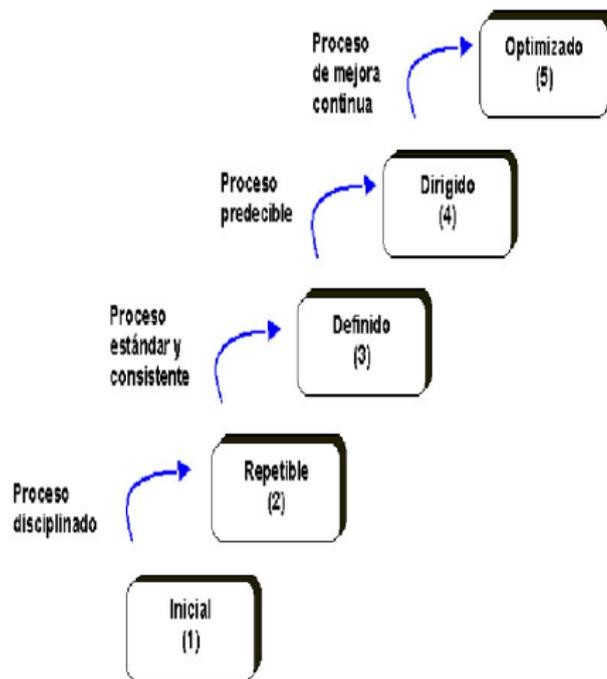


Modelo de Madurez CMM

- ▶ las **prácticas claves** (PC) describen cada APC en términos de PC que, cuando son implementadas, ayudan a satisfacer los objetivos de ese APC
- ▶ describen la infraestructura y las actividades que contribuyen a una implementación e institucionalización efectiva del APC
- ▶ ejemplo: una PC para el APC Planificación de Proyectos de Software es “el plan de desarrollo del proyecto de software es desarrollado de acuerdo a un procedimiento documentado”



Niveles CMM



CMM nivel I

- ▶ la organización no provee un entorno estable para desarrollo y mantenimiento de software
- ▶ cuando una organización no posee prácticas de administración sólidas, los beneficios de las buenas prácticas de Ingeniería de Software se desvalorizan debido a:
 - ▶ un planeamiento ineficaz
 - ▶ requerimientos de sistemas conducidos por reacción
- ▶ durante una crisis, los proyectos habitualmente abandonan los procedimientos planificados y se pasa a codificar y testear



CMM nivel I

- ▶ el éxito depende enteramente de tener un gerente experto y un equipo de software efectivo y experimentado
- ▶ la capacidad del proceso de software de las organizaciones de nivel 1 es impredecible ya que el proceso de software está cambiando permanentemente o se modifica a medida que el trabajo progresa
- ▶ el proceso es ad hoc
- ▶ los cronogramas, presupuestos, la funcionalidad y la calidad del producto son generalmente impredecibles
- ▶ la performance depende de las capacidades de los individuos, y varía con sus perfiles, conocimientos, y motivaciones



CMM nivel II

- ▶ las políticas para administrar un proyecto de software, y los procedimientos para implementar tales políticas están establecidas
- ▶ la planificación y administración de nuevos proyectos está basada en la experiencia con proyectos similares
- ▶ un objetivo para lograr el nivel 2 es institucionalizar procesos efectivos de administración de proyectos de software
- ▶ esto permite repetir prácticas satisfactorias desarrolladas en proyectos anteriores



CMM nivel II

- ▶ un proceso efectivo se puede caracterizar como practicado, documentado, entrenado, medido, y capaz de mejorar
- ▶ los proyectos en organizaciones de nivel 2 tienen instalado controles básicos de administración de software
- ▶ los compromisos de un proyecto están basados en resultados observados de proyectos anteriores y en los requerimientos del proyecto actual
- ▶ los gerentes **controlan** los costos del software, las planificaciones, y la funcionalidad



CMM nivel II

- ▶ los problemas para cumplir los compromisos se tratan a medida que surgen
- ▶ se definen estándares del proyecto de software, y la organización asegura que se cumplen
- ▶ la capacidad del proceso de software de las organizaciones de nivel 2 se resume como disciplinada debido a que:
 - ▶ la planificación y seguimiento del proyecto de software es estable
 - ▶ los éxitos anteriores pueden repetirse
- ▶ el proceso está bajo el control de un sistema de administración de proyectos, que sigue planes realistas basados en la performance de proyectos previos



CMM nivel III

- ▶ el proceso estándar para desarrollo y mantenimiento de software a través de toda la organización está documentado, incluyendo Ingeniería de Software y los procesos de administración, y estos procesos están integrados en un todo coherente
- ▶ este proceso es conocido en CMM como el **proceso estándar de software** de la organización
- ▶ los procesos establecidos en el nivel 3 son usados (y si es necesario modificados) para ayudar a los gerentes de software y al personal técnico a realizar sus tareas de manera más efectiva



CMM nivel III

- ▶ una organización explota las prácticas de Ingeniería de Software cuando estandariza sus procesos de software
- ▶ existe un grupo que es responsable por las actividades del proceso de software de la organización
- ▶ se implementa un programa de entrenamiento en toda la organización para asegurar que el personal y los gerentes tienen los conocimientos y perfiles necesarios para cumplir con los roles asignados



CMM nivel III

- ▶ los proyectos personalizan el proceso de software estándar de la organización para desarrollar sus propios procesos de software, teniendo en cuenta las características propias de cada proyecto
- ▶ el proceso personalizado es conocido en CMM como el proceso de software definido para el proyecto
- ▶ la capacidad del proceso de software de las organizaciones de nivel 3 se resume como estándar y consistente ya que la Ingeniería de Software y las actividades de administración son estables y repetibles



CMM nivel IV

- ▶ en este nivel, la organización fija **objetivos de calidad** cuantitativos para el producto y el proceso de software
- ▶ para todos los proyectos de la organización, se mide la productividad y la calidad de las actividades importantes del proceso de software
- ▶ las mediciones se realizan como parte de un programa de mediciones de la organización
- ▶ se usa una gran base de datos de procesos de software para coleccionar y analizar datos disponibles de los procesos de software definidos para los proyectos
- ▶ en el nivel 4 los procesos de software son instrumentados con mediciones bien definidas y consistentes



CMM nivel IV

- ▶ las mediciones constituyen el fundamento cuantitativo para evaluar los procesos y los productos
- ▶ la capacidad del proceso de software de las organizaciones de nivel 4 se resume como predecible debido a que el proceso es medido y opera dentro de los límites medibles
- ▶ los productos de software son de una alta calidad predecible



CMM nivel V

- ▶ en este nivel, toda la organización se focaliza en la **mejora continua** de los procesos
- ▶ la organización tiene los medios para identificar las debilidades y fortalecer el proceso pro-activamente, con el objetivo de prevenir la ocurrencia de defectos
- ▶ los datos sobre la efectividad del proceso de software son usados para realizar análisis de costo beneficio de nuevas tecnologías, y proponer cambios al proceso de software de la organización



CMM nivel V

- ▶ se identifican las innovaciones que explotan las mejores prácticas de la Ingeniería de Software, y se transfieren a través de toda la organización
- ▶ en el nivel 5, los equipos de proyectos de software analizan los defectos para determinar las causas
- ▶ los procesos son evaluados para prevenir tipos de defectos conocidos, y las lecciones aprendidas se transfieren a otros proyectos



CMM nivel V

- ▶ la capacidad del proceso de software de las organizaciones de nivel 5 se resume como de mejora continua debido a que las organizaciones están continuamente supervisando para mejorar el rango de capacidad de sus procesos
- ▶ de esta forma, mejoran la performance del proceso de sus proyectos
- ▶ las mejoras ocurren por avances incrementales, y por innovaciones usando nuevas tecnologías y métodos



Áreas de Proceso Clave

- ▶ las APC fueron diseñadas para residir en un sólo nivel de madurez
- ▶ las APC son bloques de construcción que indican las áreas que una organización debe atender para mejorar su proceso de software
- ▶ las APC identifican los objetivos que se deben seguir para lograr un nivel de madurez



APC nivel II

- ▶ administración de la configuración del software
- ▶ aseguramiento de la calidad del software
- ▶ administración de subcontratos del software
- ▶ control y seguimiento del proyecto de software
- ▶ planificación del proyecto de software
- ▶ administración de requerimientos



APC nivel III

- ▶ revisiones por pares
- ▶ coordinación entre grupos
- ▶ ingeniería de productos de software
- ▶ administración de software integrada
- ▶ programa de entrenamiento
- ▶ definición del proceso de optimización
- ▶ foco del proceso de organización



APC nivel IV

- ▶ administración de la calidad del software
- ▶ administración de procesos cuantitativos



APC nivel V

- ▶ administración del proceso de cambio
- ▶ administración de cambios de tecnología
- ▶ prevención de defectos



CMMI

- ▶ tras la publicación del modelo **CMM for Software**, se comenzaron a desarrollar modelos para mejorar la madurez de las capacidades en otras áreas y ámbitos:
 - ▶ P-CMM: People CMM
 - ▶ SA-CMM: Software Acquisition CMM
 - ▶ SSE-CMM: Security Systems Engineering CMM
 - ▶ T-CMM: Trusted CMM
 - ▶ SE-CMM: Systems Engineering CMM
 - ▶ IPD-CMM: Integrated Product Development CMM



CMMI

- ▶ a finales de los 90 algunas organizaciones llevaban a cabo planes de calidad que integraban de forma simultánea varios modelos CMM
- ▶ para facilitar la incorporación de varios CMM's, SEI desarrolla y publica en 2001 el modelo **CMMI** que integra:
 - ▶ CMM-SW
 - ▶ SE-CMM
 - ▶ IPD-CMM
- ▶ desde entonces estos tres modelos ya no evolucionan de forma separada
- ▶ CMMI se basa en los mismos conceptos que CMM: **madurez** y **capacidad**



CMMI

- ▶ **madurez**: atributo de las organizaciones que desarrollan o mantienen los sistemas de software
- ▶ en la medida que éstas llevan a cabo su trabajo siguiendo procesos, y en la que éstos se encuentran homogéneamente implantados, definidos con mayor o menor rigor; conocidos y ejecutados por todos los equipos de la empresa; y medidos y mejorados de forma constante, las organizaciones serán más o menos “maduras”
- ▶ **capacidad**: atributo de los procesos. El nivel de capacidad de un proceso indica si sólo se ejecuta, o si también se planifica se encuentra organizativa y formalmente definido, se mide y se mejora de forma sistemática



CMMI niveles de capacidad

- ▶ **0 Incompleto** el proceso no se realiza, o no se consiguen sus objetivos
- ▶ **1 Ejecutado** el proceso se ejecuta y se logra su objetivo
- ▶ **2 Gestionado** además de ejecutarse, el proceso se planifica, se revisa y se evalúa para comprobar que cumple los requisitos
- ▶ **3 Definido** además de ser un proceso “gestionado” se ajusta a la política de procesos que existe en la organización, alineada con las directivas de la empresa
- ▶ **4 Cuantitativamente gestionado** además de ser un proceso definido se controla utilizando técnicas cuantitativas
- ▶ **5 Optimizado** además de ser un proceso cuantitativamente gestionado, de forma sistemática se revisa y modifica para adaptarlo a los objetivos del negocio



CMMI niveles de madurez

- ▶ son los mismos 5 que los descritos en el modelo SW-CMM, si bien se les han revisado los nombres a los niveles 2 y 4
- ▶ nivel 1: Inicial
- ▶ nivel 2: Gestionado
- ▶ nivel 3: Definido
- ▶ nivel 4: Gestionado cuantitativamente
- ▶ nivel 5: Optimizado



CMMI

CMMI Main Page - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://www.sei.cmu.edu/cmmi/cmmi.html

Carnegie Mellon Software Engineering Institute Home Search Contact Us Site Map What's New

About the SEI Management Engineering Acquisition Work with Us Products and Services Publications

Organizations Improving management practices Individuals

MANAGEMENT

- CMMI Main Page
- [What Is CMMI?](#)
- [Models](#)
- [Adoption](#)
- [Training, Events, & Forums](#)
- [Performance Results](#)
- [Appraisals](#)
- [Frequently Asked Questions \(FAQs\)](#)
- [Background](#)
- [Contact Information](#)

Welcome to the CMMI® Web Site

From this page, you can view information about how to [navigate](#) and [search](#) this site, [submit](#) change requests, and [work](#) with the SEI.

If you are new to CMMI, view the [What is CMMI?](#) page for introductory information.

CMMI News

The [CMMI Survival Guide: Just Enough Process Improvement](#) by Suzanne Garcia and Richard Turner is now available.

Become an SEI-Certified [SCAMPI High Maturity Lead Appraiser](#).

Changes to [policies on SCAMPI A appraisals](#) and the [sunset period for v1.1](#) have been announced.

The [CMMI version 1.2](#) Product Suite is now available.

A report about the impact and benefits of using CMMI, titled [Performance](#)