

# **TEMA 5: CALIDAD EN SISTEMAS MULTIMEDIA**

# Índice

- Introducción
- Definición de calidad
- Situación actual
- Niveles de calidad
- Modelos de calidad del Software
  - Calidad del producto: ISO 9126
  - Calidad de proceso: ISO 90003
  - Calidad de proceso: CMMI
  - Calidad de proceso: ISO 15504 SPICE
- Coste de la calidad
- Bibliografía

# Introducción

- El éxito de un proyecto se suele medir sobre 3 ejes:
  - Agenda: ¿El proyecto fue entregado a tiempo?
  - Coste: ¿El proyecto se mantuvo dentro del presupuesto?
  - Calidad: ¿El producto tuvo una buena calidad?
- Un proyecto exitoso es aquel que se completa a tiempo, permanece dentro del presupuesto y tiene desarrolladas las funciones requeridas

# Introducción

- ¿Qué quieren los clientes?
- Algunos responsables del proyecto piensan que los clientes están más interesados en una entrega del proyecto a tiempo o de un desarrollo a bajo coste
  - Seguro??
- Los clientes pueden perdonar que un proyecto esté ligeramente fuera de fechas
- Pueden incluso perdonar un proyecto con un coste mayor de lo planificado
- Pero ningún cliente perdonaría un proyecto de baja calidad

# Introducción

- DRAE: Calidad
  - Propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor.
  - Buena calidad, superioridad o excelencia
- ¿Qué tiene más calidad?



# Introducción

- Definición coloquial:
  - En la vida cotidiana “la calidad representa las propiedades inherentes a un objeto que permiten apreciarlo como mejor, igual o peor que otros objetos de su especie”
- Definición profesional:
  - Totalidad de las características y aspectos de un producto o servicio en los que se basa su aptitud para satisfacer una necesidad dada

# Introducción

- La calidad del software es un **concepto complejo**
  - El producto desarrollado cumple su especificación: criterio insuficiente
    - La especificación se centra en las características deseadas por el usuario, y se suelen olvidar otras importantes (por ejemplo, mantenimiento)
    - Difícil especificar detalladamente y de forma medible ciertas características de calidad (facilidad de uso, mantenimiento,...)
    - Cuando la especificación del software es incompleta, el usuario percibe falta de calidad
  - Diferentes atributos de la calidad (mantenibilidad, eficiencia, portabilidad, rendimiento, fiabilidad,...)

# Definición de calidad

*“Conjunto de propiedades y características de un producto o servicio que le confieren su aptitud para satisfacer unas necesidades explícitas o implícitas.” ISO 8402*

*“Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente” R.S.Pressman (1992)*



# Definición de calidad

*“Calidad es la idoneidad de uso. Es decir, las características del producto que satisfacen las necesidades del cliente y, por tanto, producen satisfacción de producto. La calidad es la inexistencia de deficiencias” Juran*

*“La calidad se define, desde el punto de vista del cliente, como cualquier cosa que aumenta su satisfacción” Deming*

# Definición de calidad

*“Nivel al que una serie de características inherentes satisfacen los requisitos” ISO 9000: 2000*

*“La capacidad de un conjunto de características inherentes de un producto, o componente del producto, o proceso, de satisfacer por completo los requisitos del cliente” CMMI® (S.E.I.)*

# Concepto de calidad

- Calidad es un concepto:
  - **Relativo**. La calidad está en los ojos del observador y es relativa a las personas, su edad y circunstancias, al espacio, tiempo, ...
  - **Multidimensional**. Referida a varias cualidades: Funcionalidad, Oportunidad, Coste, ...
  - Sujeta a **restricciones**. Presupuesto disponible
  - Ligado a **compromisos aceptables**. Plazos de fabricación

# Concepto de calidad

- El objetivo no es necesariamente alcanzar una **calidad perfecta**, sino la necesaria y suficiente para cada **contexto de uso** a la hora de la entrega y del uso por parte de los usuarios
- Es necesario comprender las necesidades reales de los usuarios con tanto detalle como sea posible (**requisitos**)

# Perspectivas de la calidad

- Es importante entender lo que compromete la buena calidad

La calidad se puede ver desde diferentes perspectivas:

- **Perspectiva de calidad del cliente**
- **Perspectiva de calidad del desarrollador**

# Perspectivas de la calidad

## Perspectiva de calidad del cliente

- Un producto está construido para cumplir los requisitos del cliente.
- Estos **requisitos** son de **dos tipos**:
  - **Explícitos**: Lo que el cliente plantea explícitamente
  - **Implícitos**: Lo que el cliente no especifica pero espera

# Perspectivas de la calidad

## Perspectiva de calidad del desarrollador

- Para los desarrolladores, un producto es de calidad cuando todas las especificaciones dadas por el cliente se han cumplido
- Sin embargo, los clientes pueden fallar al especificar algunos requisitos o no especificarlos claramente
- A menudo, el fracaso al construir un producto de calidad es el resultado de una mala especificación de requisitos o de la especificación de requisitos ambiguos

# Perspectivas de la calidad

## Perspectiva de calidad del desarrollador

- En el contexto del software, el entendimiento entre los desarrolladores es distinto cuando los requisitos implícitos no son recogidos durante la fase de análisis
- Por lo tanto, estos requisitos no son incluidos en las especificaciones que los desarrolladores intentan cumplir
- Además, el desarrollador puede interpretar especificaciones ambiguas de forma diferente a lo que el cliente intentó



# Factores de calidad del Software

- Se pueden clasificar en dos grandes grupos (Pressman):
  - Factores que pueden ser medidos directamente
  - Factores que sólo pueden ser medidos indirectamente
- Se centran en tres aspectos importantes de un producto software (McCall):
  - Características operativas
  - Capacidad de soportar los cambios
  - Adaptabilidad a nuevos entornos

# Factores de calidad del Software

- **Características operativas**

- Corrección. ¿Hace lo que quiero?
- Fiabilidad. ¿Lo hace de forma fiable todo el tiempo?
- Eficiencia. ¿Se ejecutará en mi hardware lo mejor que pueda?
- Seguridad (Integridad). ¿Es seguro?
- Facilidad de uso. ¿Está diseñado para ser usado?

# Factores de calidad del Software

- **Capacidad de soportar los cambios**
  - Facilidad de mantenimiento. ¿Puedo corregirlo?
  - Flexibilidad. ¿Puedo cambiarlo?
  - Facilidad de prueba. ¿Puedo probarlo?
- **Adaptabilidad a nuevos entornos**
  - Portabilidad. ¿Podré usarlo en otra máquina?
  - Reusabilidad. ¿Podré reutilizar alguna parte del software?
  - Interoperabilidad. ¿Podré hacerlo interactuar con otro sistema?

# Situación actual

- La industria del software no ha acabado de salir de la fase artesanal
- Padecemos de “prisa patológica”, que consecuencia directa de:
  - Desorganización
  - Falta de planificación
- Dedicamos nuestros esfuerzos de hoy a arreglar lo que se hizo mal ayer

# Situación actual

- En una organización inmadura:
  - Procesos software normalmente improvisados
  - Si se han especificado, no se siguen rigurosamente
  - Organización reactiva (resolver crisis inmediatas)
  - Planes y presupuestos excedidos sistemáticamente, al no estar basados en estimaciones realistas
  - Si hay plazos rígidos, se sacrifican funcionalidad y calidad del producto para satisfacer el plan

# Situación actual

- No existen bases objetivas para juzgar la calidad del producto
- Cuando los proyectos están fuera de plan, las revisiones o pruebas se recortan o eliminan
- El 90% de los proyectos no alcanzan los objetivos
- El 40% fracasan por completo
- El 29% no se entregan nunca

# Situación actual

- ¿Qué se puede hacer?
- Cambio cultural de todos los involucrados!!

**ARTESANÍA → INGENIERÍA**

# Niveles de calidad

La calidad del software puede gestionarse a distintos niveles:

- Nivel de producto
- Nivel de proyecto
- Nivel de proceso



# Niveles de calidad

- **A nivel de producto:**
  - Cuando nos centramos en el proceso de desarrollo de software y hacemos una serie de pruebas en paralelo con cada etapa, para detectar y corregir los posibles defectos que puedan surgir
- **A nivel de proyecto:**
  - Cuando nos centramos en controlar todas las fases y áreas de gestión de proyecto, implantando metodologías y mejores prácticas que aseguren la correcta gestión de las mismas

# Niveles de calidad

- **A nivel de proceso:**

- Cuando nos centramos en gestionar todas las áreas de proceso de una organización, mediante la implantación de una metodología.
- De esta forma se consigue tener mayor información de los procesos de modo que puedan controlarse y mejorarse, y produzcan así, un aumento de la calidad de los productos y servicios relacionados con ellos

# GESTIÓN DE LA CALIDAD

## GESTIÓN de PROCESO

CMMI

SPICE

## GESTIÓN de PROYECTO

Gestión de la  
integración

Gestión del  
alcance

Gestión del  
tiempo

Gestión de  
RRHH

Gestión de  
la calidad

Gestión de la  
comunicación

Gestión de  
costes

Gestión del  
riesgo

Gestión de la  
logística

## GESTIÓN de PRODUCTO

### GESTIÓN de REQUISITOS

Requisitos  
de  
Negocio

Requisitos  
de  
Pruebas

Estrategia  
De  
Pruebas

Definir  
Casos de  
Uso

### PRUEBAS FUNCIONALES

Crear/automatizar  
casos de prueba

Defectos

Crear plan  
de pruebas

Ejecutar  
escenarios de  
prueba

Analizar resultados  
Informar defectos

### PRUEBAS RENDIMIENTO

Evaluar

Diagnosticar  
problemas

Incidencia

Planificar  
capacidad

Ajuste

### PRUEBAS SEGURIDAD

Probar  
seguridad

Diagnosticar  
problemas

Incidencia

Planificar  
seguridad

Ajuste

## METRICAS de CALIDAD

## PUBLICACIÓN y DIFUSIÓN - KPIs

## DISEÑO y DESARROLLO

# Modelos de calidad del Software

- **Objetivo:** La mejora de procesos software
- Diversos modelos que **buscan:**
  - Determinar las fuerzas y debilidades en una organización
  - Aglutinar esfuerzos para conseguir acuerdos sobre lo que es un buen proceso.

# Modelos de calidad del Software

- **Principales iniciativas:**

- **ISO 9001 y 9000-3:**

- Muy útil en compañías que además de software fabrican equipos
    - Define los procesos de calidad tanto en compañías de hardware como de software.
    - Muy utilizado en Europa.

- **Capability Maturity Model (CMM) del SEI (Carnegie Mellon University )**

- El modelo más empleado y maduro
    - Valora el desarrollo de software en sistemas de gran complejidad
    - Visión completa del proceso de madurez organizacional
    - Incluye mecanismos para la mejora continua de los procesos

# Modelos de calidad del Software

- **Principales iniciativas:**

- **Bootstrap:**

- Enfocado a pequeñas y medianas empresas
    - Valora la madurez global de una organización
    - Examina procesos individuales de software y valora la conveniencia y el impacto de nuevas tecnologías

- **SPICE:**

- Combina elementos de ISO, CMM y Bootstrap
    - Enfocado a estudiar el nivel de madurez de los procesos individuales (tiene en cuenta el contexto de los procesos evaluados).
    - Objetivo: definir un marco común de referencia en el que convivan el resto de los modelos mencionados.
    - Produce un perfil del proceso, en vez de un resultado válido/no válido.

# Calidad del producto: ISO 9126

- La **ISO 9126** define un modelo de calidad para **calidad** tanto **interna** como **externa** y un modelo de **calidad en uso**
  - ISO/IEC 9126-1: Modelo de Calidad
  - ISO/IEC 9126-2: Métricas Externas
  - ISO/IEC 9126-3: Métricas Internas
  - ISO/IEC 9126-4: Métricas de Calidad en Uso

# Calidad del producto: ISO 9126

- Utilidades:
  - Validar la **completitud** de una definición de **requisitos**
  - **Identificar requisitos** software
  - Identificar **objetivos** para el **diseño** software
  - Identificar requisitos para las **pruebas del software**
  - Identificar requisitos para el **aseguramiento de la calidad**
  - Identificar **criterios de aceptación** para un producto software terminado



# Calidad del producto: ISO 9126

- La **calidad interna** es la totalidad de características del producto software desde una visión interna (componentes/productos intermedios)
- La calidad interna es medida y evaluada contra los requisitos de calidad internos
- Los detalles de calidad del producto software pueden ser mejorados durante la implementación, revisión y pruebas del código, pero la naturaleza fundamental de la calidad del producto software representado por calidad interna permanece sin cambios a menos que se lleven a cabo actividades de re-diseño

# Calidad del producto: ISO 9126

- La **calidad externa** es la totalidad de características del producto desde una visión externa (producto final)
- Es la calidad cuando el software es ejecutado, que es típicamente medido y evaluado durante las pruebas en un entorno simulado con datos simulados utilizando métricas externas
- Durante las pruebas, la mayoría de los fallos deberían ser descubiertos y eliminados
- Sin embargo, algunos fallos pueden permanecer después de las pruebas

# Calidad del producto: ISO 9126

- Características de calidad ISO 9126

Funcionalidad	Fiabilidad	Usabilidad	Eficiencia	Mantenibilidad	Portabilidad
Adecuación	Madurez	Capacidad de entender	Tiempo de respuesta	Capacidad analizar	Adaptabilidad
Exactitud	Tolerancia a fallos	Capacidad de aprender	Utilización de recursos	Capacidad de cambio	Instalabilidad
Seguridad de acceso	Capacidad de recuperación	Capacidad para operar	Cumplimiento de eficiencia	Estabilidad	Coexistencia
Cumplimiento funcional	Cumplimiento de fiabilidad	Capacidad de atracción		Capacidad para pruebas	Capacidad de reemplazo
		Cumplimiento usabilidad		Cumplimiento mantenibilidad	Cumplimiento portabilidad

# Calidad del producto: ISO 9126

- Funcionalidad
  - Capacidad del producto software de proporcionar funciones que cumplan las necesidades cuando el software es utilizado bajo condiciones específicas
- Fiabilidad
  - Capacidad del producto software para mantenerse a un nivel específico de rendimiento cuando es utilizado bajo condiciones específicas

# Calidad del producto: ISO 9126

- Usabilidad
  - Capacidad del producto de ser entendido, utilizado y ser atractivo al usuario cuando es utilizado bajo condiciones específicas
- Eficiencia
  - Capacidad del producto para proporcionar un apropiado rendimiento, relativo al conjunto de recursos utilizados bajo condiciones establecidas

# Calidad del producto: ISO 9126

- Mantenibilidad
  - Capacidad del producto de ser modificado
  - Las modificaciones pueden incluir correcciones, mejoras o adaptación del software a cambios en el entorno, en requisitos o en especificaciones funcionales
- Portabilidad
  - Capacidad del producto software de ser transferido desde un entorno a otro

# Calidad del producto: ISO 9126

## FUNCIONALIDAD

- **Adecuación:** Proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados
- **Exactitud:** Proporcionar los resultados o efectos correctos acordados, con el grado necesario de precisión
- **Interoperabilidad:** Interactuar con uno o más sistemas especificados interactuar especificados
- **Seguridad de acceso:** 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 :** Adherirse a normas, convenciones o regulaciones en leyes y prescripciones similares relacionadas con funcionalidad

# Calidad del producto: ISO 9126

## FIABILIDAD

- **Madurez:** Evitar fallar como resultado de fallos en el software
- **Tolerancia a fallos:** Mantener un nivel especificado de prestaciones en caso de fallos software o de infringir sus interfaces especificados
- **Capacidad de recuperación:** Reestablecer un nivel de prestaciones especificado y de recuperar los datos directamente afectados en caso de fallo
- **Cumplimiento de la fiabilidad:** Adherirse a normas, convenciones relacionadas con la fiabilidad



# Calidad del producto: ISO 9126

## USABILIDAD

- **Entendibilidad:** Permitir al usuario entender si el software es adecuado y cómo puede ser usado para unas tareas o condiciones de uso particulares
- **Aprendibilidad:** Permitir al usuario aprender sobre su aplicación
- **Operatividad:** Permitir al usuario operarlo y controlarlo
- **Atracción:** Ser atractivo al usuario
- **Cumplimiento de la usabilidad:** Adherirse a normas, convenciones, guías de estilo o regulaciones relacionadas con la usabilidad

# Calidad del producto: ISO 9126

## EFICIENCIA

- **Comportamiento temporal:** Proporcionar tiempos de respuesta, tiempos de proceso y potencia apropiados
- **Utilización de recursos:** Usar las cantidades y tipos de recursos adecuados cuando el software lleva a cabo su función
- **Cumplimiento de la eficiencia:** Adherirse a normas o convenciones relacionadas con la eficiencia

# Calidad del producto: ISO 9126

## MANTENIBILIDAD

- **Analizabilidad:** Ser diagnosticadas deficiencias o causas de los fallos en el software, o para identificar las partes que han de ser modificadas
- **Cambiabilidad:** Permitir que una determinada modificación sea implementada
- **Estabilidad:** Evitar efectos inesperados debidos a modificaciones del software
- **Facilidad de prueba:** Permitir que el software modificado sea validado
- **Cumplimiento de la mantenibilidad:** Adherirse a normas o convenciones relacionadas con la mantenibilidad

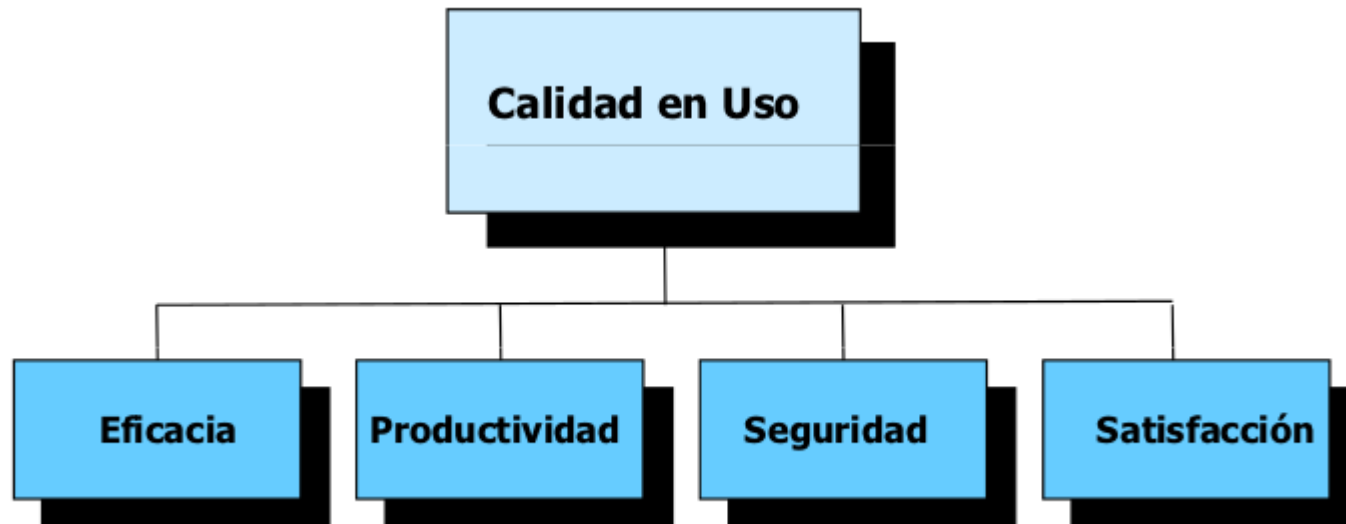
# Calidad del producto: ISO 9126

## PORTABILIDAD

- **Adaptabilidad:** Ser adaptado a diferentes entornos especificados, sin aplicar acciones o mecanismos distintos de aquellos proporcionados para este propósito por el propio software considerado
- **Instalabilidad:** Ser instalado en un entorno especificado
- **Coexistencia:** Coexistir con otro software independiente, en un entorno común, compartiendo recursos comunes
- **Reemplazabilidad:** ser usado en lugar de otro producto software, para el mismo propósito, en el mismo entorno
- **Cumplimiento de la portabilidad:** Adherirse a normas o convenciones relacionadas con la portabilidad

# Calidad del producto: ISO 9126

- Modelo de calidad en uso



# Calidad del producto: ISO 9126

## Modelo de calidad en uso:

- **Eficacia:** Para permitir a los usuarios alcanzar objetivos específicos con precisión y completamente
- **Productividad:** Para permitir a los usuarios emplear recursos apropiados con relación a la eficacia alcanzada
- **Seguridad:** Para alcanzar niveles aceptables de riesgo hacia la gente, negocio, software, propiedad o medio ambiente
- **Satisfacción:** Para satisfacer al usuario

# Calidad del producto: ISO 9126

- **Calidad de Datos**

- Características que deben tener los datos como materias primas para que, utilizando un proceso de producción adecuado, se pueda generar un producto de información

- **Calidad de Información**

- Aquellas características que debería tener un Producto de Información para que su utilización sea adecuada, es decir, cumpla con los requisitos de usuario

- **Dimensiones de Calidad de Datos**

- Son criterios que permiten juzgar la calidad de los datos desde un determinado punto de vista. Se definen en la norma ISO 25012 (similar a la 9126 para el Software)

# Calidad del producto: ISO 9126

- La Calidad de los Datos depende de:
  - Los propios datos (extensión):
    - Influyen en la efectividad de los procesos de negocio (dependencia de la semántica de los negocios)
  - El esquema de los datos (intensión):
    - Ejemplo: Tablas no normalizadas convenientemente
    - Influye en el ciclo de vida de los datos
    - Puede no dar el soporte para los aspectos de calidad requerida por el usuario
  - Procesos técnicos sobre los datos (SGBD):
    - Pueden no implementar mecanismos que aseguren que no se producen errores en los datos, o que los datos satisfagan los requisitos de los usuarios



# Calidad de proceso: ISO 90003

- La norma ISO 90003 proporciona, a las organizaciones, una guía para la adaptación de la ISO 9001:2008 para la adquisición, suministro, desarrollo, instalación y mantenimiento de software y servicios de soporte
- 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

# Calidad de proceso: ISO 90003

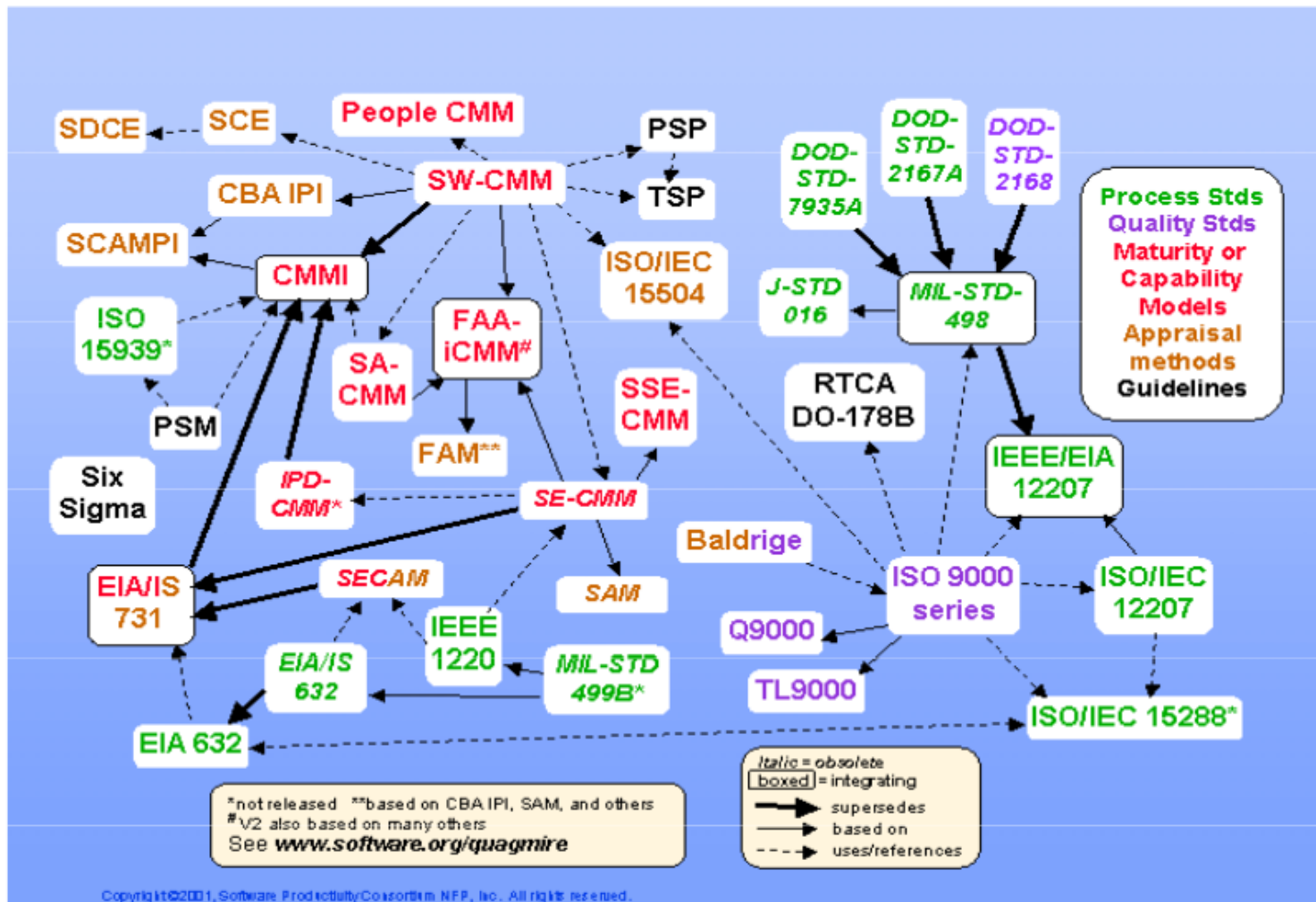
- La organización debe establecer, documentar, implementar y mantener un sistema de gestión de la calidad software y mejorar continuamente su eficacia, de acuerdo con los siguientes requisitos generales:
  - **Identificar** los **procesos** necesarios para el sistema de gestión de la calidad y su aplicación a través de la organización. (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 software, la planificación de la calidad y el desarrollo.

# Calidad de proceso: ISO 90003

- **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 **análisis** de estos procesos
- **Implementar las acciones necesarias** para alcanzar los resultados planificados y la mejora continua de estos procesos

# Calidad de proceso: ISO 90003

- Existen multitud de normas sobre procesos de ingeniería del software, su calidad y su mejora



# Calidad de proceso: CMMI

- CMMI (Capability Maturity Model Integrated) 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:
  - Desarrollado por el SEI (Software Engineering Institute) de la Universidad de Carnegie Mellon, USA. Es la evolución del anterior CMM

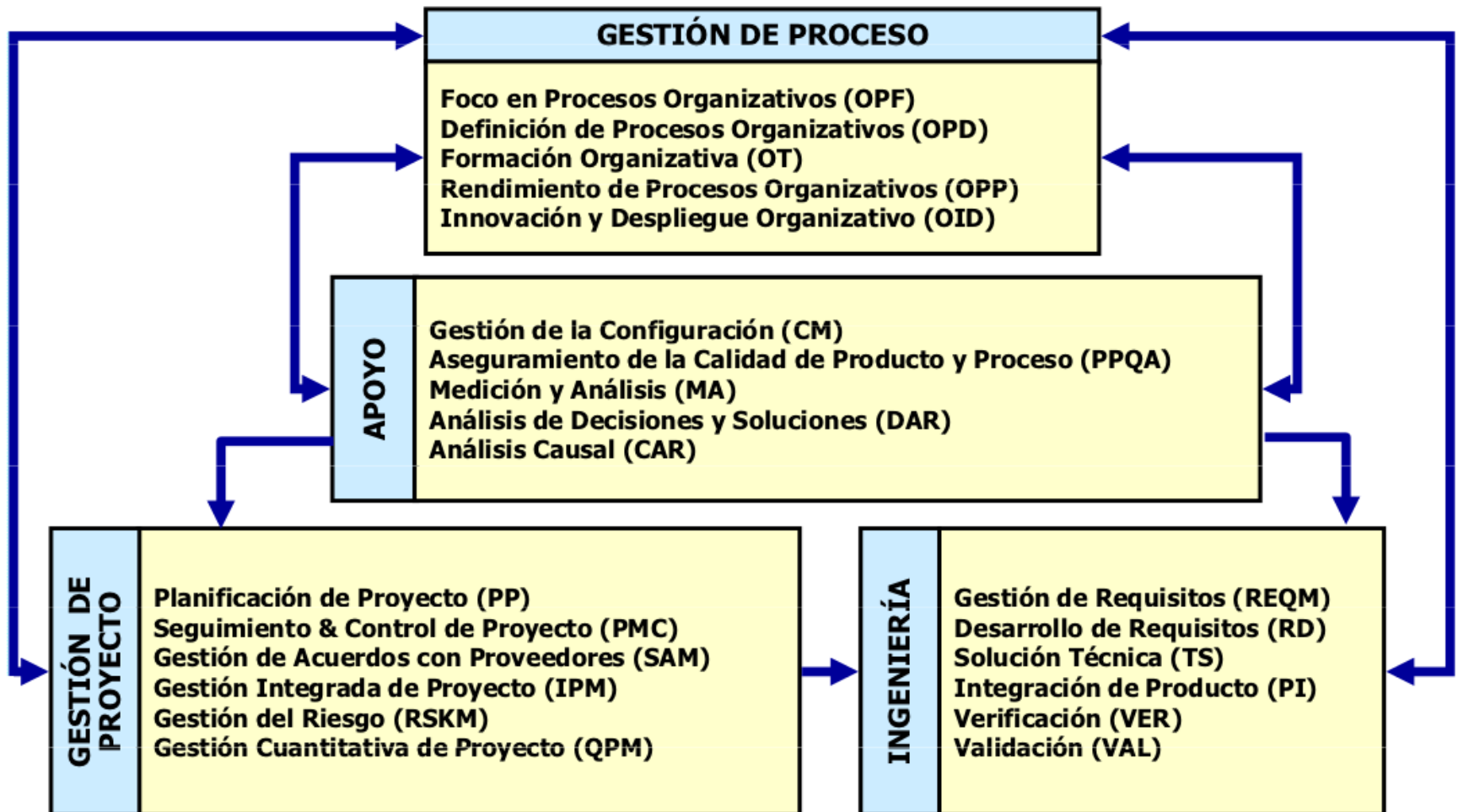
# Calidad de proceso: CMMI

- El CMMI sirve para dos cosas principales:
  - **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
- Existen, en la v1.2, tres modelos

Modelo	Descripción	Extensión
<b>CMMI-DEV SW</b> <b>Ago-06</b>	CMMI for Development	CMMI-DEV IPPD
<b>CMMI-ACQ</b> <b>Nov-07</b>	CMMI for Acquisition	
<b>CMMI-SEV</b> <b>Feb-09</b>	CMMI for Services	

# Calidad de proceso: CMMI

- Áreas de Proceso Claves (KPAs)



# Calidad de proceso: CMMI

- El modelo de referencia CMMI establece una serie de áreas clave (KPA's) agrupadas en los distintos niveles de madurez
- Se establecen 5 niveles de madurez para clasificar las organizaciones en función de qué áreas de procesos consiguen sus objetivos y se gestionan con principios de ingeniería



# Calidad de proceso: CMMI

## Nivel 1: Ejecutado

- Procesos caóticos (ad-hoc)

## Nivel 2: Gestionado

- Gestión de requisitos (REQM)
- Planificación del proyecto (PP)
- Monitorización y Control del Proyecto (PMC)
- Gestión del Acuerdo con el Suministrador (SAM)
- Medición y Análisis (M & A)
- Aseguramiento de la Calidad del Proceso y Producto (PPQA)
- Gestión de la Configuración (CM)

# Calidad de proceso: CMMI

## Nivel 3: Definido

- Desarrollo de Requisitos (RD)
- Solución Técnica (TS)
- Integración del Producto (PI)
- Verificación (VER)
- Validación (VAL)
- Enfoque Proceso Organizacional (OPF)
- Definición del Proceso Organizacional (OPD)
- Formación de la Organización (OT)
- Gestión Integrada de Proyectos (IPM)
- Gestión de Riesgos (RSKM)
- Análisis de Decisión y Resolución (DAR)

# Calidad de proceso: CMMI

## Nivel 4: Gestionado cuantitativamente

- Rendimiento del Proceso Organizacional (OPP)
- Gestión Cuantitativa de Proyectos (QPM)

## Nivel 5: Optimizado

- Innovación y Distribución Organizacional (OID)
- Análisis Causal y Resolución (CAR)

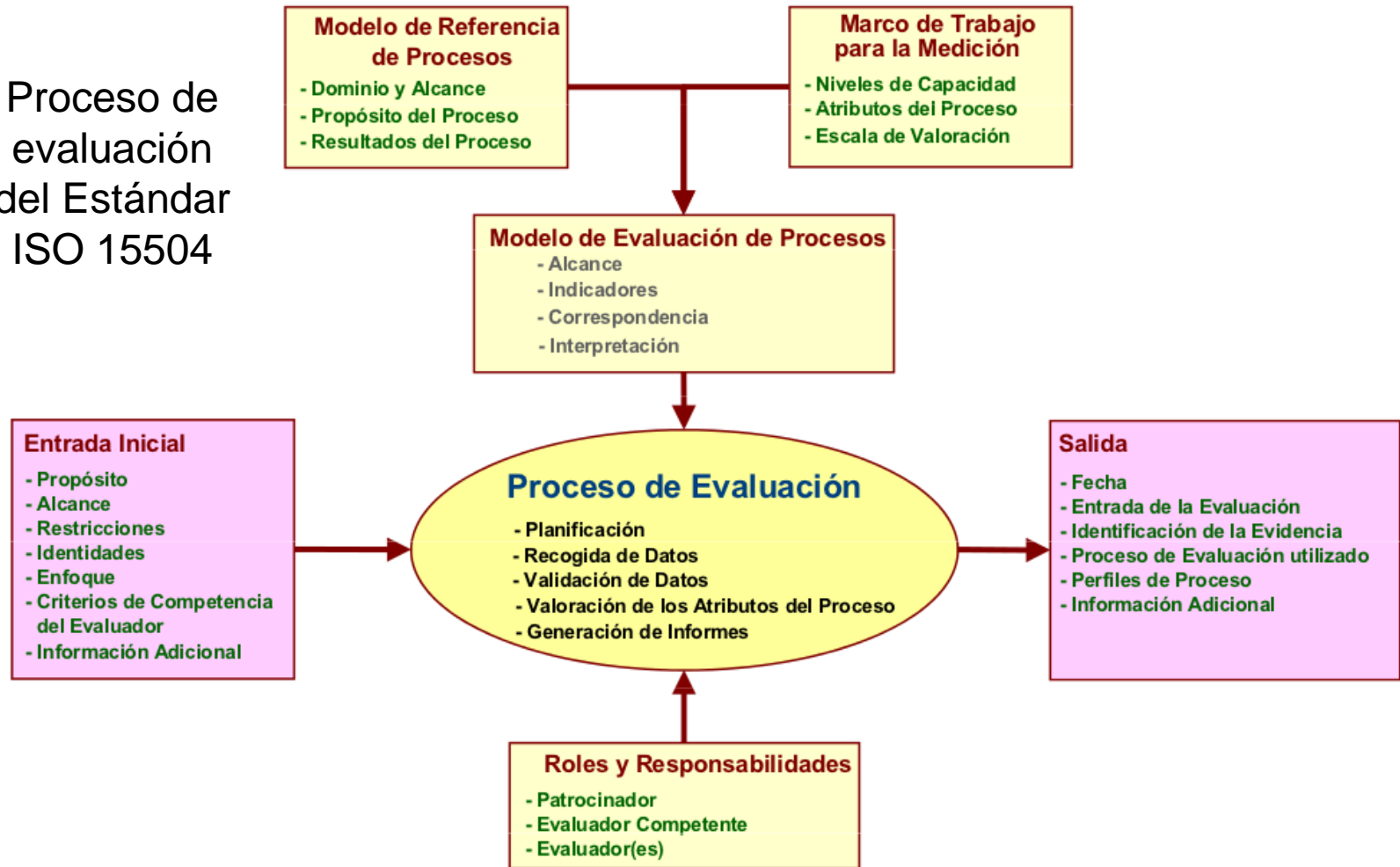
# Calidad de proceso:

## ISO 15504 SPICE

- El estándar ISO/IEC 15504 proporciona:
  - Un marco de trabajo para la evaluación de procesos software
  - Establece los requisitos mínimos para realizar una evaluación que asegure la repetibilidad y consistencia de las valoraciones obtenidas
  - El objetivo de la evaluación del proceso es conocer la capacidad de los procesos de una organización

# Calidad de proceso: ISO 15504 SPICE

Proceso de  
evaluación  
del Estándar  
ISO 15504



# Calidad de proceso:

## ISO 15504 SPICE

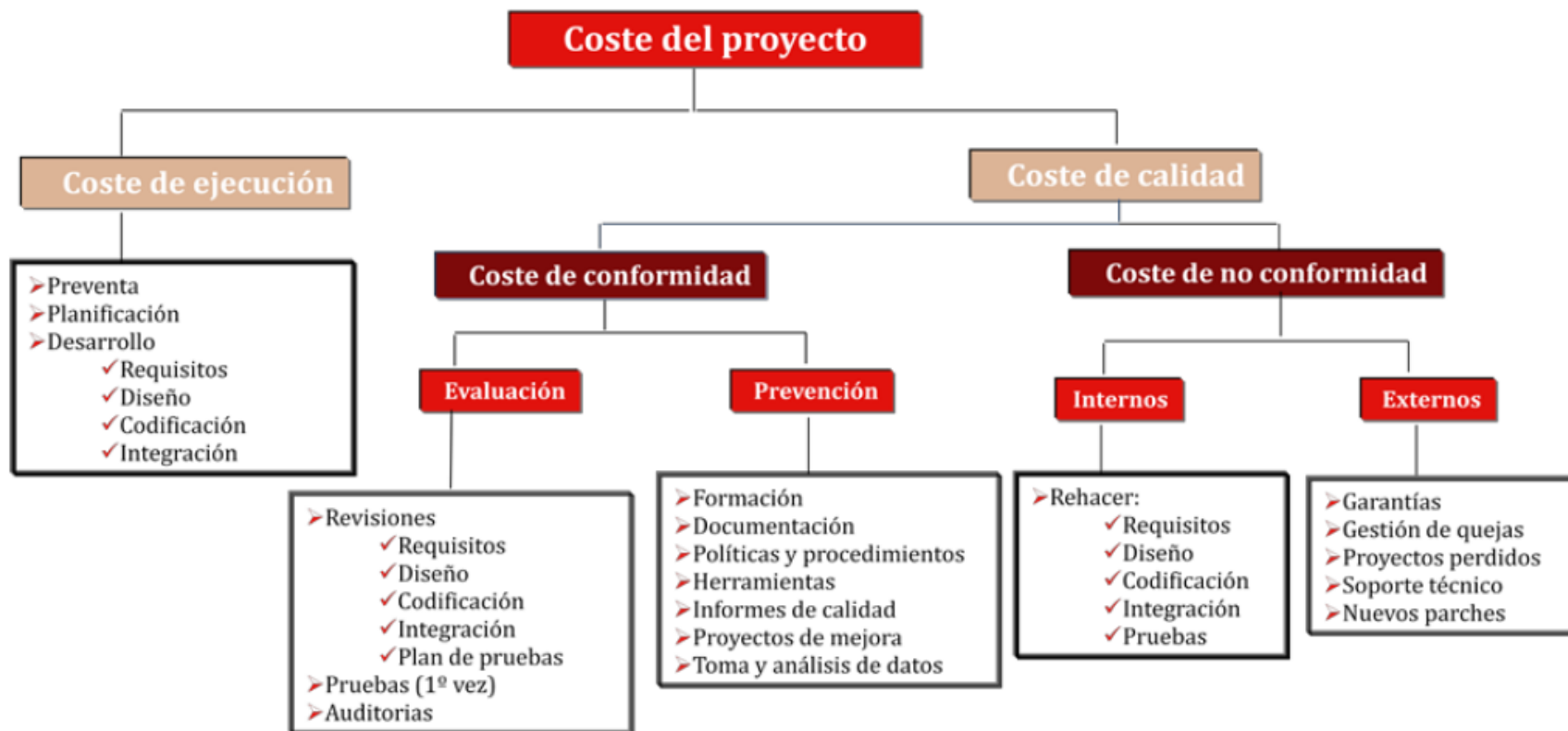
Frente a CMMI:

- Ventajas:
  - Es estándar internacional oficial (alineado con los demás estándares ISO)
  - Es más completo y versátil
- Desventajas:
  - Está menos implantado a nivel industrial (lleva menos años)

# Coste de la calidad

- Todo proyecto tiene su coste
- **Coste de ejecución:**
  - Son los costes que supone la planificación y el propio desarrollo del proyecto
- **Coste de calidad:**
  - Son los costes derivados de temas relacionados con la calidad
  - Estos a su vez se dividen en dos grandes grupos:
    - **Costes de conformidad:** Costes de prevención y evaluación
    - **Costes de no conformidad:** Costes de fallos internos y externos

# Coste de la calidad



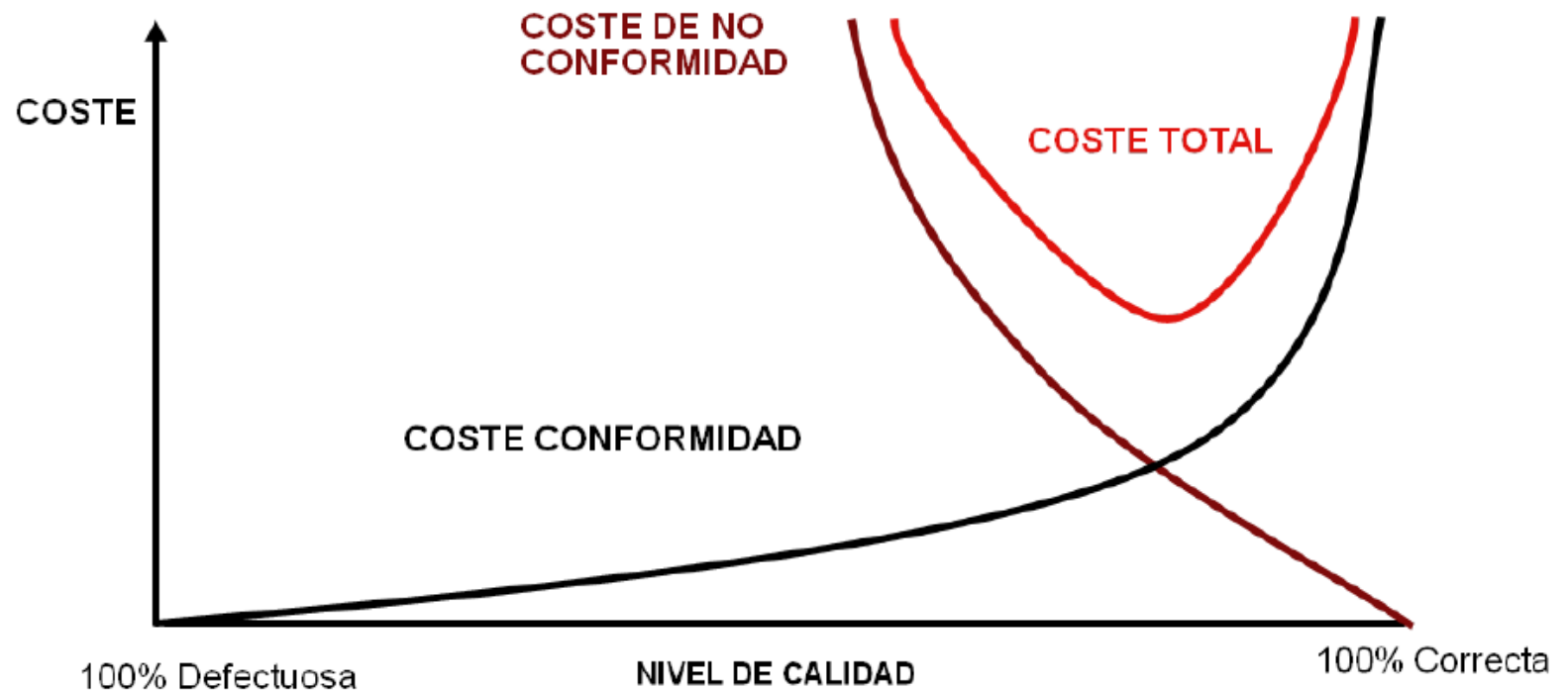


# Coste de la calidad

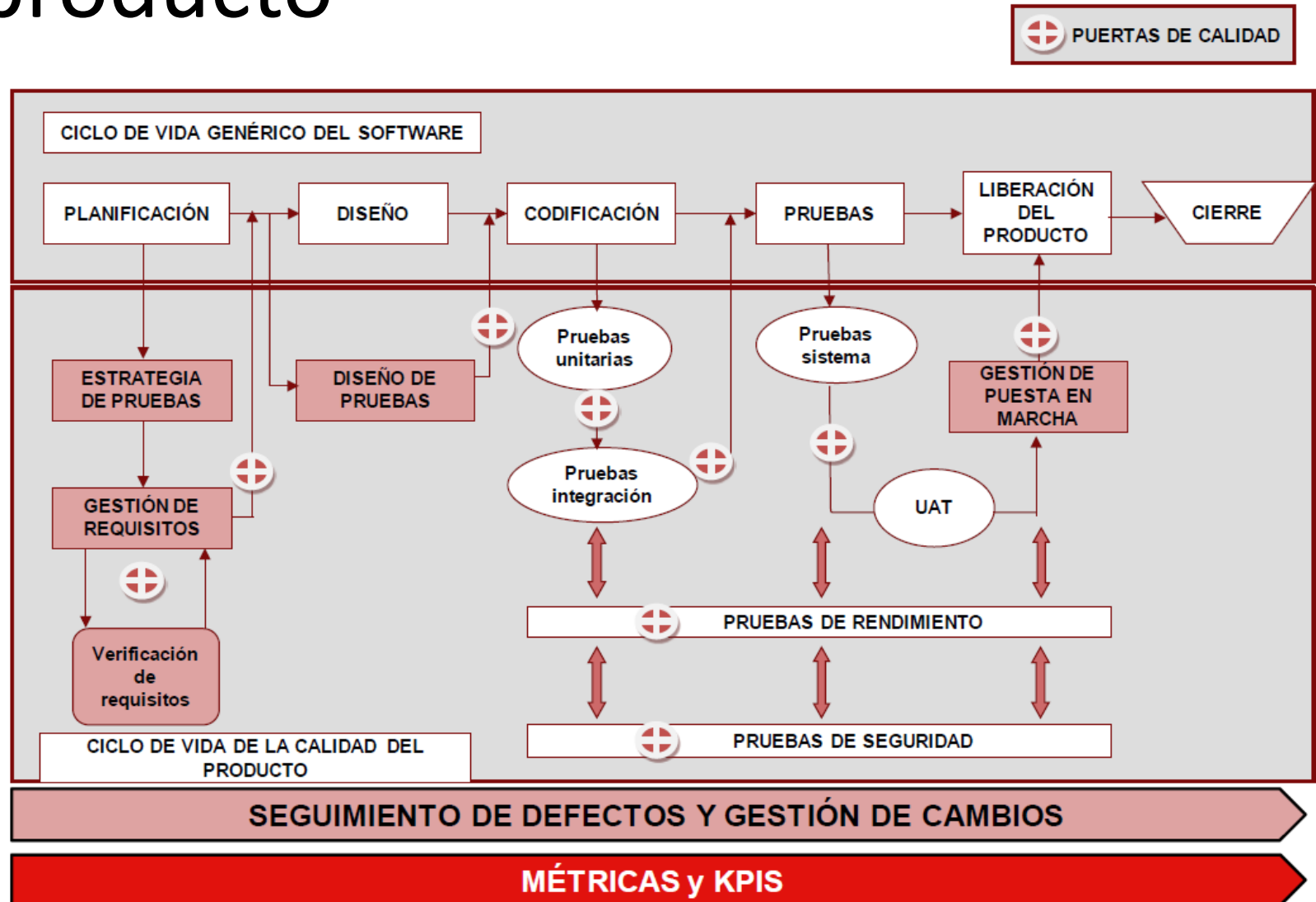
- El **coste de calidad** se define como la **suma** del **coste de conformidad** más el **coste de no conformidad**
- Los costes de alcanzar calidad (costes de conformidad) y los costes debidos a la falta de calidad (costes de no conformidad) tienen una relación inversa el uno respecto del otro
- Cuando la inversión para conseguir calidad aumenta, los costes debidos a la falta de calidad disminuyen

# Coste de la calidad

- Relación del coste con la calidad



# Gestión efectiva de la calidad del producto



# Gestión de requisitos

- Los mejores productos, desde el **punto de vista del usuario**, son aquellos creados por desarrolladores que tienen muy claro lo que se pretende conseguir con el producto y cómo obtenerlo
- Para llegar a este punto, se debe entender el trabajo del usuario, cómo afectará el producto a su trabajo y cómo se adecuará a los objetivos de la organización
- Lo que hace el producto y las condiciones que debe satisfacer en este contexto son los **requisitos del producto**

# Gestión de requisitos

- Los **requisitos** del producto deben ser **entendidos** por todas las partes (**cliente y desarrollador**) antes de que comience su construcción o el proyecto fracasará
- Sólo cuando se conocen los requisitos correctos se podrá diseñar y construir un producto que permita a los usuarios hacer su trabajo de forma que **satisfaga las necesidades del negocio**

# Gestión de requisitos

- Aunque los **desarrolladores** tienen la oportunidad de subsanar la mayoría de los errores en la definición de requisitos, muchas veces **se precipitan o hacen suposiciones** que, como consecuencia, dan lugar a un **producto erróneo** por falta de tiempo o de presupuesto entre otros motivos
- La verificación de los requisitos es un punto clave dentro del proceso de gestión de requisitos
- En esta fase, el usuario final añade criterios de aceptación para cada requisito

# Gestión de requisitos

- La **puerta de calidad** es un punto por el que pasan cada uno de los requisitos antes de formar parte de la especificación
- Una de las tareas de las puertas de calidad es **asegurarse** de que cada **requisito cumple** con el **criterio** que tiene **asignado**
- Este criterio es una medida del requisito que le hace entendible y con capacidad para ser probado

# Diseño de pruebas

- Pasos a llevar a cabo durante esta fase del ciclo de vida de calidad de un producto software:
  1. Antes de ejecutar las pruebas, es necesario saber: qué se va a probar, cuáles van a ser las entradas y los resultados producidos y cómo se van a preparar y ejecutar las pruebas
- El **análisis de pruebas** es el proceso de examinar todo aquello que pueda proporcionar información para las pruebas. Estas serán las bases para las pruebas



# Diseño de pruebas

2. El siguiente paso sería definir **condiciones de pruebas**

- Una condición de prueba es un elemento o evento de un componente o sistema que puede ser verificado mediante uno o más casos de prueba, es decir, algo que podría probarse
- Para definirlas, habrá que acudir a las bases de pruebas
- Las condiciones de pruebas deberían poderse enlazar/trazar con su origen en las bases de pruebas

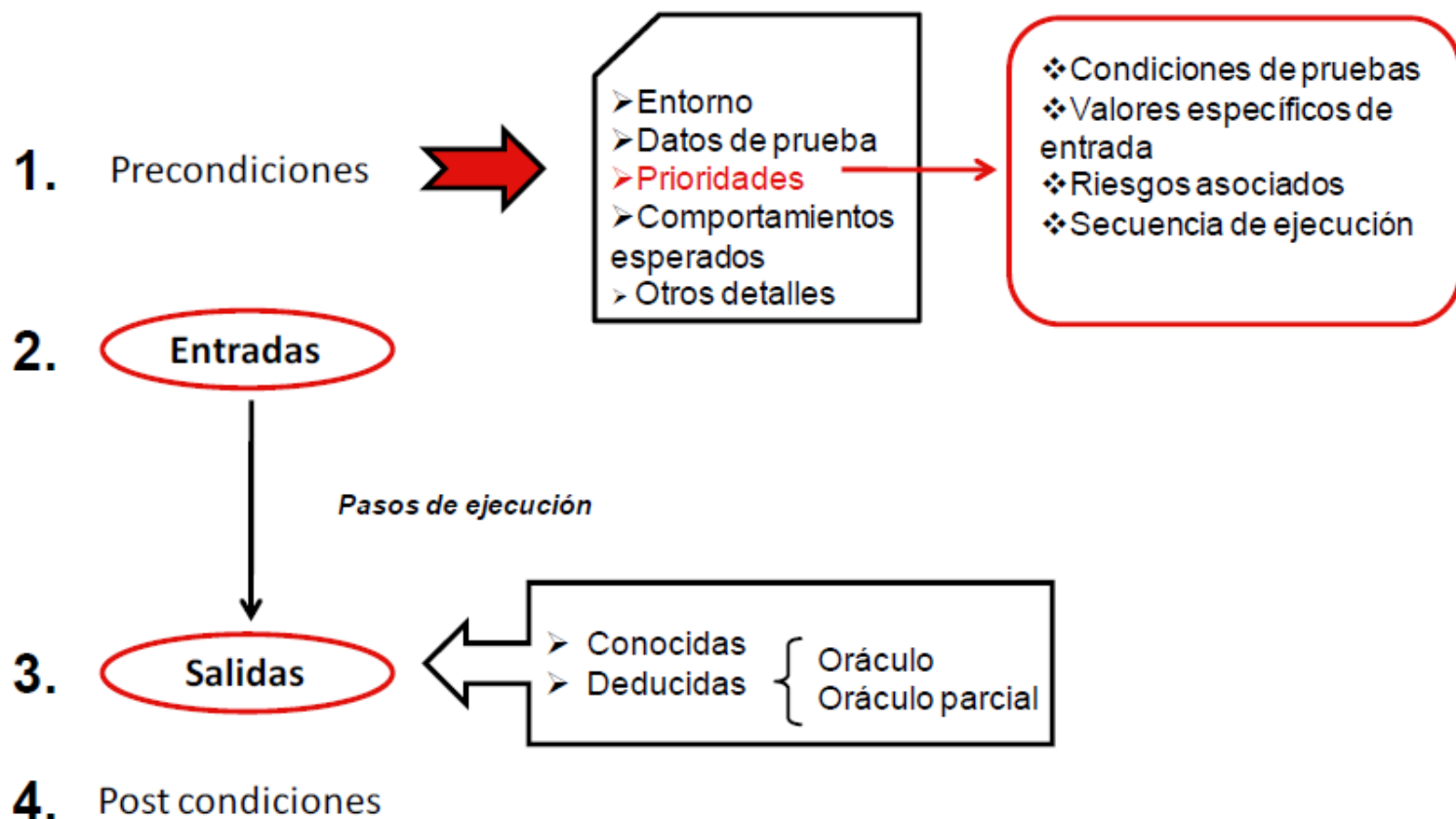
# Diseño de pruebas

## 3. Crear **casos de pruebas**

- Las condiciones de pruebas pueden ser imprecisas y cubrir un gran rango de posibilidades
- Sin embargo al crear casos de pruebas es necesaria una especificación detallada de los mismos
- No es recomendable usar descripciones generales

# Diseño de pruebas

## Casos de prueba



# Diseño de pruebas

4. Una vez que tenemos los casos de pruebas, el siguiente paso es **agrupar los casos de pruebas** en el orden de su ejecución y especificar los pasos secuenciales que se necesitan llevar a cabo para ejecutar la prueba

  - Por ejemplo, un conjunto de pruebas simples que cubra el alcance de un sistema puede formar un grupo de pruebas de regresión

# Gestión de defectos

- Es común realizar informes de defectos encontrados en el código o en el propio sistema
- Sin embargo, también se ven casos donde los defectos se refieren a los requisitos o a la especificación del diseño, a la documentación e incluso a las pruebas
- Es más, la mayoría de los errores son introducidos en las primeras fases del ciclo de vida del proyecto

# Gestión de defectos

- Encontrar defectos desde las primeras etapas tempranas del proceso de desarrollo permite su pronta corrección y que el resto de fases no se desvíen de la previsión inicial, de modo que se satisfaga finalmente tanto a las personas de negocio como a los clientes
- Sin embargo, la mayoría de ellos se detectan durante las pruebas de aceptación y producción

# Gestión de defectos

- Esto es debido a que no se dedica el esfuerzo necesario en gestión de calidad en las etapas iniciales del desarrollo de software
- Lo más habitual es dedicar el tiempo a buscar posibles defectos en la etapa de ejecución pruebas, cuando esta etapa es ya bastante tardía en el desarrollo, y los posibles defectos introducidos en fases anteriores ya se han propagado
- Esta tendencia debe cambiarse y empezar la gestión de defectos desde la etapa de definición de requisitos

# Gestión de defectos

- En fabricación, por ejemplo, un ejemplo de acciones a tener en cuenta en lo relativo a la gestión de defectos sería:
  - Cualquier persona en la planta de fábrica puede parar el proceso entero de fabricación si se dan cuenta de que algo no va bien.
  - Cualquier persona en la línea de montaje es validador de calidad.
  - Tener claro que cuando un problema es detectado pronto, es mucho más barato arreglarlo en ese momento en vez de desperdiciar un grupo de coches defectuosos, por ejemplo, y más tarde tener que rediseñarlos o incluso retirarlos del mercado.



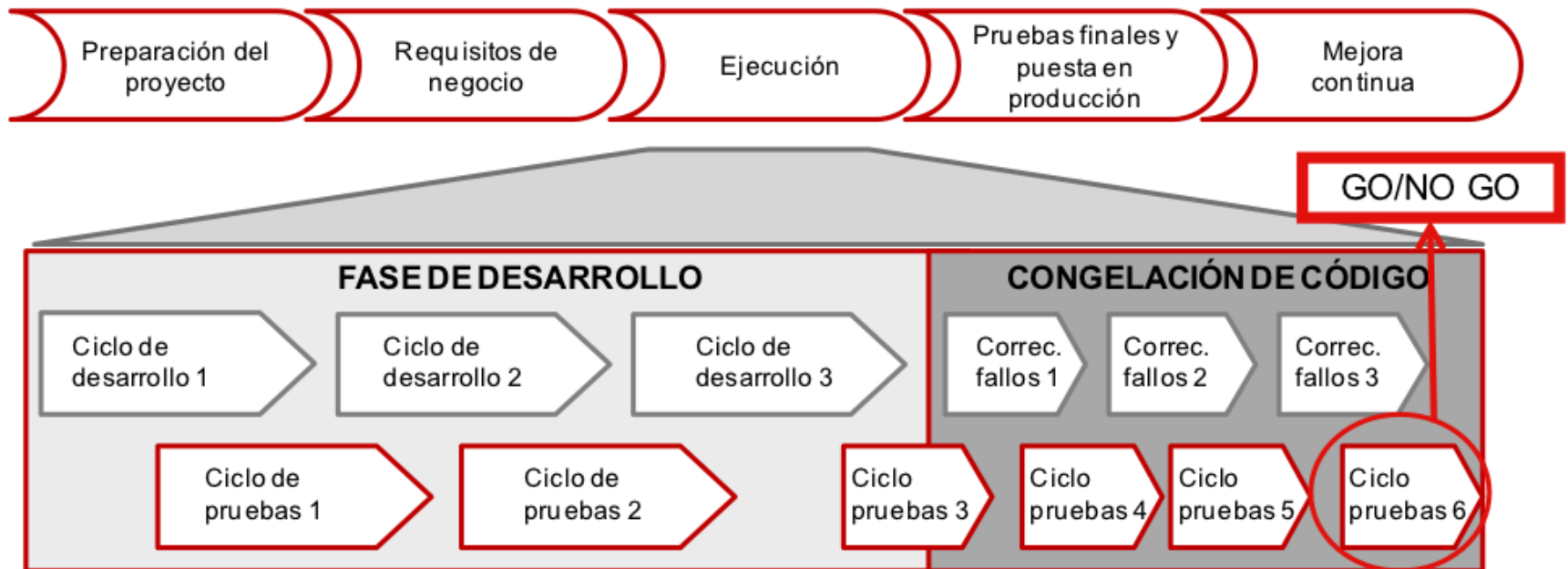
# Gestión de la puesta en marcha

- La gestión de la puesta en producción tiene como objetivos:
  - Entregar
  - Distribuir
  - Hacer un seguimiento de los cambios que se presenten en la puesta en producción
- Es conveniente que este proceso esté integrado con los procesos de:
  - Gestión de la configuración
  - Gestión de cambios

# Gestión de la puesta en marcha

- Antes de la puesta en producción hay que tener en cuenta una serie de acciones y criterios a seguir:
  - Hay que establecer políticas de congelación de código
  - Todos los requisitos han de estar cerrados
  - Se deben realizar evaluaciones de métricas de riesgos
- Deben existir estrategias para mitigar los riesgos post-puesta en producción.
- Para cumplir con este criterio se debe contar con un plan que ayude a abordar los riesgos que puedan surgir en producción

# Gestión de la puesta en marcha



# Gestión de la puesta en marcha

- Se deben identificar los distintos tipos de riesgos que pueden surgir: mejoras, solicitudes de cambio, incidentes...
- Se hace necesaria una integración entre la gestión del cambio, los requisitos que genera y el aseguramiento de la calidad

# Bibliografía

- Calidad de sistemas informáticos. Mario Piattini, Félix García e Ismael Caballero
- Ingeniería del software. Séptima edición. Ian Sommerville
- <http://www.aec.es>: Asociación Española para la Calidad
- <http://www.asq.org>: American Society for Quality