



Facultad de Ingeniería  
Universidad de Cuenca  
Grado en Ingeniería de Sistemas  
Curso 2020-2021

# Calidad de Software

## Capítulo 4: Construcción de Modelos de Calidad de Software

Departamento de Ciencias de la Computación  
Universidad de Cuenca, Ecuador  
email: [priscila.cedillo@ucuenca.edu.ec](mailto:priscila.cedillo@ucuenca.edu.ec)

# Contenido

- Introducción
- Alternativas para abordar la construcción de modelos de calidad
- Factores técnicos (funcionales y no funcionales)
- Factores no técnicos (proveedor, políticos, económicos, etc.)
- Solapamiento de características
- Características de calidad internas y externas
- Interdependencias entre características de calidad
- Métricas del software
- Métodos de construcción de modelos de calidad

# Contenido

- **Introducción**
- Alternativas para abordar la construcción de modelos de calidad
- Factores técnicos (funcionales y no funcionales)
- Factores no técnicos (proveedor, políticos, económicos, etc.)
- Solapamiento de características
- Características de calidad internas y externas
- Interdependencias entre características de calidad
- Métricas del software
- Métodos de construcción de modelos de calidad

# Introducción

- n La calidad **no** es una **propiedad fija** de un sistema de software.
- n Esta depende de las **necesidades** y **objetivos** de los stakeholders.
- n Se deben **mapear** esas necesidades a **propiedades técnicas**.
- n No solamente planearemos **que** calidad queremos tener sino adicionalmente **cómo** la construiremos y la aseguraremos.

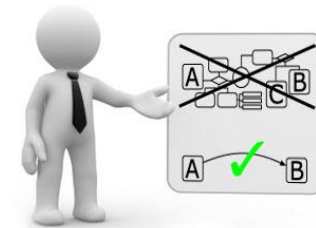
# Introducción



Stakeholders



**Necesidades y Objetivos**



**Propiedades Técnicas**



**Qué**



**Cómo**

# Contenido

- Introducción
- Alternativas para abordar la construcción de modelos de calidad
- Factores técnicos (funcionales y no funcionales)
- Factores no técnicos (proveedor, políticos, económicos, etc.)
- Solapamiento de características
- Características de calidad internas y externas
- Interdependencias entre características de calidad
- Métricas del software
- Métodos de construcción de modelos de calidad

# Alternativas para abordar la construcción de modelos de calidad

- n Los requisitos no-funcionales describen las propiedades que no representan la funcionalidad primaria del sistema
- n Son características que hacen al producto atractivo, usable, rápido, confiable
- n El problema es cómo elicitar y evaluar los requisitos de calidad en una forma estructurada y comprensible
- n Los principales enfoques para elicitar requisitos de calidad son:
  - Chequear diferentes tipos de requisitos y construir prototipos
  - Usar escenarios positivos y negativos (Casos de Uso)
  - Refinar metas de calidad por gráficos de objetivos

# Alternativas para abordar la construcción de modelos de calidad

- n Un ejemplo para chequear tipos de requisitos son los checklists para SLAs.
- n En los casos de uso, los requisitos de calidad son adjuntados como información adicional (ej. máximo tiempo de respuesta del sistema en tal caso de uso)
- n Gráficos de meta son a menudo descompuestos de metas de calidad a sentencias lógicas para luego operacionalizarlas como requisitos funcionales.





## Alternativas para abordar la construcción de modelos de calidad

- n Especificar requisitos de calidad y construir modelos de calidad pueden fusionarse en un solo proceso.
- n Mientras añadimos nueva información a un modelo de calidad cuando lo construimos, especificamos requisitos y escogemos las partes necesarias de modelos existentes.

# Contenido

- Introducción
- Alternativas para abordar la construcción de modelos de calidad
- Factores técnicos (funcionales y no funcionales)
- Factores no técnicos (proveedor, políticos, económicos, etc.)
- Solapamiento de características
- Características de calidad internas y externas
- Interdependencias entre características de calidad
- Métricas del software
- Métodos de construcción de modelos de calidad

# Factores Técnicos

- n Descomposición de factores en factores de más bajo nivel de abstracción
- n Sólo ocasionalmente a la eliminación o modificación de algún factor ya existente
- n Los factores técnicos se refieren a la calidad intrínseca del producto de software.

# Factores No-Técnicos

- n Carvallo, Franch y Quer destacan que se pueden incluir en los modelos de calidad factores no-técnicos, y ellos así lo hacen; como ejemplo tenemos:
- Proveedor: Características del proveedor que pueden influenciar en la calidad del producto de software
  - Negocio: Características del contrato entre proveedor y cliente que pueden influenciar la calidad del producto del software
  - Producto: Características de los aspectos comerciales del producto que pueden influenciar su calidad.

# Contenido

- Introducción
- Alternativas para abordar la construcción de modelos de calidad
- Factores técnicos (funcionales y no funcionales)
- Factores no técnicos (proveedor, políticos, económicos, etc.)
- **Métricas del software**
- Solapamiento de características
- Características de calidad internas y externas
- Interdependencias entre características de calidad
- Métodos de construcción de modelos de calidad

# Métricas del Software

- n Las métricas son un buen método para medir, entender, monitorizar, predecir y probar el desarrollo de software y los proyectos de mantenimiento (Briand et al., 1996)
- n La medición persigue tres objetivos fundamentales:
  - Entender qué ocurre durante el desarrollo y el mantenimiento.
  - Controlar qué es lo que ocurre en los proyectos de desarrollo.
  - Mejorar los procesos y productos.
- n Se pueden ver a las medidas como medios para asegurar la calidad en los producto de software.

# Métricas del Software

## n Problemas

- Todavía se falla en dar objetivos medibles cuando desarrollamos productos de software.
  - Ejm. Se dice que será usable sin especificar esa usabilidad en términos medibles.
- Fallamos en medir diferentes componentes para calcular costes reales de proyectos.
  - Ejm. No sabemos cuánto tiempo fue invertido en el diseño comparado con las pruebas.
- No intentamos cuantificar la calidad de los productos que producimos.
  - Ejm. No podemos decir a un usuario cómo de fiable va a ser un producto en términos de fallos en un período de uso dado.



# Métricas del Software

## n Problemas

- Solemos ver informes que hacen afirmaciones como que el 80% de los costes del software son de mantenimiento o que hay una media de 55 errores en cada 1.000 líneas de código.
- Pero no se dice
  - Cómo se han obtenido esos resultados
  - Cómo se han ejecutado los experimentos
  - Qué entidades fueron medidas y cómo
  - Los márgenes de error.

# Atributos Medibles

- n En software hay **tres clases de entidades** cuyos atributos podemos medir:
- **Procesos:** actividades del desarrollo
  - **Productos:** entregables, artefactos o documentos generados en el ciclo de vida del software
  - **Recursos:** todos aquellos elementos que hacen de entrada a la producción de software.

# Atributos a medir

## n Procesos:

- El tiempo (duración del proceso)
- El esfuerzo (asociado al proceso)
- El número de incidentes de un tipo específico

## n Productos:

- La fiabilidad del código
- La entendibilidad de un documento de especificación
- La mantenibilidad de un código fuente.
- La longitud, funcionalidad, modularidad o corrección sintáctica de los documentos de especificación

## n Recursos:

- El personal
- Los materiales
- Las herramientas y métodos
- El coste
- La productividad

# Utilidad de las Métricas del Software

- n Estimación del esfuerzo
- n Modelos y medidas de productividad
- n Modelos y medidas de calidad
- n Modelos de fiabilidad
- n Modelos de evaluación y rendimiento
- n Métricas de complejidad estructural

# Ejemplo

- n Se quiere contratar software as a service (SaaS) de supermercados.
- n El software debe estar disponible. La intensidad de fallos del software es 1 fallo por 100 CPU/hr. El sistema especializado ejecuta 20 CPU/hr por semana y hay 800 clientes que usarán el servicio.
- n Se requiere proporcionar a los clientes un servicio de reparación.
- n Cada empleado puede realizar 4 llamadas de servicio por día y el servicio estará disponible 5 días por semana.
- n ¿Cuántos empleados necesitamos contratar?

- n Usando el valor de intensidad de fallos, cada sistema experimenta 0.2 fallos por 20 horas de operación o (0.2 fallos por semana en promedio)
- n El total de fallos para 800 clientes es de 160 por semana o 32 por día.
- n Cada empleado puede realizar 4 visitas por día. Así el número de empleados debería ser  $32/4=8$ .

# Contenido

- Introducción
- Alternativas para abordar la construcción de modelos de calidad
- Factores técnicos (funcionales y no funcionales)
- Factores no técnicos (proveedor, políticos, económicos, etc.)
- Solapamiento de características
- Características de calidad internas y externas
- Interdependencias entre características de calidad
- Métricas del software
- **Métodos de construcción de modelos de calidad**

# Pasos para construir el modelo a través de la Ingeniería de Requisitos

1. Identificar a los stakeholders relevantes
2. Definir metas generales
3. Analizar documentos relevantes y elicitar nueva información
4. Escoger y definir actividades
5. Definir metas de calidad
6. Identificar artefactos afectados y escoger entidades
7. Escoger entidades y analizar material relevante
8. Escoger factores y definir nuevas características de producto
9. Especificar requisitos de calidad



## Otras propuestas..

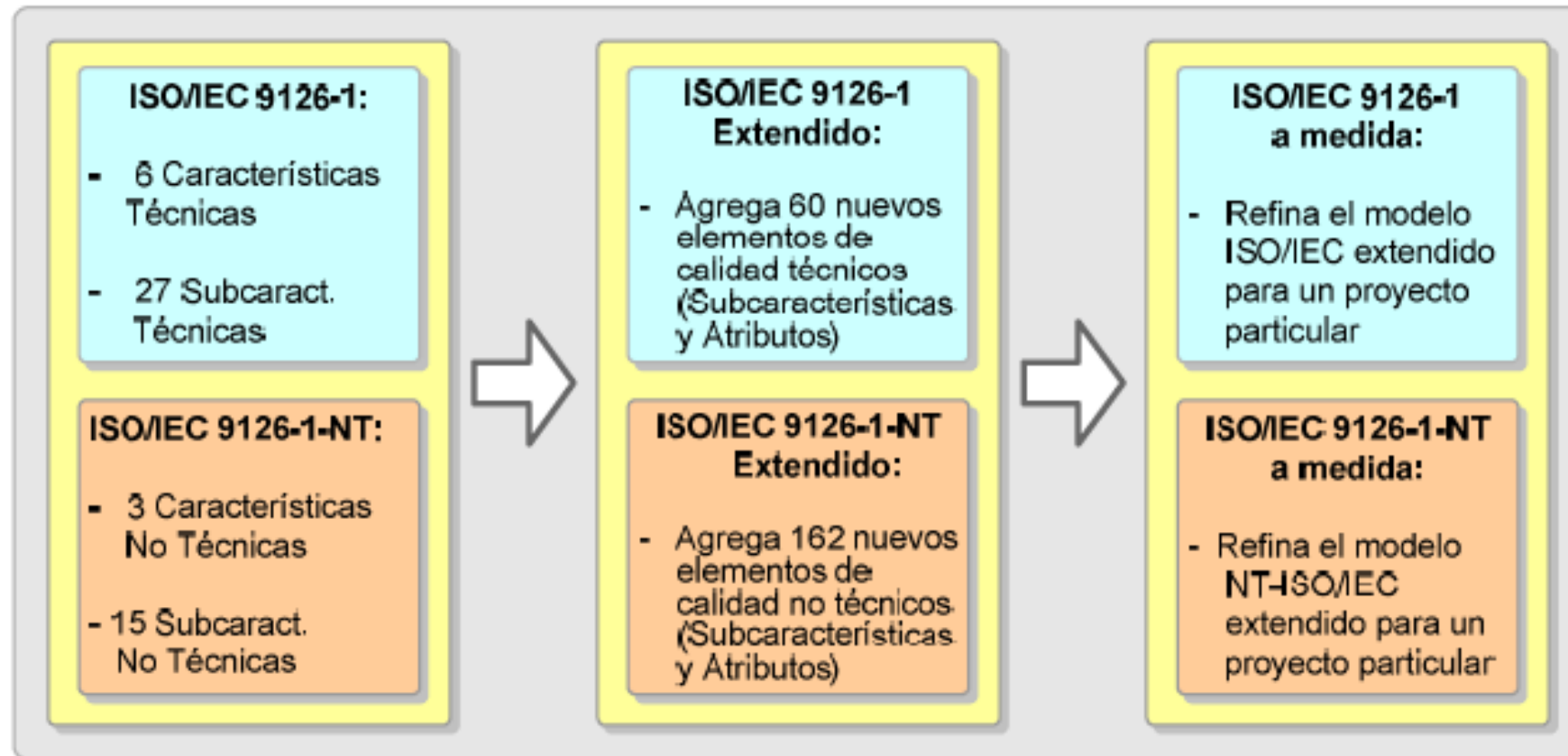
- n Por su complejidad se requiere de una guía para su construcción.
- n Existen propuestas generales:
  - Bohem-78: 6 pasos
  - GQM: 5 pasos
  - Dromey: 5 pasos
- n Método IQMC (Individual Quality Model Construction)
  - Diseñado para producir modelos compatibles con el estándar ISO/IEC 9126-1

# El método IQMC para la construcción de modelos de calidad

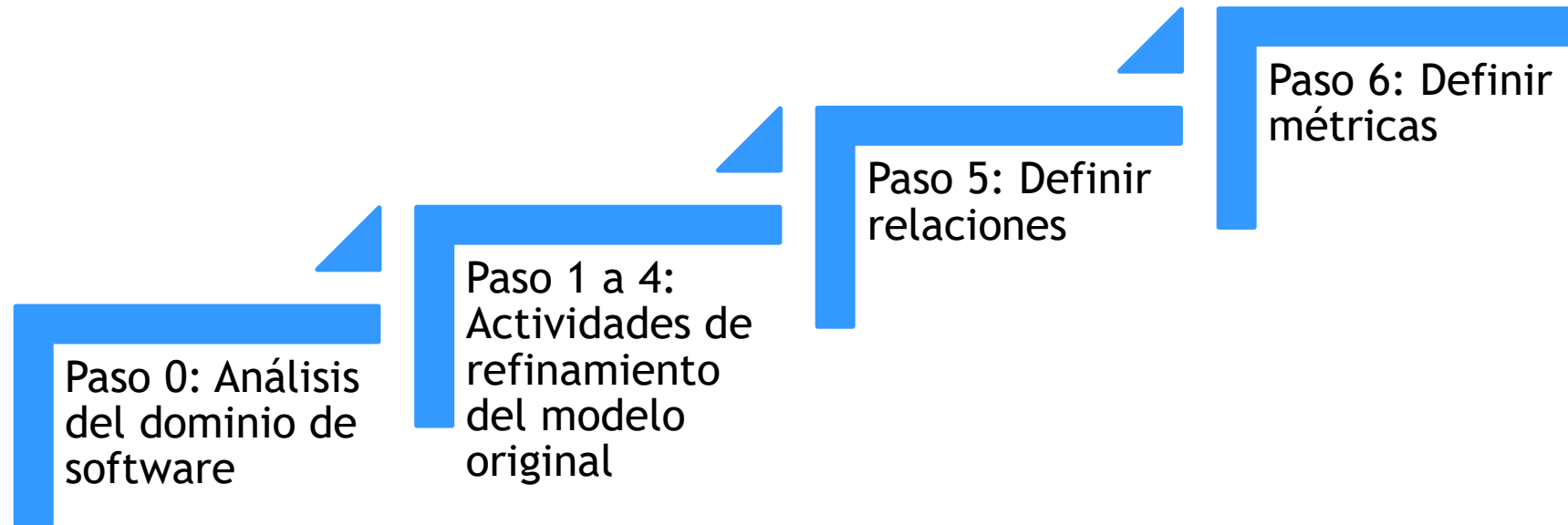
- n Circunstancias que intervienen en la construcción de un modelo de calidad:
1. El equipo realiza la construcción del modelo, en el caso de que este equipo **no tenga experiencia** en la construcción de modelos o en el contexto del dominio.
  2. El dominio para el que se construye el modelo, para el que en muchas ocasiones **no existe una terminología común**.
  3. **Factores metodológicos**, ya que es difícil conocer el nivel de profundidad hasta el que es necesario descomponer los modelos.

**IQMC** : Conjunto de guías y técnicas para la identificación de los factores de calidad apropiados que deben ser incluidos en un modelo de calidad que permita analizar la calidad de componentes pertenecientes a un cierto dominio de software.

# Extensión del ISO 9126-1



# Método IQMC



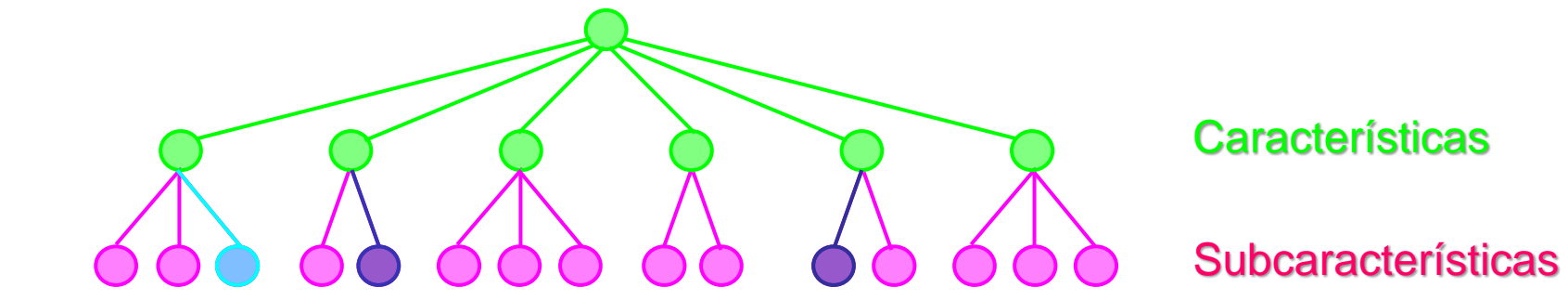
- El estándar IQMC consiste en 7 pasos que aunque se presentan como secuenciales, pueden ser simultáneos de ser necesario.
- Es importante hacer hincapié en la diferencia de detalle que existe en el catálogo entre los factores técnicos y no-técnicos.

# Paso 0

- n Estudio del ámbito del software.
- n Paso opcional que puede soslayarse en caso de poseer el conocimiento suficiente.
- n Puede ser necesario realizar algún tipo de modelización para realizar una unificación de la terminología identificada en las distintas fuentes de información de cara a los pasos siguientes.
- n Tener cuidado con la terminología a ser empleada, muchas personas utilizan diferentes términos para un mismo concepto o lo que es peor el mismo término para diferentes conceptos.

# Paso 1

- n Determinación de sub-características de calidad. Teniendo en cuenta que partimos del catálogo ISO/IEC 9126-1 extendido, el añadido de sub-características no será muy habitual.
- n Puede pasar que alguna sub-característica deba reformularse ligeramente para adaptarla al dominio de interés, o eliminarse.



## Paso 2

- n Refinamiento de la jerarquía de sub-características. Se descomponen las sub-características del más bajo nivel de abstracción formando jerarquías de sub-características.
- n Al igual que en el paso anterior, el añadido de sub-características no será muy habitual, excepto en el caso de la descomposición de la sub-característica.
- n En cuanto a sub-características no-técnicas, lo que se realizará es un purgado de las sub-características que no interesen al proyecto en cuestión.

## Paso 3

- n Refinamiento de sub-características en atributos.
- n Descomposición de esos atributos de calidad en unos más concretos.
- n Objetivos: llegar a tener descompuestas las sub-características en atributos medibles ya sea de forma directa o indirecta a través del valor de otro atributos básicos.
- n Un atributo puede estar en más de una sub-característica.



## Paso 4

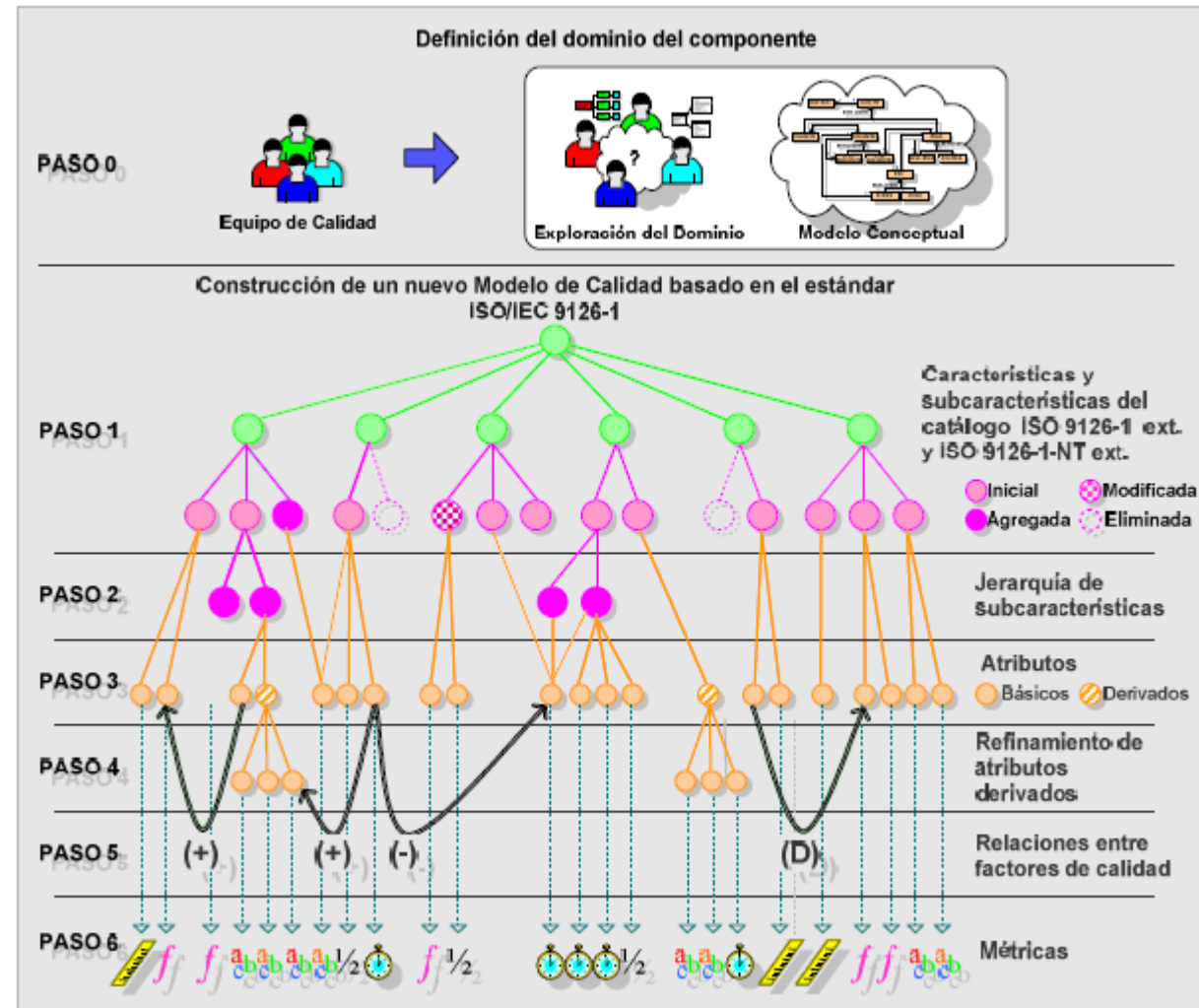
- n Atributos básicos y derivados
- n Refinamiento de atributos derivados en básicos. Se descomponen los atributos derivados hasta obtener los atributos básicos.
- n Una vez realizado lo anterior los atributos básicos pueden ser medidos de forma directa.

## Paso 5

- n Determinar las métricas para los atributos básicos.
- n Establecimiento de relaciones entre factores de calidad.
- n Permiten conocer las dependencias entre los distintos factores de calidad del modelo.
- n Colaboración, daño o dependencia

## Paso 6

- n Determinación de métricas para los atributos. Se determinan las métricas para los atributos identificados.



# Goal Question-Metric

- n Técnica definida por Basili y Weiss (1984) y Rombach (1990), para seleccionar y generar métricas tanto del proceso como de los resultados del proyecto.



- n Propone definir un objetivo, refinarlo en preguntas y definir métricas que intenten dar información para responder estas preguntas.

# Fases del método GQM

## n Planificación

- Se selecciona, define, caracteriza y planifica un proyecto para la aplicación de la medición, obteniéndose como resultado un **plan del proyecto**.

## n Definición

- Se **define y documenta el programa de la medición** (objetivos, preguntas, métricas e hipótesis).

## n Recopilación de Datos

- Se reúnen los **datos reales de la medición**.

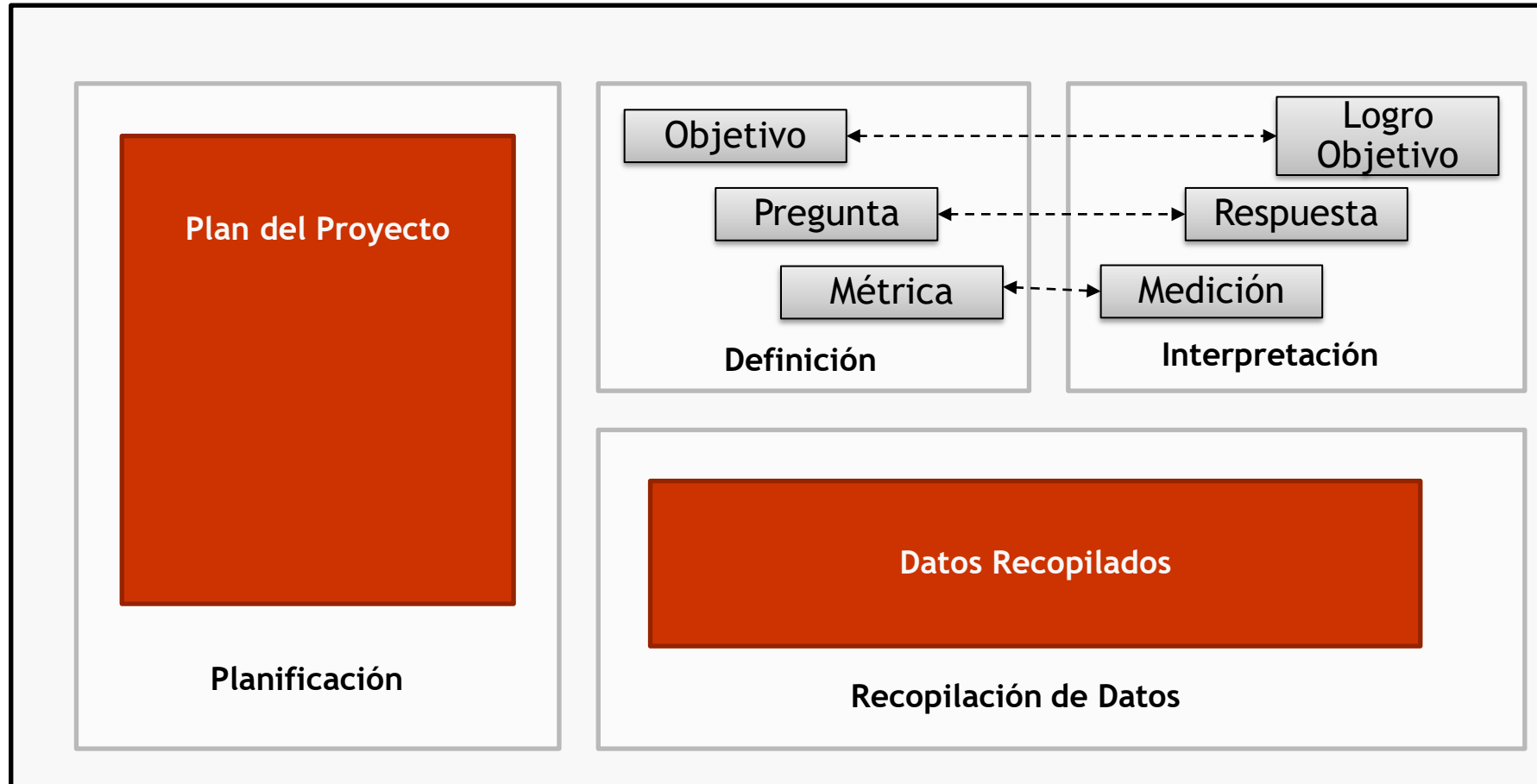
## n Interpretación

- Se **procesan los datos** recopilados respecto a las métricas definidas en forma de resultados de medición, que proporcionan **respuestas a las preguntas** definidas a partir de las cuales se puede evaluar el logro del objetivo planteado.

# Planificación

- n Recopilar toda la información necesaria para un inicio exitoso de un proyecto de medición.
- n Motivación y preparación de los miembros de la organización para llevar a cabo el programa de medición.
- n Plan de proyecto - producto principal de esta fase.
- n Incluye:
  - Documentos
  - Procedimientos
  - Calendarios
  - Objetivos del programa de medición.

# El proceso GQM



El Proceso GQM (van Soligen y Berghout, 1999)

# Planificación

## n Establecer el equipo GQM

- Garantiza la continuidad de los programas de medición.
- Cuando existe apuro se descuida, por lo cual GQM debería tener las siguientes cualidades:
  - Ser independiente de los equipos del proyecto.
  - No ser “parte interesada”
  - Poseer conocimiento previo sobre los objetos de la medición.
  - Conciencia de su rol.
  - Mentalidad de orientación a la mejora.



# Planificación

## n Seleccionar Áreas de Mejora

- Problemas evidentes a los que se enfrenta la organización.
- Areas a mejorar identificadas, en base a los objetivos del negocio.
- Problemas que podrían ocurrir.
- Influencias externas.
- Tecnologías.
- Leyes.
- Procesos y productos.
- Experiencia en medición de las personas implicadas.

# Planificación

- n Seleccionar el proyecto de aplicación y establecer un equipo del proyecto.
  - Depende de la voluntad, motivación y entusiasmo de los miembros del equipo de proyecto.
  - Los objetivos de la medición deben estar alineados con las ideas de mejora del proyecto.

# Planificación

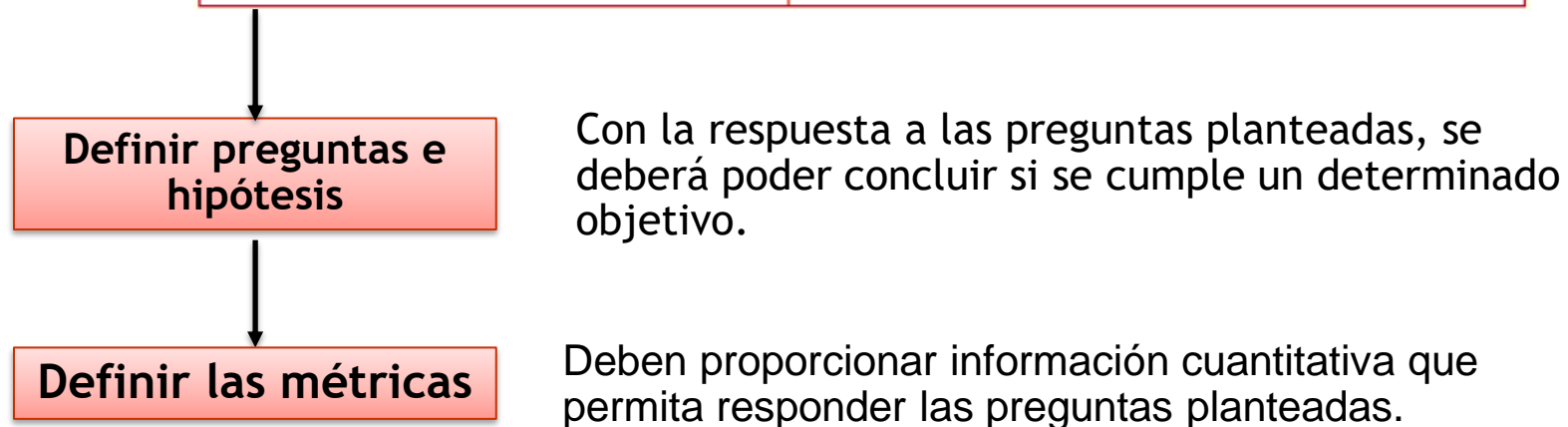
## n Crear el plan del proyecto

- Resumen ejecutivo: presentar en 20 líneas aprox. el programa de medición.
- Introducción: Presenta el alcance del programa de medición. Relación entre los objetivos de la mejora y los objetivos del proyecto de desarrollo de software.
- Calendario: incluye planificación temporal, entregables, asignación de recursos y análisis coste-beneficio del programa de medición.
- Organización: Estructuras organizacionales del proyecto y equipo GQM que son relevantes para el programa de medición.
- Procesos de gestión: Contiene prioridades, procedimientos de generación de informes de gestión así como actividades de control de riesgos.
- Formación y promoción: Plan para la formación de los miembros del equipo del proyecto y la comunicación de los resultados de la organización.

# Definición

- Definir los objetivos de la medición
  - Se obtiene una definición formal y bien estructurada de los objetivos, para lo cual se utilizan plantillas como la que se muestra:

<b>Analizar</b>	el objeto bajo medición
<b>Con el propósito de</b>	entender, controlar, o mejorar el objeto
<b>Con respecto a</b>	el enfoque de calidad del objeto en el que se centra la medición
<b>Desde el punto de vista de</b>	las personas que miden el objeto
<b>En el contexto de</b>	el entorno en el que la medición tiene lugar



# Definición

Revisar o elaborar los modelos de proceso software

-Dan soporte a la definición de las mediciones

Realizar entrevistas GQM

-Se extrae del equipo toda la información relevante

Definir y revisar preguntas e hipótesis

Definir las métricas

# Ejemplo

## n Métricas para las BD Relacionales

### n Objetivo GQM:

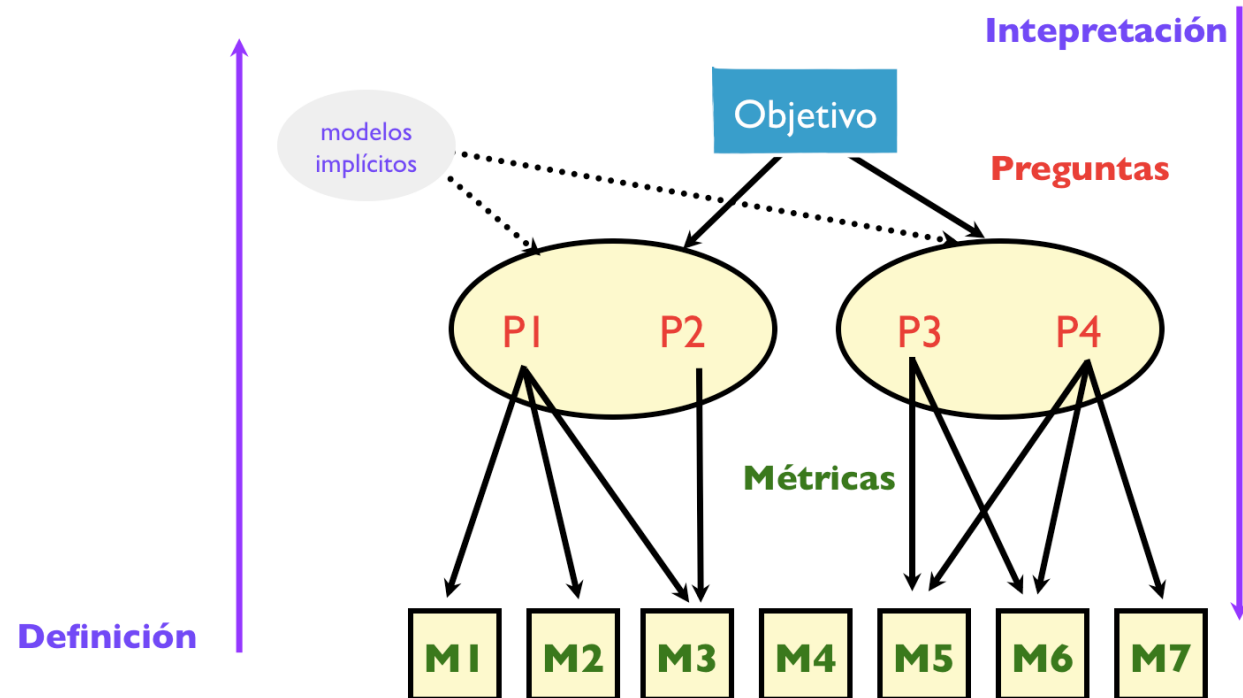
- Analizar: Bd Relacionales
- Con el propósito de: Asegurar
- Con respecto a: La mantenibilidad
- Desde el punto de vista de: BD Relacional

### n Preguntas

- Cómo influye la **complejidad de las tablas** en la mantenibilidad de las BD Relacionales.
- Cómo influye la **complejidad entre tablas** en la mantenibilidad de las BD Relacionales

# Niveles GQM

- n Se consituyen 3 niveles dentro de esta técnica
- Nivel Conceptual (Objetivos-Goals)
  - Nivel Operacional (Preguntas - Questions)
  - Nivel Cuantitativo (Métricas - Metrics)



# Ejemplo

## n Métricas

### ▪ Pregunta 1

- $NA(T)$  - NÚMERO DE ATRIBUTOS DE UNA TABLA
- $NFK(T)$  - NÚMERO DE CLAVES AJENAS
- $RFK(T)$  - RATIO DE CLAVES AJENAS DE UNA TABLA

$$RFK(T) = \frac{NFK(T)}{NA(T)}$$

### ▪ Pregunta 2

- $NT$  - NÚMERO DE TABLAS
- $NA$  - NÚMERO DE ATRIBUTOS
- $NFK$  - NÚMERO DE CLAVES AJENAS (NFK)



# Bibliografía

- n Galin D., Software Quality Assurance From theory to implementation, 2004
- n Chappell, D. (2012). THE THREE ASPECTS OF SOFTWARE QUALITY : FUNCTIONAL , STRUCTURAL , AND PROCESS Sponsored by Microsoft Corporation. *David Chappel & Associates, 1.0*. Retrieved from [http://www.davidchappell.com/writing/white\\_papers/The\\_Three\\_Aspects\\_of\\_Software\\_Quality\\_v1.0-Chappell.pdf](http://www.davidchappell.com/writing/white_papers/The_Three_Aspects_of_Software_Quality_v1.0-Chappell.pdf)
- n O'Regan, G. (2014). *Introduction to Software Quality*. <http://doi.org/10.1007/978-3-319-06106-1>
- n Software, D., Rosa, V., & Zepeda, V. (2012). Metodología para el Aseguramiento de la Calidad en la Adquisición del Software ( proceso y producto ) y servicios.
- n Wagner, S. (2013). *Software Product Quality Control*. <http://doi.org/10.1007/978-3-642-38571-1>
- n Carvallo J. P., Presentaciones, 2014-2015.