

Auditoría, Calidad y Gestión de Sistemas

ACG

Tema 1. Calidad



- Sistemas de información
- ¿Qué es la calidad?
- Atributos de calidad
- Trazabilidad y control de requisitos
- Control de calidad
- El coste de la calidad
- Sistema de calidad
- Estándar de calidad y modelos de madurez



- **Sistemas de información**
- ¿Qué es la calidad?
- Atributos de calidad
- Trazabilidad y control de requisitos
- Control de calidad
- El coste de la calidad
- Sistema de calidad
- Estándar de calidad y modelos de madurez



¿Qué entendemos por sistema de información?





¿Qué entendemos por sistema de información?

Creo que no

Creo que sí

No sé...



- Sistemas de información
- **¿Qué es la calidad?**
- Atributos de calidad
- Trazabilidad y control de requisitos
- Control de calidad
- El coste de la calidad
- Sistema de calidad
- Estándar de calidad y modelos de madurez



¿Qué es la calidad?



def 1. Es el cumplimiento con:

- los **requerimientos** funcionales y de rendimiento **explícitamente** establecidos,
- los **estándares** de desarrollo **explícitamente** documentados
- con las **características implícitas** que se esperan de todo software desarrollado profesionalmente.

(Pressman, 1998)



def 2. El grado con el que un sistema, componente o proceso cumple con los requisitos especificados y las necesidades o expectativas del cliente o usuario.

(IEEE.Std.610-1990)

(ISO/IEC 25010 – SQuaRe)



def 3. La totalidad de características de un producto de software que tienen como habilidad, satisfacer necesidades explícitas o implícitas.

(ISO/IEC DEC 91260)



- cumplir con
- especificaciones/requisitos
- de
- características/necesidades
 - explícitas
 - implícitas



¿Existen características estándar relacionadas con la calidad del software?

Los **modelos de calidad** identifican estas características a través de **atributos de calidad**.

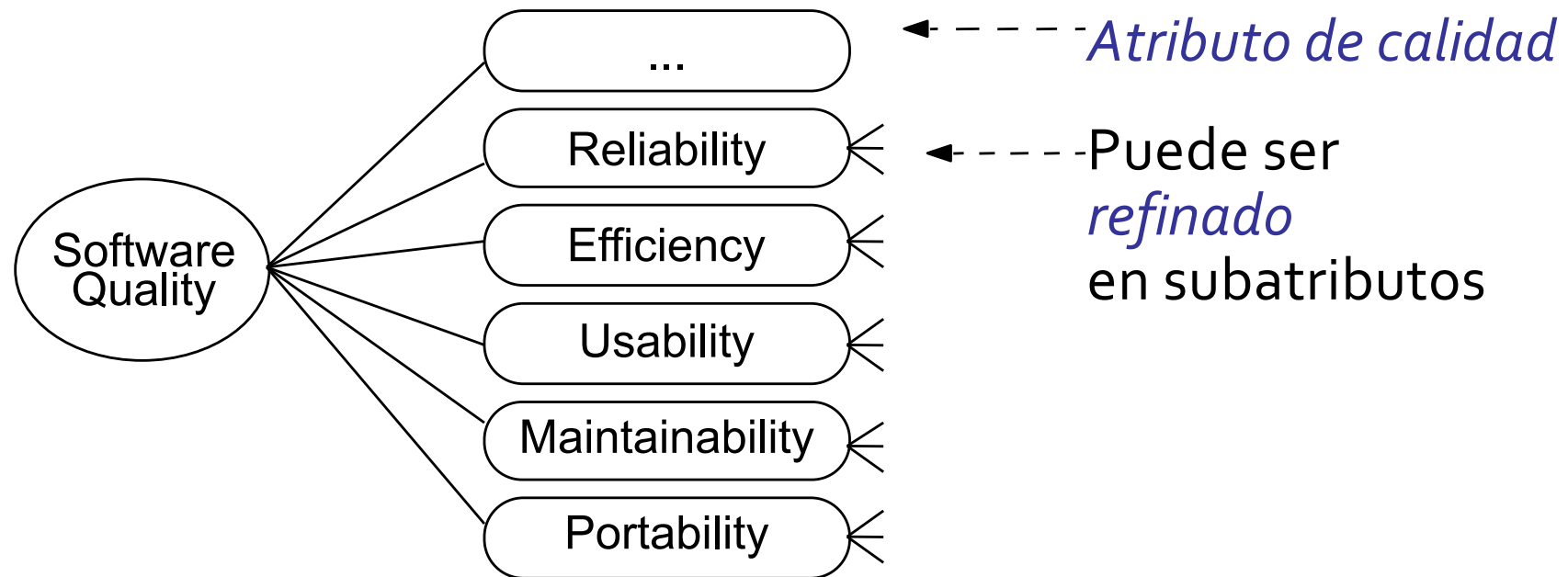


- Sistemas de información
- ¿Qué es la calidad?
- **Atributos de calidad**
- Trazabilidad y control de requisitos
- Control de calidad
- El coste de la calidad
- Sistema de calidad
- Estándar de calidad y modelos de madurez



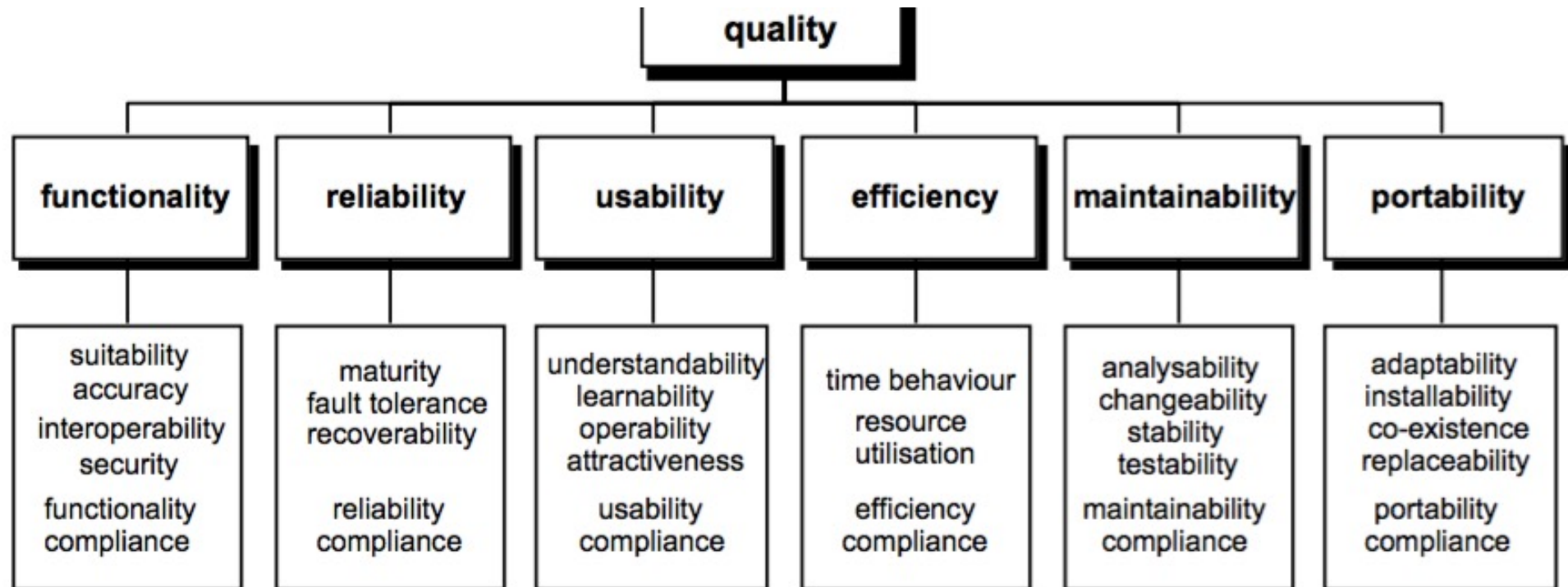
Definir la calidad a través de un modelo de calidad jerárquico, es decir, un conjunto de atributos de calidad (a.k.a. quality factors, quality aspects, ...) organizados en estructura de árbol.

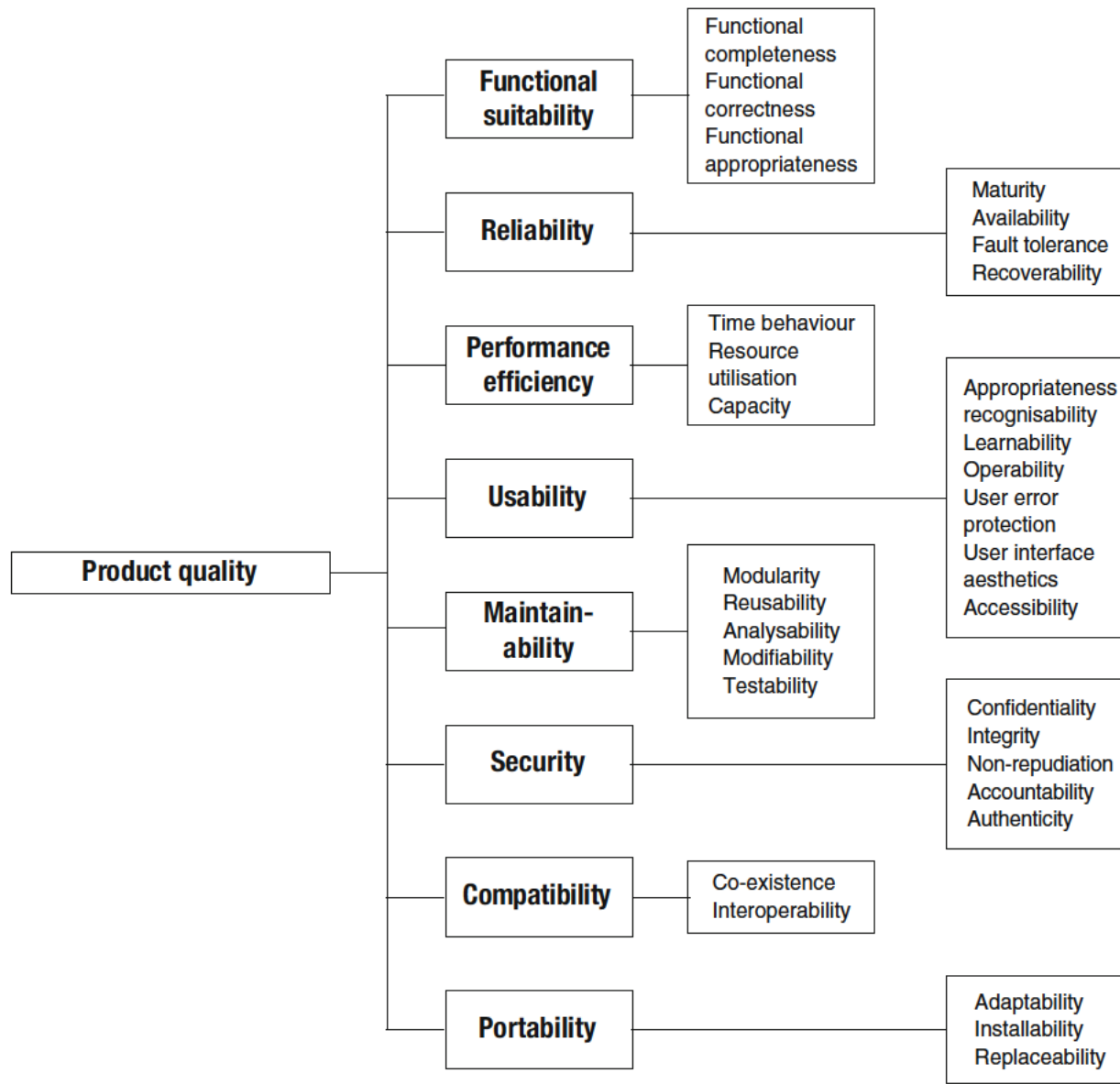
Elegir atributos de calidad (y pesos) dependiendo del contexto del Proyecto.



ATRIBUTOS DEL MODELO DE Mc CALL









¿Con qué atributo de calidad se relacionan las frases?



Mantenibilidad

- Cómo de fácil es *cambiar* un sistema después de su lanzamiento inicial
 - Cambios necesarios para:
 - correcciones
 - mejoras
 - requisitos cambiantes
 - Entropía del software (software entropy) \Rightarrow medida de desorden del software que refleja la complejidad de su mantenimiento. A medida que se hacen modificaciones o se agrega nuevo código este va perdiendo su estructura inicial y aumenta su entropía.



La mantenibilidad a menudo se refina en...

Repairability

- Cuánto trabajo es necesario para *corregir* un defecto

Evolvability (Adaptability)

- Cuánto trabajo es necesario para *adaptarse* a requisitos cambiantes

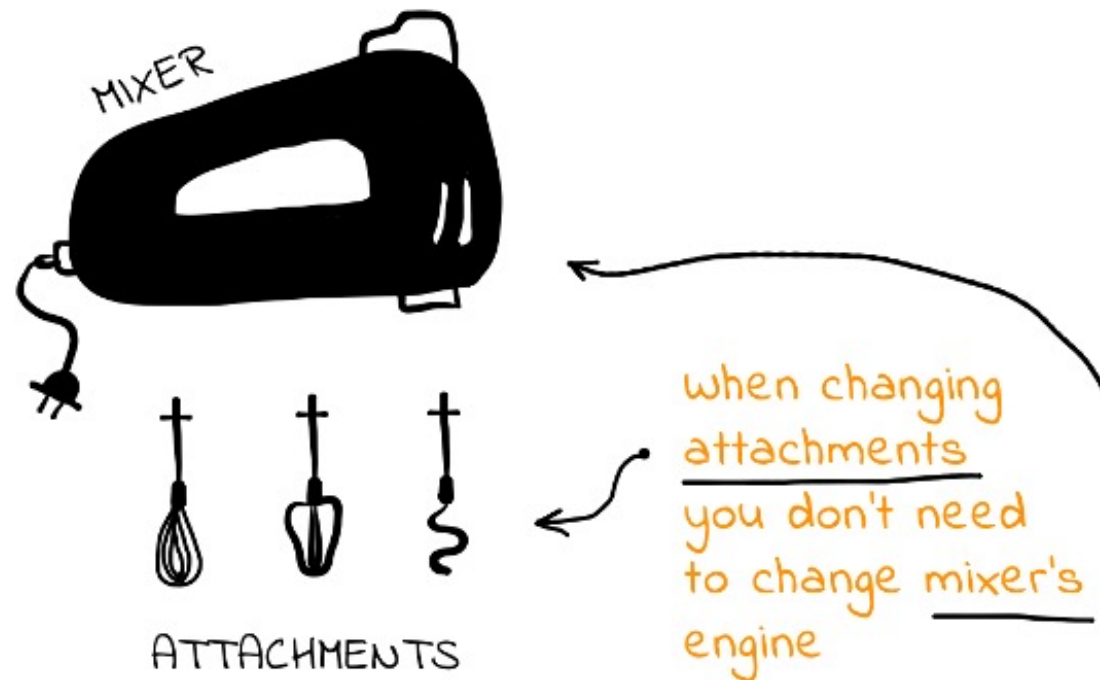
Portability

- Cuánto trabajo es necesario para *portar/trasladar* el sistema a un nuevo entorno o plataforma



0 – OPEN/CLOSED PRINCIPLE

Software entities should be open for EXTENSION, but closed for MODIFICATION. Allow behaviour to be extended without modifying the source code.



```
public class EmployeeSalary {  
  
    public Long calculateSalary(Employee emp) {  
        Long salary = null;  
  
        if (emp.getType().equals("PERMANENT")) {  
  
            salary = (totalWorkingDay * basicPay) + getCompanyBenefits() + getBonus();  
        } else if (emp.getType().equals("CONTRACT")) {  
  
            salary = (totalWorkingDay * basicPay);  
        }  
        return salary;  
    }  
}
```



@vikasrajputin

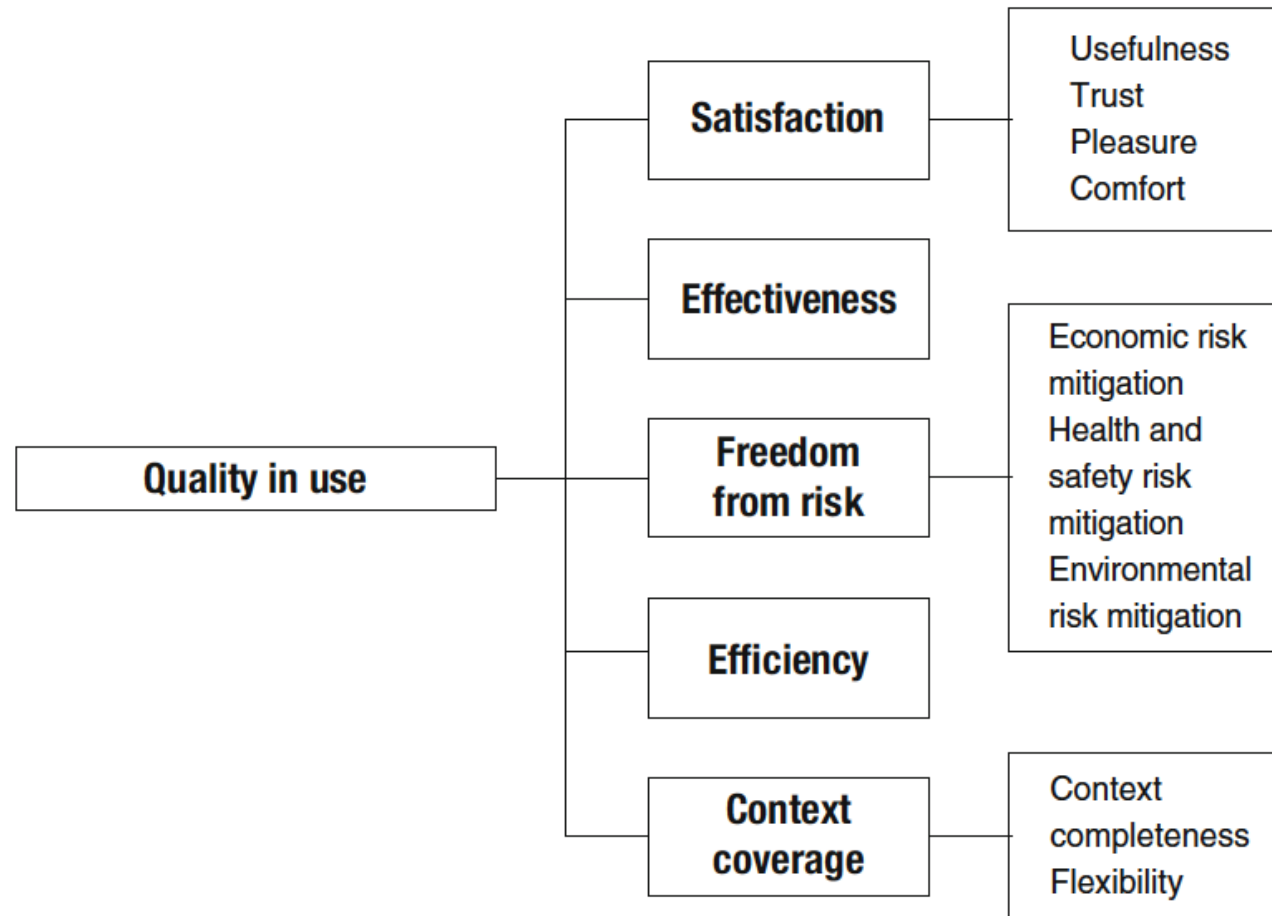
```
public interface EmployeeSalary {  
    public Long calculateSalary();  
}
```

```
public class PermanentEmployeeSalary implements EmployeeSalary{  
  
    @Override  
    public Long calculateSalary() {  
        return (totalWorkingDay * basicPay);  
    }  
}
```

```
public class ContractEmployeeSalary implements EmployeeSalary{  
  
    @Override  
    public Long calculateSalary() {  
        return (totalWorkingDay * basicPay)  
            + getCompanyBenefits() + getBonus();  
    }  
}
```

Open - Close Principle

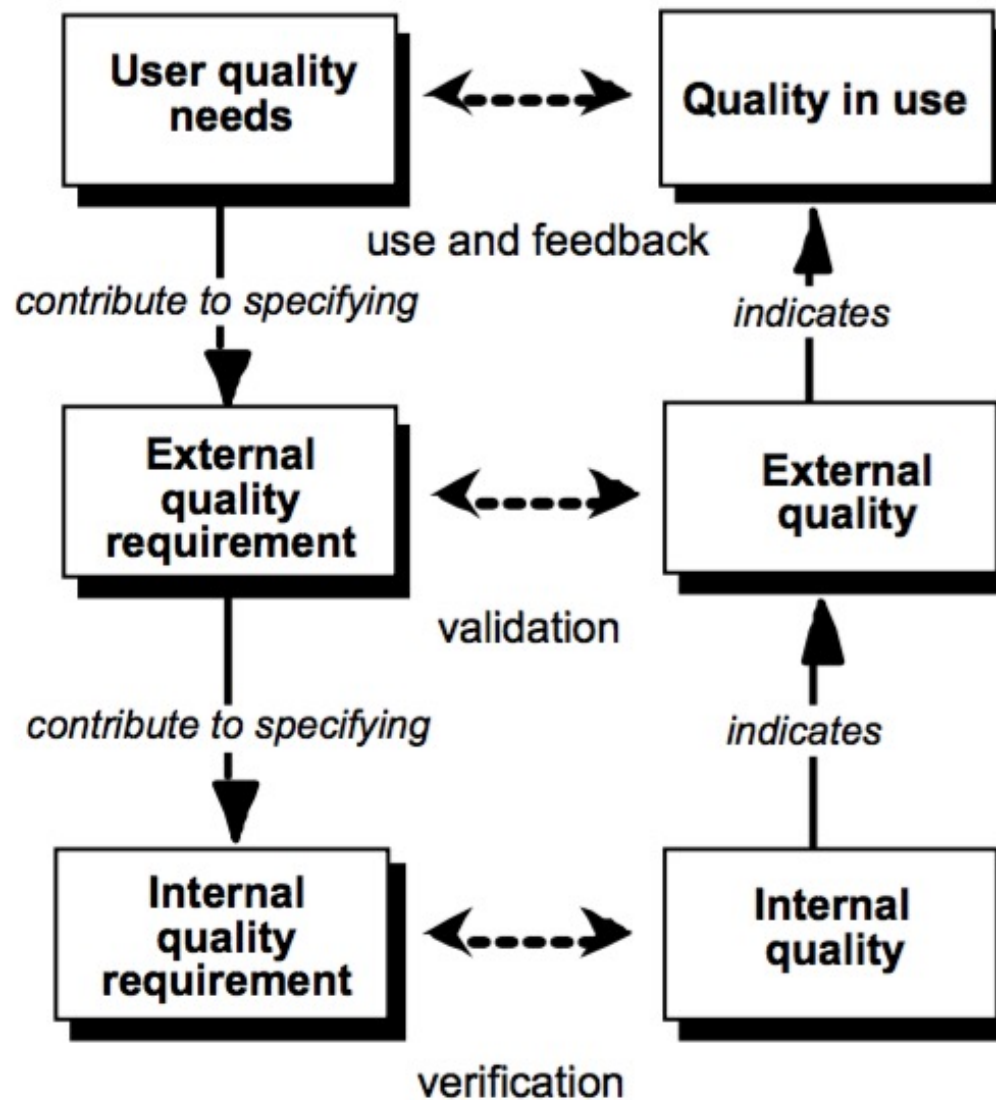




Los atributos de calidad pueden ser internos o externos:

- **Interno:** Derivados directamente de la descripción del producto o del proceso
 - e.g. Conformidad con los requisitos especificados
 - e.g. Conformidad con los estándares de código
- **Externo:** Derivados de la relación entre el entorno y el sistema (para ello, el proceso o el sistema debe ejecutarse)
 - e.g. Reliability, Robustness, Usability

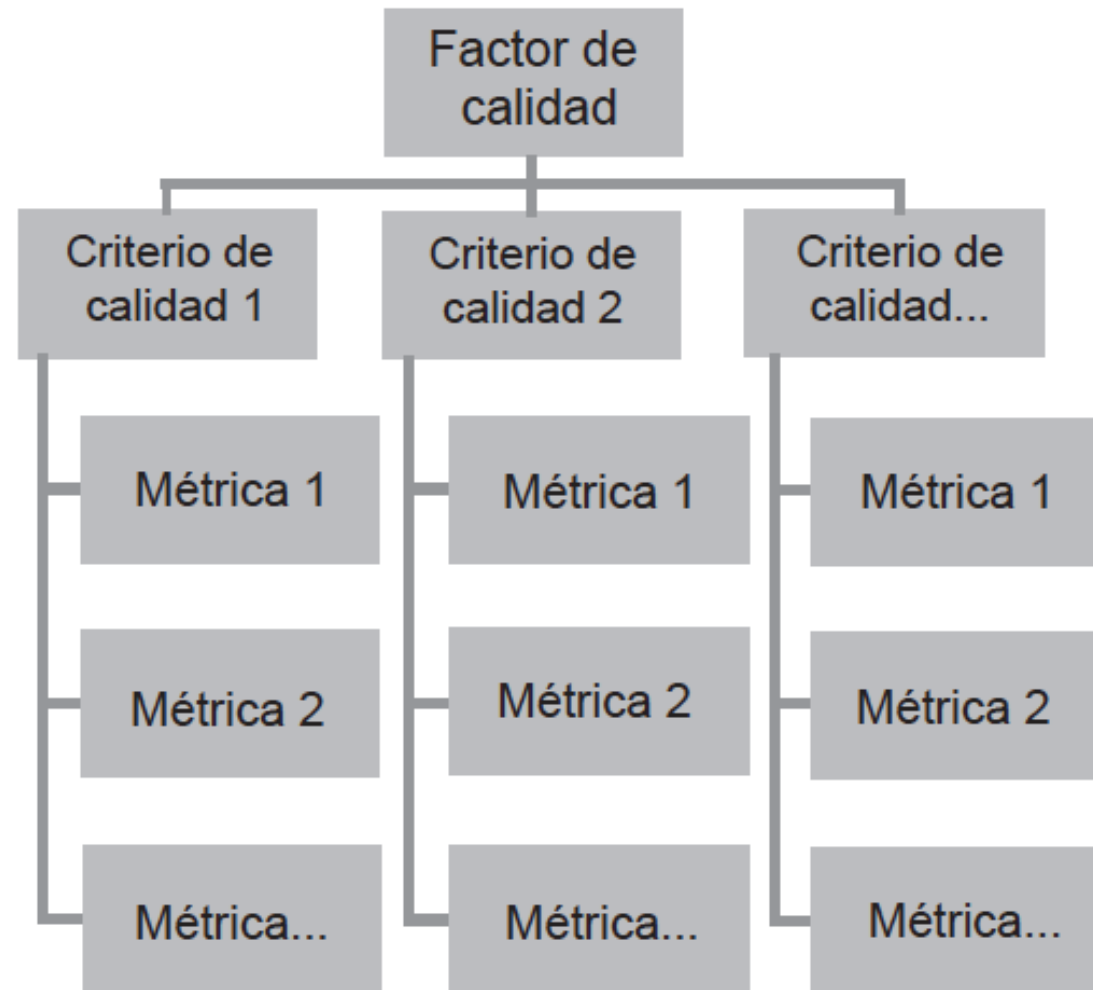




Estamos
construyendo el
producto correcto?

Estamos construyendo el
producto correctamente?





- Según Fenton, desarrollar una métrica única es semejante a la búsqueda imposible del *santo grial*.
- La medición es necesaria si se desea conseguir calidad.
- Las métricas deben ser:
 - Simples y fáciles de usar
 - Empírica e intuitivas
 - Consistentes y objetivas



■ *Mantenibilidad*

ISO 9126

- Analizabilidad: Capacidad del producto software de diagnosticar sus deficiencias o causas de fallos, o de identificar las partes que deben ser modificadas.
- Cambiabilidad: Capacidad del producto software de permitir implementar una modificación especificada previamente. La implementación incluye los cambios en el diseño, el código y la documentación. Si el software es modificado por el usuario final, entonces, la cambiabilidad puede afectar a la operabilidad.
- Estabilidad: Capacidad del producto software de minimizar los efectos inesperados de las modificaciones.
- Facilidad de prueba: Capacidad del producto software de permitir evaluar las partes modificadas.
- Conformidad: Capacidad del producto software de satisfacer los estándares o convenciones relativas con la mantenibilidad.



- La complejidad y la mantenibilidad son las que tienen una relación más directa sobre los costes del mantenimiento: un **aumento en la complejidad implica una mayor dificultad de mantenimiento**, es decir, una reducción de la mantenibilidad.
- Efectos sobre la complejidad:
 - Durante las actividades de mantenimiento, la complejidad del software tiende a aumentar. Si este fenómeno no se corrige, se llegará a una situación en la que un eventual cambio será impracticable porque requiere más esfuerzo que el disponible para implementarlo.
 - La **reducción de la complejidad del software redunda en un aumento de su fiabilidad y mantenibilidad**.



- Efectos sobre la mantenibilidad: Algunos cambios en el software pueden reducir la mantenibilidad. Los que producen este efecto con más asiduidad son:
 - Violar los estándares de codificación
 - Reducir la cohesión
 - Incrementar el acoplamiento
 - Incrementar la complejidad esencial

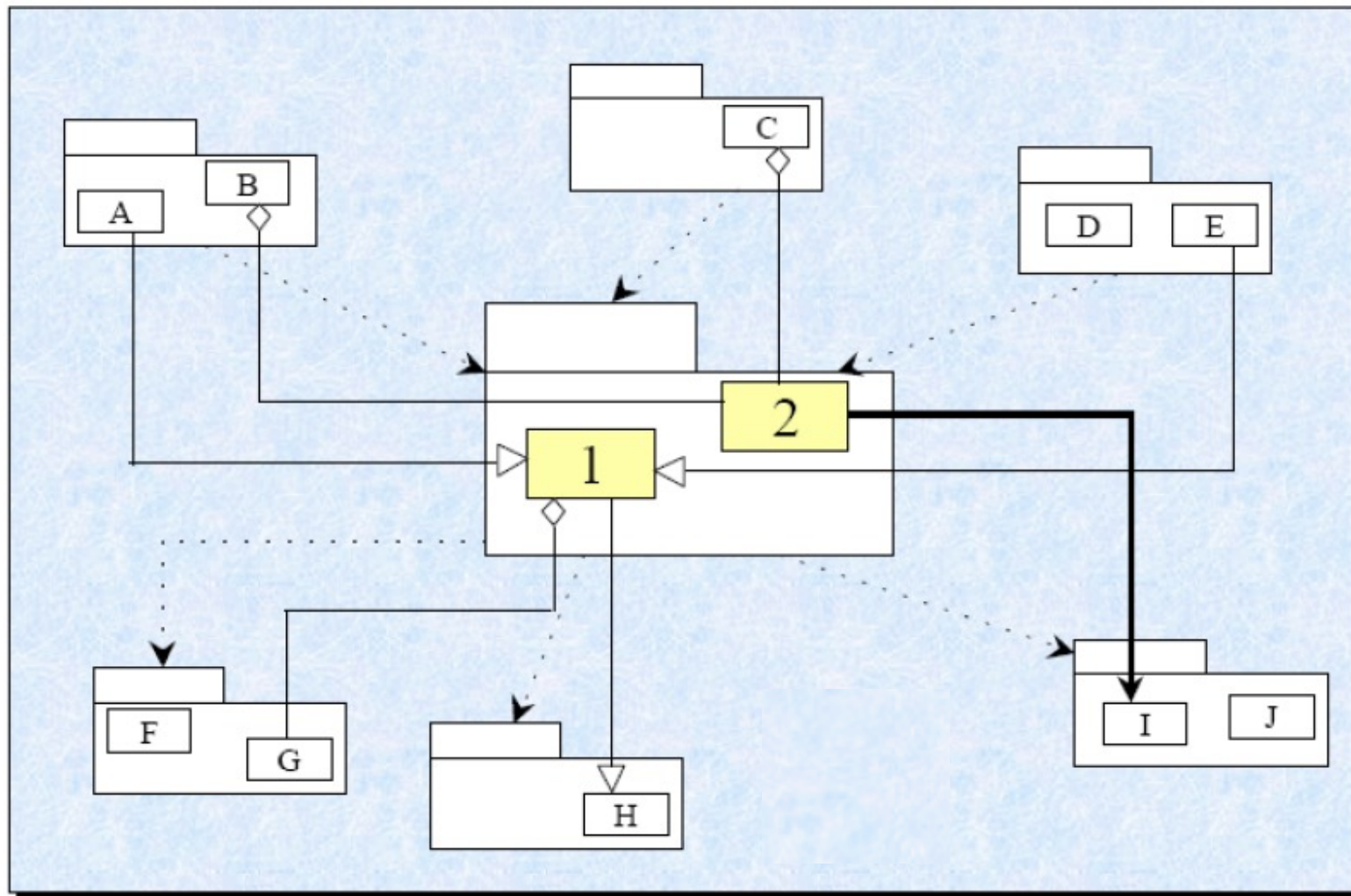


- ¿Cómo medir la mantenibilidad?
- La respuesta a esta pregunta se encuentra en la estrecha relación que existe entre los conceptos de calidad del software y de mantenibilidad. Por esta razón, los estándares ISO e IEEE proponen métricas de calidad para medir la mantenibilidad del software.



- *Métricas de calidad*
 - Acoplamiento
 - Cohesión
 - Complejidad Ciclomática de McCabe
 - Código Churn
 - Code Coverage
 - Código Muerto
 - Duplicación de Código





- Las especificaciones y requisitos normalmente son:
 - *incompletas*
 - *inconsistentes*
- *Puede darse tensión entre:*
 - Requisitos del cliente (efficiency, reliability, etc.)
 - Requisitos del desarrollador (maintainability, reusability, etc.)
- No todos los requisitos se pueden medir/evaluar directamente:
 - Fácil de medir con métricas, por ejemplo (e.g., errors/KLOC),
 - Difícil de medir y solo con métricas indirectas (e.g., usability).

