



Universidad Nacional de Tucumán



Universidad Nacional de Tucumán
Facultad de Ciencias Exactas y Tecnología

Ingeniería de Software II

Gestión de calidad del Software

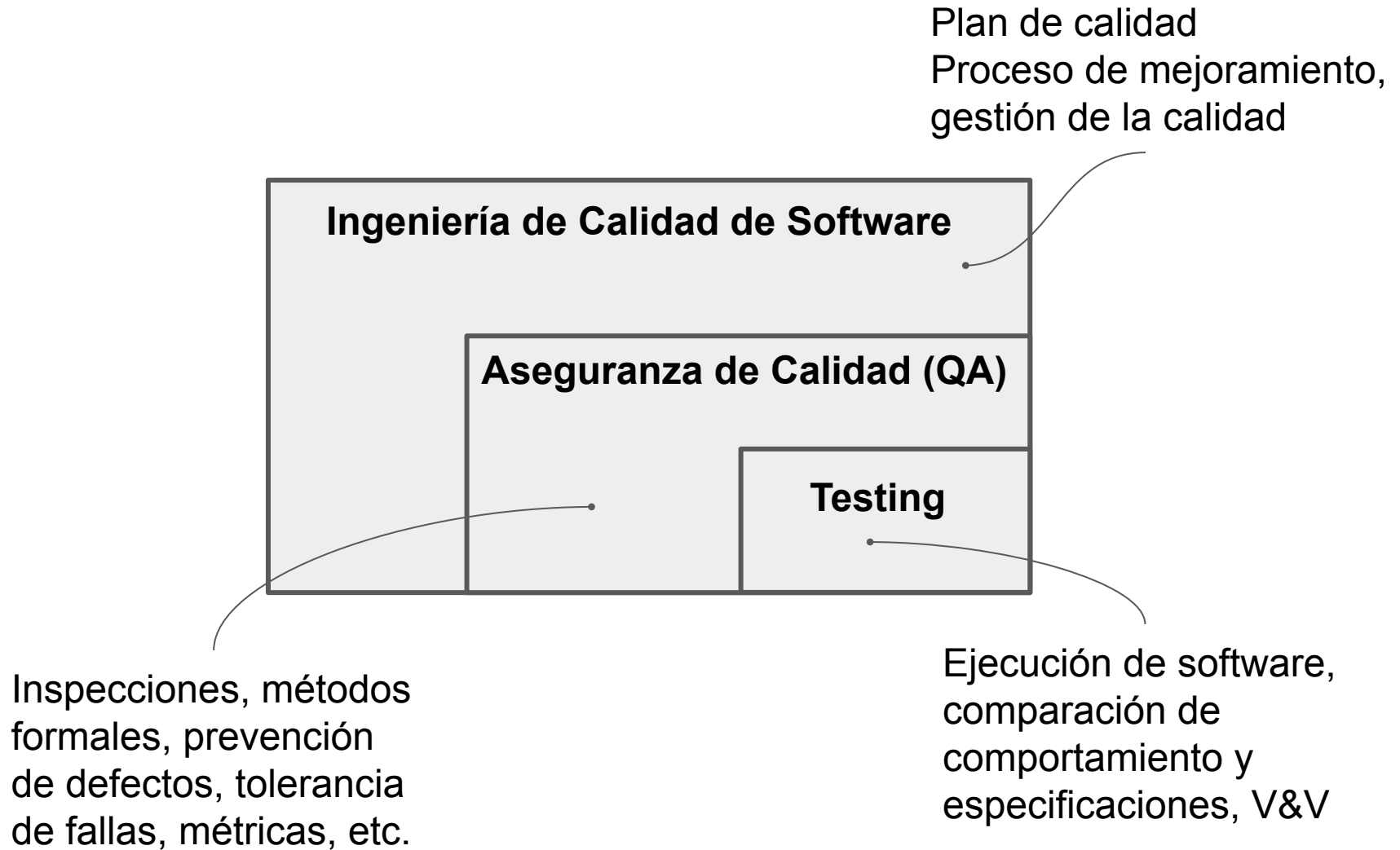
Mg. Héctor A. Valdecantos

Introducción - Historia y Contexto

- Industria manufacturera
 - Concepto de calidad - revolucion industrial
- Hacia una industria de servicios
 - De cero defectos a cero defección
- 1960 Primeros sistemas de software de gran scala
 - Misión Apolo - [Margaret Hamilton](#)
- 1968 NATO conference - Crisis del software
 - Mythical man month (1975)
- Sistemas de calidad para software
 - TQM, CMM, ISO, CMMI



Concepto de Calidad del Software - Alcance



Proceso de calidad

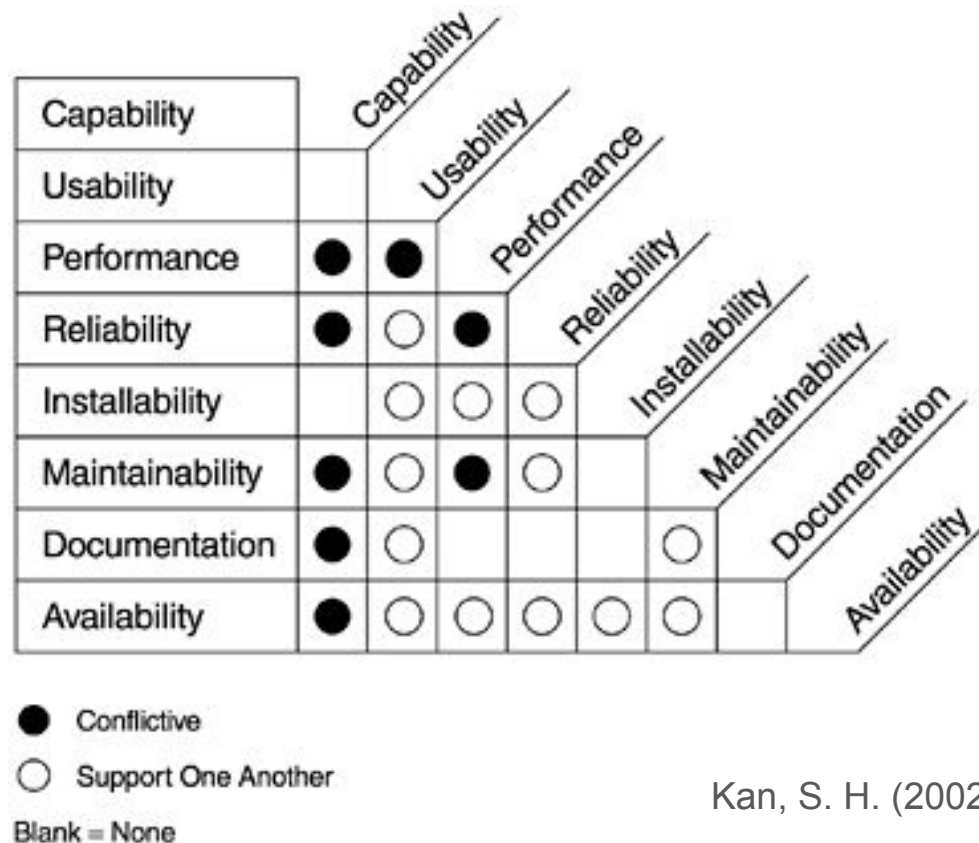
Nos permite establecer y monitorear si las expectativas de calidad del cliente son cubiertas.

Son las expectativas de calidad cubiertas y en qué grado?

- Planificación
- Ejecución de actividades QA
- Mediciones y análisis

Concepto de Calidad del SW - Atributos de calidad

- Requerimientos funcionales
- Requerimientos no-funcionales



Kan, S. H. (2002).

Concepto de Calidad del Software

“Lo sé cuando lo veo!”

[visión popular: anónimo]

“Si no lo puedes medir, probablemente no lo entiendas, y seguramente no puedes controlarlo”

[visión profesional: Lord Kelvin, Scottish Physicist – 1884]

Concepto de Calidad del Software

El concepto de calidad en el software es objetivo y subjetivo a la vez:

“Conformidad de requerimientos” (Crosby 1979)

“Aptitud para el uso” (Fitness for use) (Juran y Gryna 1970)

Concepto de Calidad del Software

“La noción de calidad es que el producto desarrollado **cumple su especificación**” (Crosby 1979)

“la **concordancia con los requisitos** funcionales y de rendimiento establecidos con los estándares de desarrollo debidamente documentados y con las **características implícitas** que se espera de todo Software desarrollado profesionalmente” (Pressman 1992)

“El conjunto de características de una entidad que le confieren su **aptitud para satisfacer las necesidades expresadas y las implícitas**” (ISO 8402, UNE 66-001-92)

¿Qué es calidad? Diferentes perspectivas

1. Trascendental:

- Muy difícil de definir calidad en términos abstractos. Puede ser reconocida si está presente aunque esté asociada con atributos algunas veces intangibles.

2. Del usuario

- Aptitud para el uso o satisfacción de las necesidades del usuario.

3. Basada en la manufacturación

- Conformancia al estándar de proceso.

4. Vista de producto

- Controlar los indicadores internos de calidad.

5. Vista basada en el valor

- Disponibilidad a pagar por el software

[de Garvin 1984, como se citó en “SW quality: the elusive target”, Kitchenham & Pfleeger, 1996]

Problemas para definir calidad en el SW

1. En la especificación de requerimientos
 - Según el cliente o usuario
 - Según la organización desarrolladora (algunas veces no se incluyen)
2. No se sabe cómo especificar características de calidad o atributos de calidad de una forma no ambigua
3. Dificultad para especificar el software.
 - Un producto software puede estar en concordancia con su especificación y los usuarios considerarlo de baja calidad

Problemas para conseguir calidad en el SW

- Especificaciones imperfectas
 - La calidad como “conformidad de especificaciones” no puede ser completa.
- Los atributos de calidad generalmente no se especifican explícitamente [Len Bass]
 - Mantenibilidad, portabilidad, eficiencia, etc
 - Atributos como intangibles y subjetivos
- Especificar los requerimientos de software no es suficiente:
 - Se necesita una “Cultura de calidad”

Escenario para un atributo de calidad

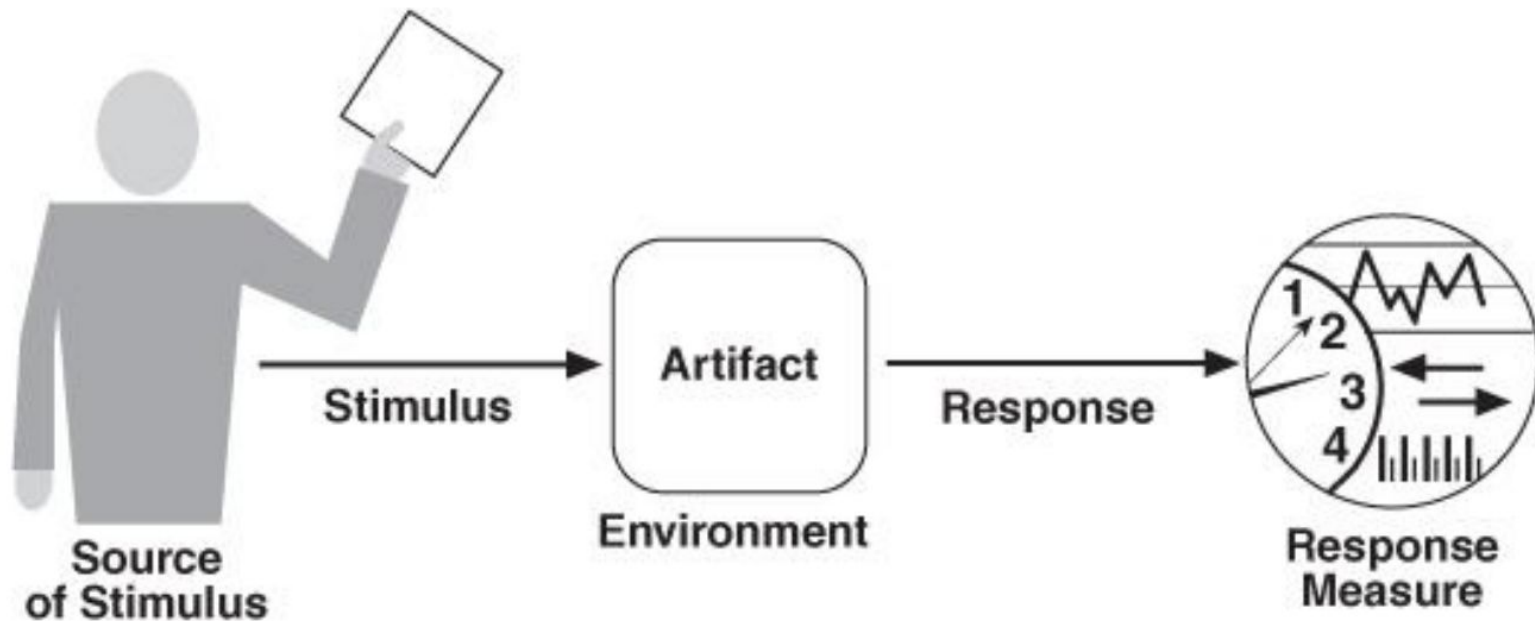


Figure 4.1. The parts of a quality attribute scenario

Escenario para un atributo de calidad: disponibilidad

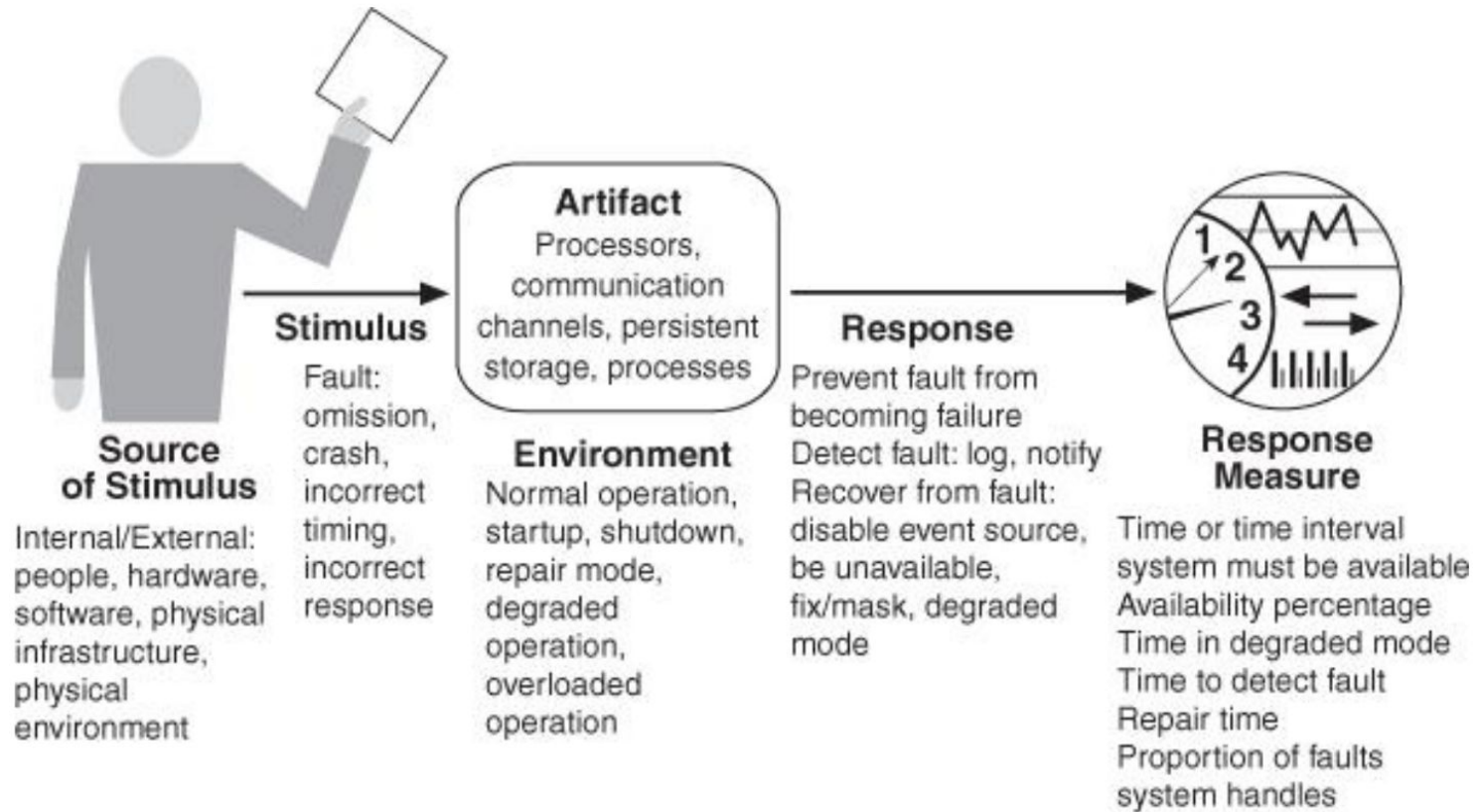


Figure 4.2. A general scenario for availability

Cómo asegurar la calidad

- Procurar proceso de desarrollo adecuado
- Especificar claramente los requerimiento
 - Funcionales, no funcionales, restricciones
- Aplicar métricas del proceso, producto, y proyecto
 - permite adaptar y mejorar el proceso
 - Establecer líneas bases para la estimación y comparación
- Establecer un marco o cultura de calidad
 - no sólo estándares y procedimientos



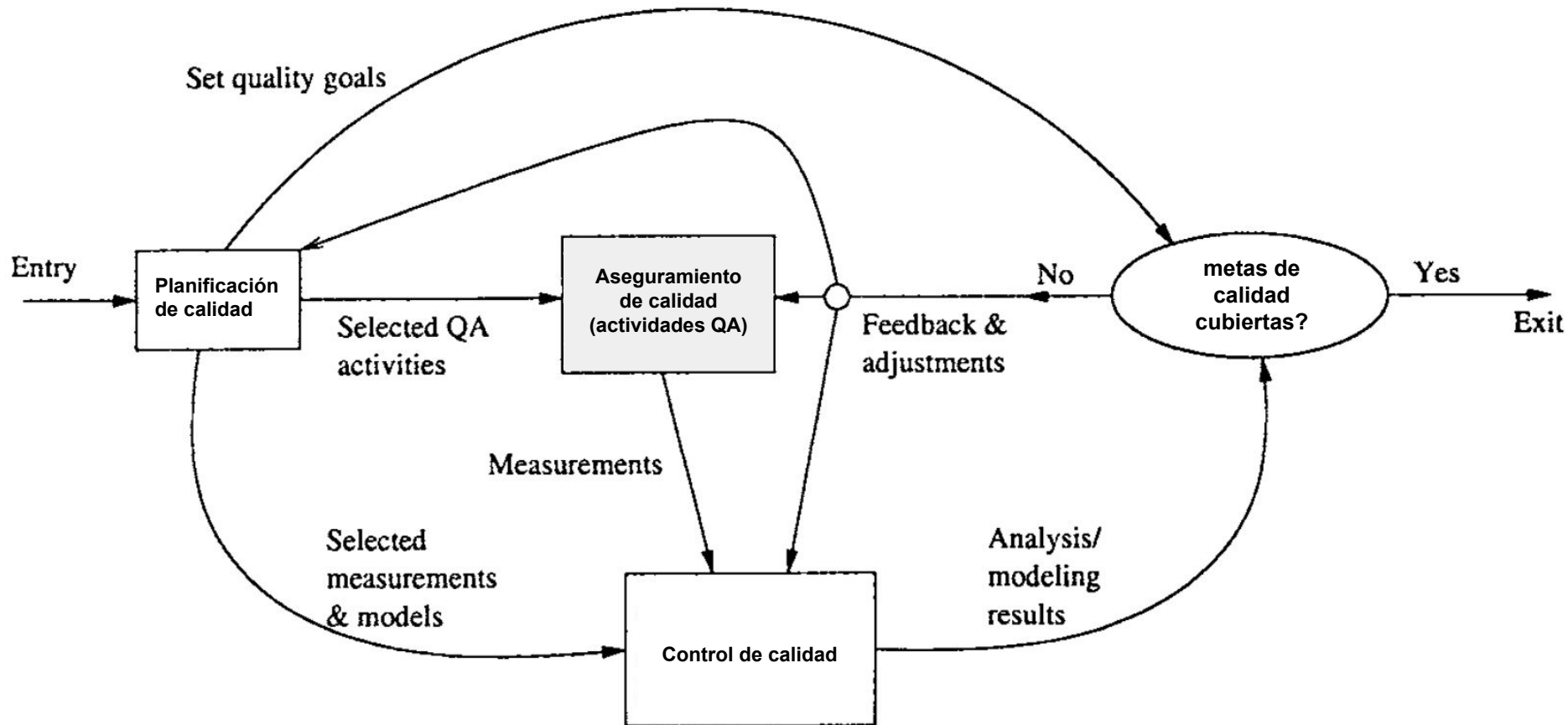
“the quality of a product is highly influenced by the quality of the processes used to produce it” [Humphrey 1989].

Proceso de gestión de calidad

Incluye tres actividades principales

- Planificación de calidad (pre QA)
 - Se especifican las metas de calidad, y la estrategia general seleccionando un marco de trabajo (actividades, medidas).
- Aseguramiento de calidad (in QA)
 - Es la ejecución de las actividades, aplican métricas, y se administran los defectos.
 - QA consume mayor recurso aplicado en calidad
- Control de calidad (post QA)
 - se proporcionan una evaluación de la calidad y se obtiene feedback.

Proceso de gestión de calidad



[Jeff Tian “software quality engineering”, pag. 54].

Aseguramiento de calidad - QA

Aporta la confianza en que el producto o servicio satisfará los requisitos dados de calidad. Se comienza con el aseguramiento de calidad antes del desarrollo. Se asegura la calidad cuando:

- Se definen procesos, estándares y marcos de calidad.
- Se seleccionan métodos y herramientas de análisis, diseño, programación, y prueba.
- Se realizan Inspecciones durante el proceso de desarrollo
- Se definen estrategias de pruebas en todas las escalas
- Se gestiona el control de cambios:
 - Documentación, diseño, código, pruebas, conf, etc
- Se definen mecanismos de medidas (def. operacionales de las métricas)
- Se realiza verificación y validación
- Se registran auditorías e informes

Aseguramiento de calidad - QA

Las actividades de QA deberían ser realizadas por un grupo independiente del equipo de desarrollo para tener una visión objetiva y evitar conflictos de intereses.

Las actividades incluyen:

- Implementación de un sistema de calidad adecuado
- Implementación de métricas para el proceso, proyecto y producto
- V&V durante el proceso
 - Pruebas e inspecciones
- Gestión de configuración del software

Un esquema para las actividades QA

Esquema de actividades QA basadas en cómo tratar con los defectos. Jeff Tian denota que QA es “lidar con defectos”

- **Prevención de defectos**
 - Capacitación del equipo
 - Métodos formales, estándares, sistemas de calidad
 - Técnicas de prevención
- **Reducción de defectos**
 - Inspecciones: Detección y eliminación directa de faltas
 - Testing: observación de fallas y remoción de faltas
 - Otras técnicas e identificación de riesgo
- **Contención de defectos**
 - Tolerancia a fallas del Software
 - Garantía de seguridad y contención de fallas

[Tian, Jeff. Software quality engineering: testing, quality assurance, and quantifiable improvement. John Wiley & Sons, pag 28, 2005.]

Un servicio necesario para mi sistema deja de responder y el sistema se cae ¿Qué debería plantear el equipo de QA?

- Prevención de defectos
- Reducción de defectos
- Contención de defectos

Estándares del Software [Sommerville, 2011]

Una parte importante de QA es la definición, selección y adaptación de estándares a aplicar en el proceso o producto: Los estándares son importante por las siguientes razones:

- Capturan conocimiento de valor para la organización sobre:
 - Las mejores y más apropiadas prácticas para la compañía. Se necesita experiencia (prueba y error) para obtener este conocimiento que incluye los estándares. (reuso de exper.)
- Definen “calidad” en un ambiente particular
 - Brindan un marco de trabajo en el cual se implementa el proceso de mejora de calidad
- Ayudan a la continuidad del trabajo
 - Un estándar asegura que todos siguen las mismas prácticas, lo que reduce el esfuerzo de aprendizaje para comenzar un trabajo nuevo

Estándares del Software [Sommerville, 2011]

Dos tipos de estándares usados en la gestión de calidad:

- **Estándares de producto:** aplican al producto siendo desarrollado
 - de documentos: estructura de doc. de requerimientos, otros
 - de documentación: encabezado de comentario de clases, etc
 - de codificación: pautas de cómo usar un lenguaje, code conventions
- **Estándares de proceso:** definen procesos a seguir en el desarrollo
 - Encapsulan buenas prácticas de desarrollo
 - Incluyen definiciones de proceso de especificaciones, de diseño y procesos de validación, procesos de soporte de herramientas, etc
 - Definen una descripción de los documentos que deben ser producidos en estos procesos.

Estándares del Software

Ejemplos de estándares de productos y procesos

Estándares de producto	Estándares de proceso
Formulario de revisión de diseño	Conducta de revisión de diseño
Estructura del doc. de requerimiento	Entrega de código nuevo para la construcción de un sistema
Formato del encabezado de método	Proceso de lanzamiento de versión
Convención de código	Proceso de aprobación del plan de proyecto
Formato del plan de proyecto	Proceso de control de cambio
Formulario de pedido de cambio	Proceso de registro de pruebas

[Sommerville, “Software Engineering”, 9th edition, pag. 659, 2011]

Planificación de la Calidad

- Establece la calidad del software:
 - uptime o disponibilidad por año 99.99% Es admisible para un banco? para un sistema de control de vuelo?

$$\% \text{ disponibilidad} = \frac{\text{disponibilidad acordada} - \text{downtime}}{\text{disponibilidad acordada}}$$

- Describe cómo se evaluarán los atributos de calidad
 - Definiciones operacionales de métricas
- Qué quiere decir “alta calidad”?
 - Se establecen los criterios de calidad
- Selecciona o define los estándares a usar
 - Para el producto y proceso de desarrollo

Planificación de la Calidad

- Un plan de calidad generalmente incluye (Humphrey 1989):
 - Introducción del producto: descripción del producto, el mercado al que se dirige y expectativas de calidad
 - Planes del producto: deadlines, responsabilidades importantes, planes de distribución y servicio.
 - Descripciones del proceso: estándares a usar para gestionar y desarrollar el producto
 - Metas de calidad: identificación y justificación de *atributos de calidad* críticos
 - Riesgos y gestión de riesgo: los riesgos que pueden afectar la calidad del producto y las acciones a tomar para abordar estos riesgos

Algunos Atributos de Calidad

accessibility
accountability
accuracy
adaptability
administrability
affordability
agility
auditability
autonomy
availability
compatibility
composability
configurability
correctness
credibility
customizability
debugability
degradability
determinability
demonstrability
dependability
deployability
discoverability
distributability
durability
effectiveness
efficiency

evolvability
extensibility
failure transparency
fidelity
flexibility
inspectability
installability
integrity
interchangeability
interoperability
learnability
maintainability
manageability
mobility
modifiability
modularity
operability
orthogonality
performance
portability
precision
predictability
process capabilities
producibility
provability
recoverability
relevance
reliability

repeatability
reproducibility
resilience
responsiveness
reusability
robustness
safety
scalability
seamlessness
self-sustainability
serviceability
security -> securability
simplicity
stability
standards compliance
survivability
sustainability
tailorability
testability
traceability
timeliness
traceability
transparency
ubiquity
understandability
upgradability
usability

Sistemas de calidad para software

- Una organización utiliza sistemas de calidad para controlar y mejorar la efectividad de los procesos utilizados para entregar un producto o servicio de calidad
- Un Sistema de Calidad es un conjunto de prácticas y procesos formales e informales que se centran en:
 - Necesidades del consumidor; Visión de liderazgo; Participación de los trabajadores; Mejora continua; Toma de decisiones informadas basadas en datos en tiempo real
- Sistemas de calidad estandarizados conocidos:
 - TQM, [CMMI](#), ISO, six sigma
 - Establecen un marco de actividades para la calidad
- Certificaciones de calidad
 - Trata de ser una valoración objetiva e independiente de los que aplican un sistema de calidad

Control de calidad

Son las técnicas y actividades de carácter operativo, utilizadas para satisfacer los requisitos relativos a la calidad, centrada en dos objetivos fundamentales:

- Mantener bajo control el proceso
- Eliminar las causas de los defectos en las diferentes fases del ciclo de vida

Los procesos de control de calidad tienden a ser directos y fácilmente comprensibles. (Costo de la calidad)

Nota: Ian Sommerville trata de evitar el término “Control de calidad” porque lo considera incluido en QA. QC es muy usado en la industria manufacturera

Control de calidad - Tipos de revisiones de calidad:

- **Inspecciones de diseño o programas**

- De formales a informales
- Inspecciones de requerimientos, de diseño, de programas
- Trata de encontrar defectos lo antes posible en el proceso de desarrollo (costo del defecto)

- **Revisiones de progreso**

- Auditan el proceso y el producto
- Progreso actual comparado con el progreso planeado
- Su objetivo es el costo, planes y calendario: entregar software en tiempo y forma

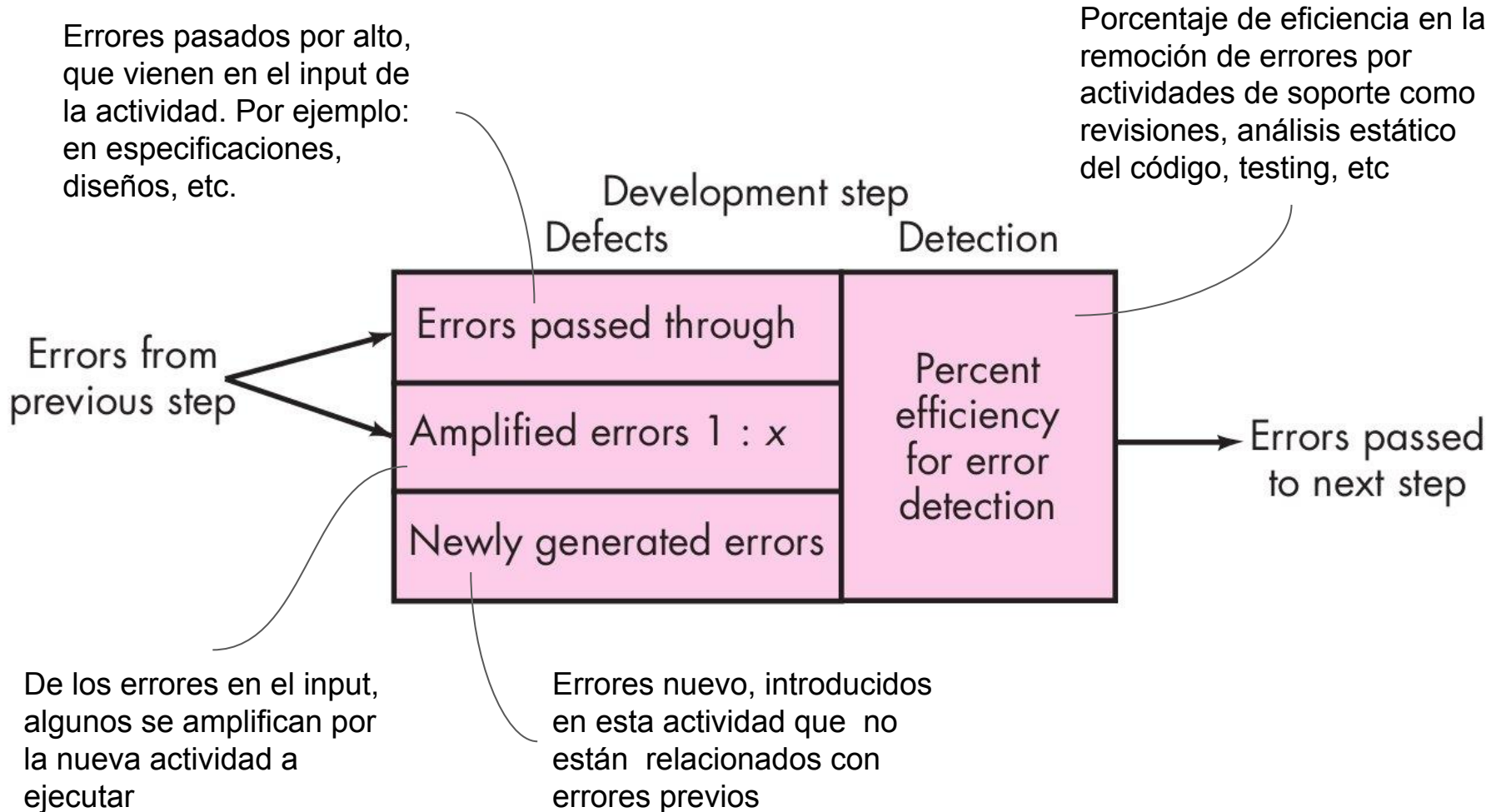
- **Revisiones de calidad**

- Chequean conformancia con los estándares, inconsistencias entre documentos,
- El objetivo es mejorar la calidad del software, no evaluar la performance de las personas

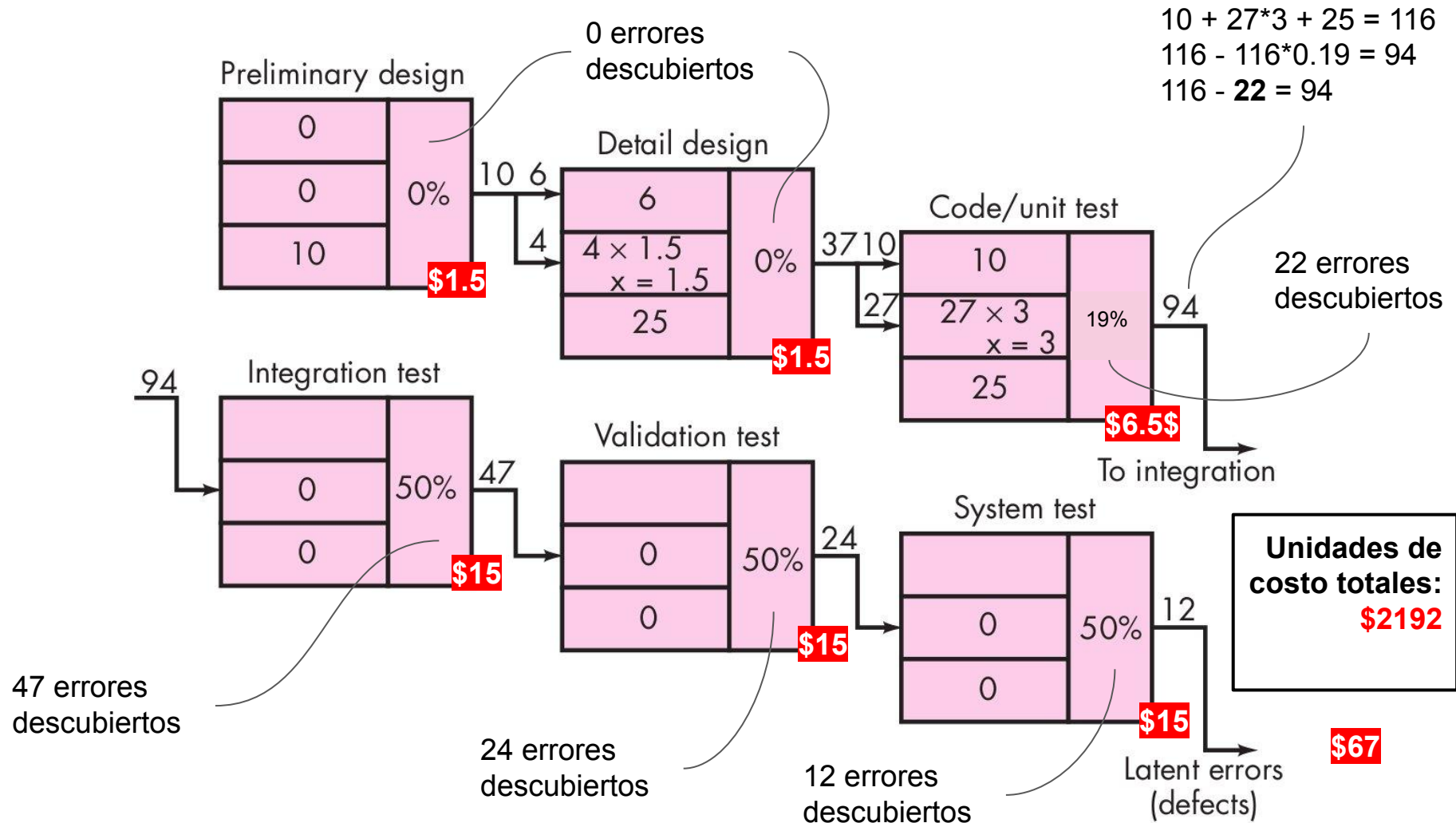
Amplificación de defectos

- Durante una acción de ingeniería de software
 - Introducción de errores
 - Trabajos sobre errores preexistentes
 - Factor de amplificación
- Revisiones
 - Pueden pasar por alto errores
 - Eficiencia de Remoción de Errores (DRE)
- Costo de los defecto
 - Más avanzado el desarrollo, más caro el defecto.

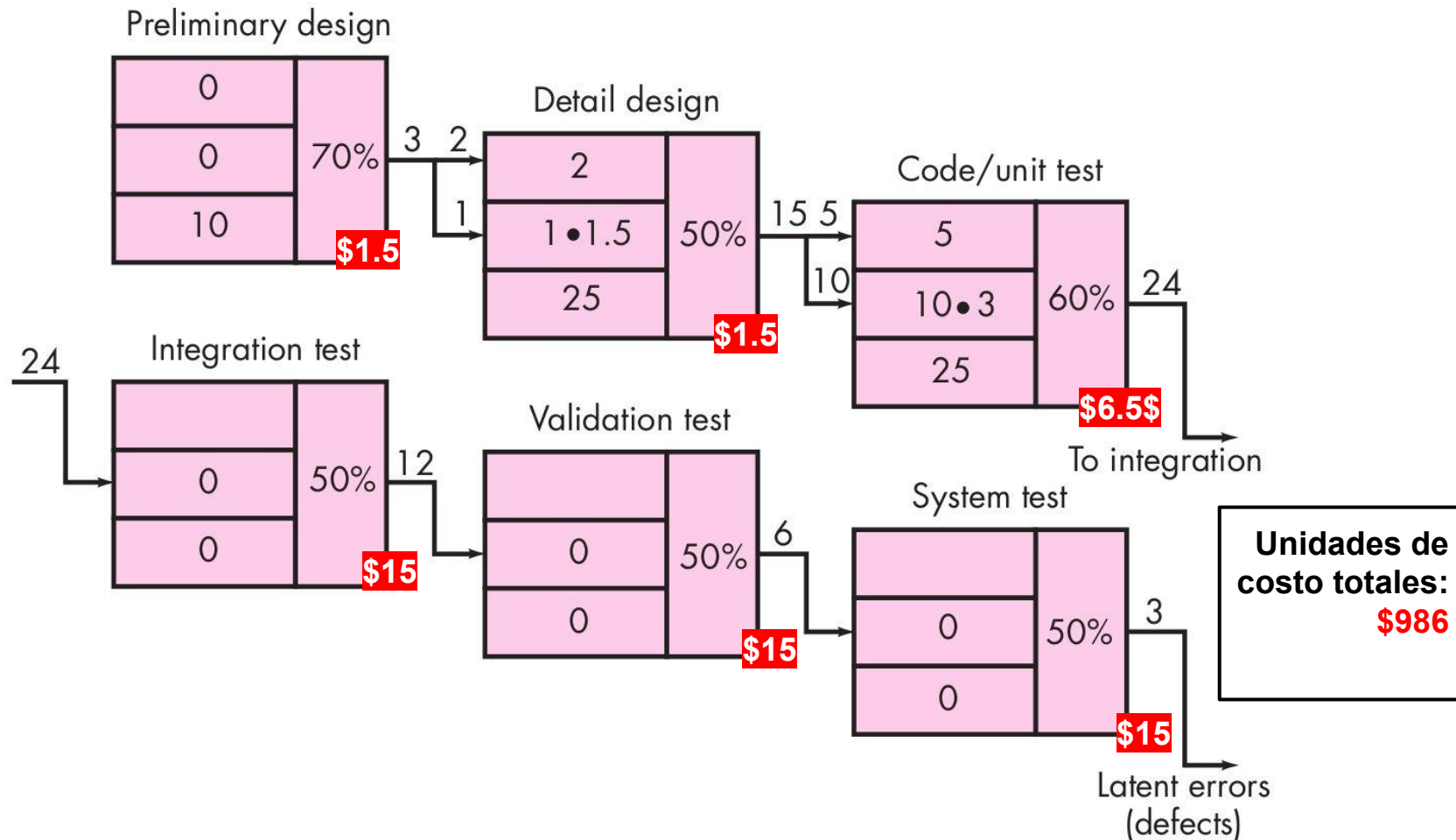
Control de calidad - Propagación de defectos



Control de calidad - Propagación de defectos

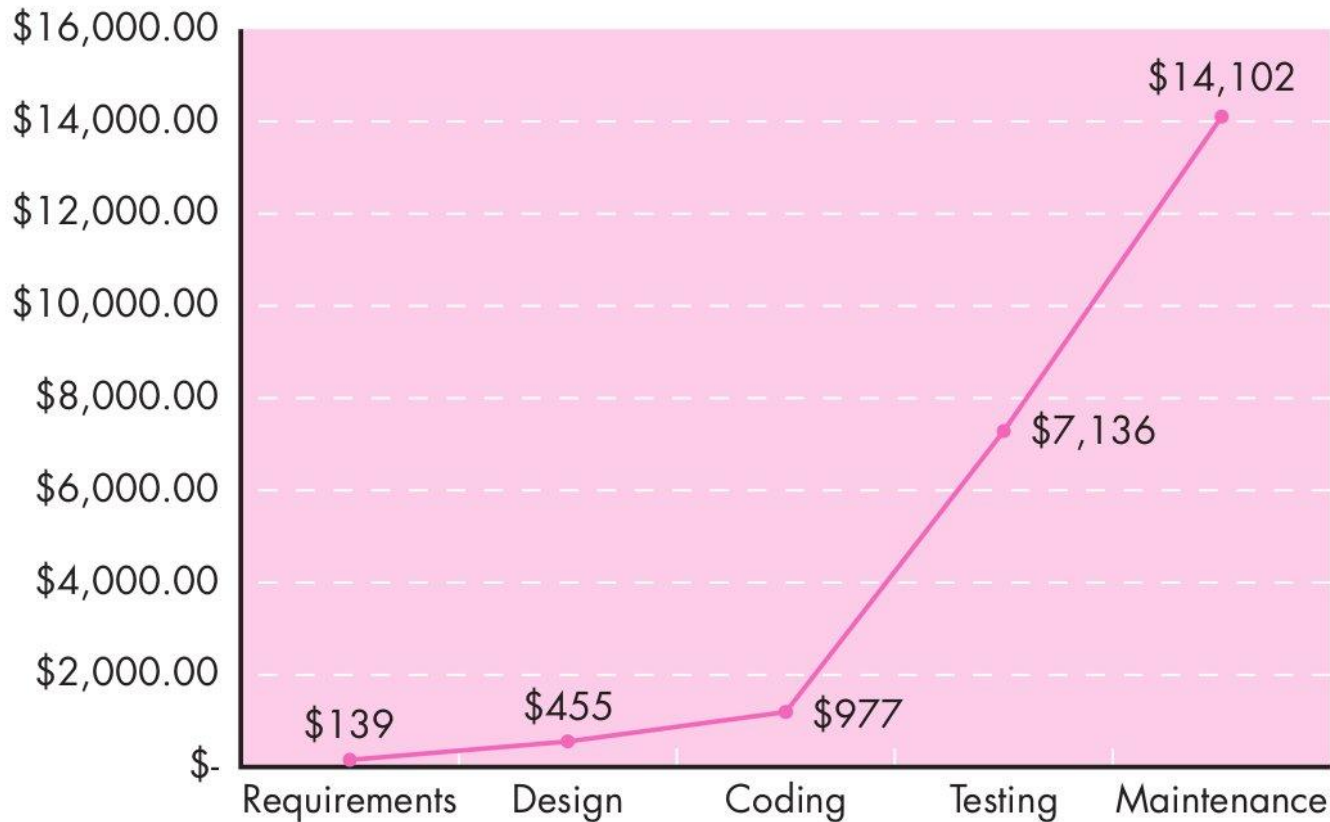


Control de calidad - Propagación de defectos



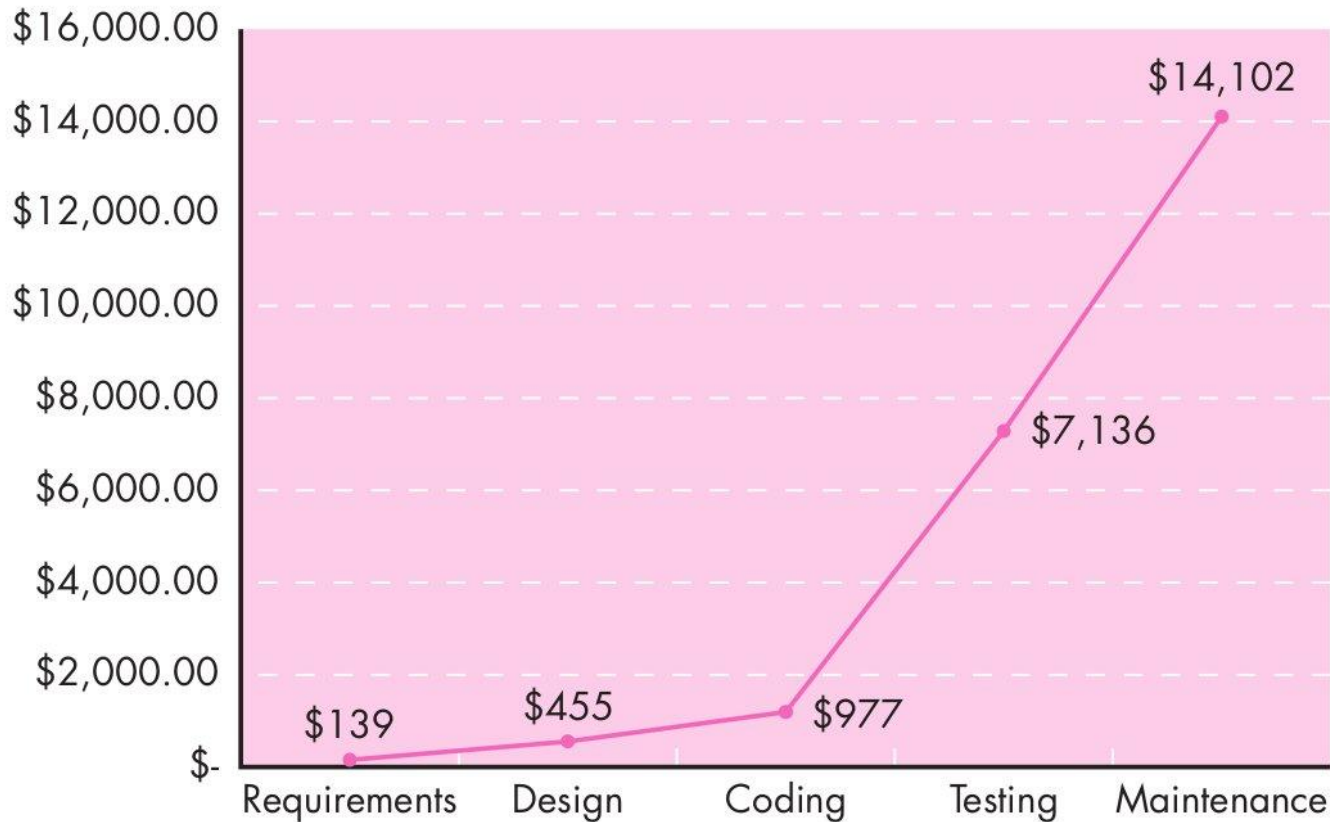
[Pressman, Roger S. Software engineering: a practitioner's approach. 7th ed, (2010), p. 419]

Control de calidad - Costo de remoción de defectos



[Pressman, "Software engineering: a practitioner's approach", 7th ed, (2010), p. 409]

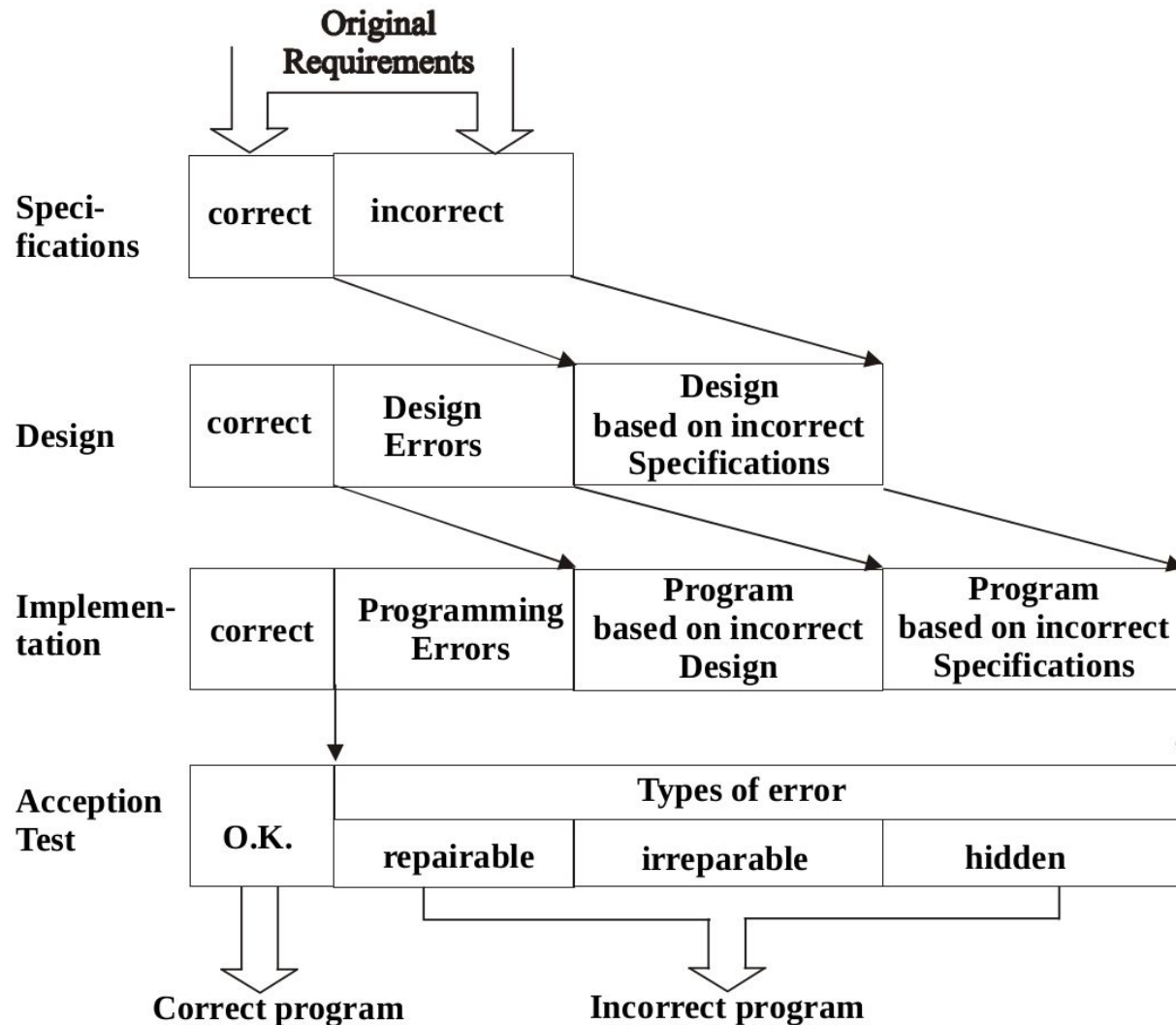
Control de calidad - Costo de remoción de defectos



[caso 1] $(0 * \$139) + (0 * \$445) + (22 * \$997) + (83 * \$7136) + (12 * \$14102) = \$783,446$

[caso 2] $(7 * \$139) + (15 * \$445) + (36 * \$997) + (21 * \$7136) + (3 * \$14102) = \$235,702$

Control de calidad - Origenes de diferentes tipos de errores



[Axel Hunger, Fundamentals of software engineering I]

Métricas de calidad del Software

Se puede medir a lo largo del proceso de ingeniería de SW y una vez que ha sido puesto en producción

- Métricas obtenidas antes de entregar el software
 - Base cuantitativa para tomar decisiones
 - Complejidad del programa, modularidad efectiva, tamaño, etc
- Métricas obtenidas después de entregar el software
 - Nro de defectos no descubiertos durante testing
 - Facilidad de mantenimiento

Métricas de calidad del Software - Factores (

contenido
opcional

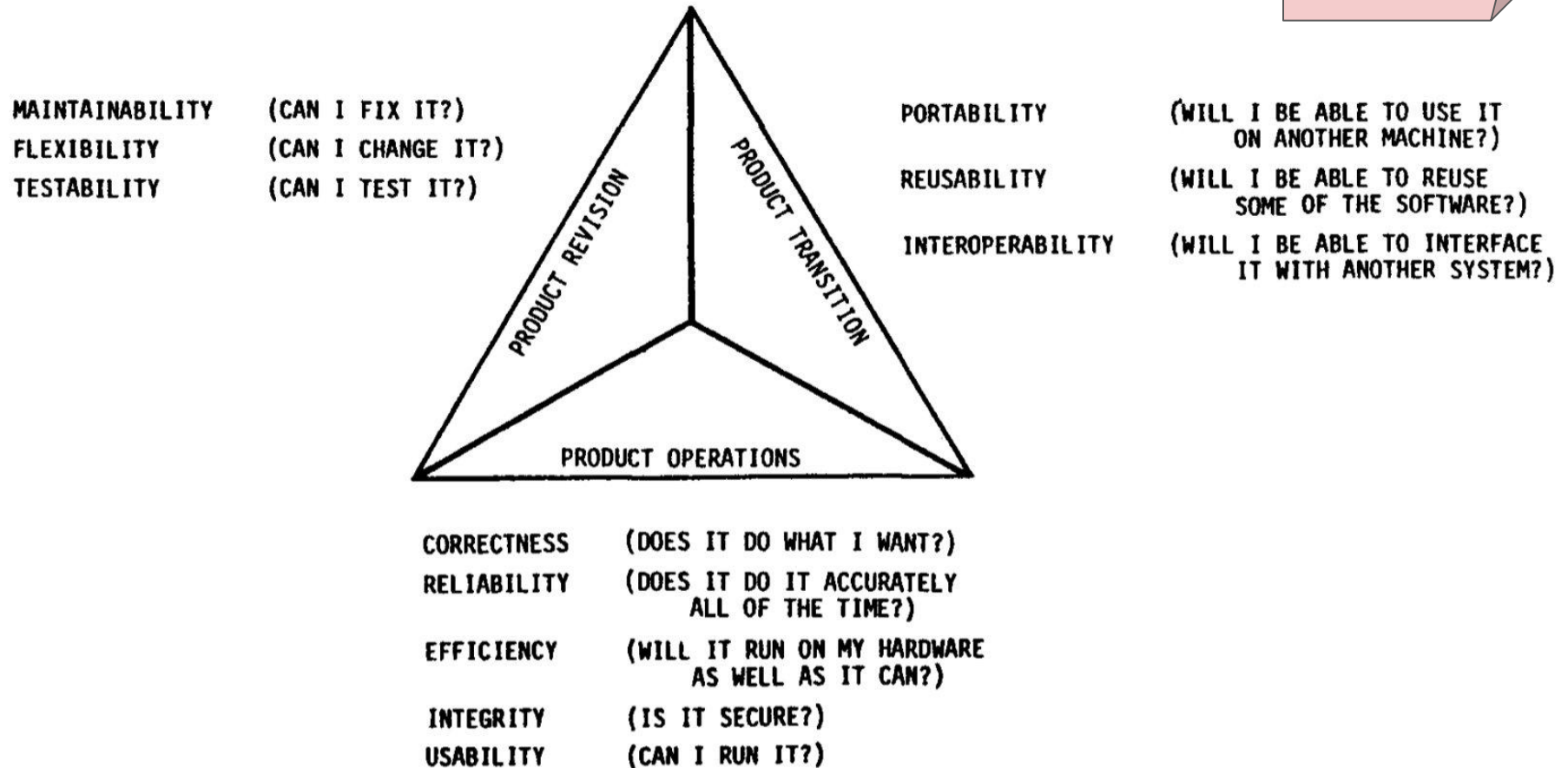
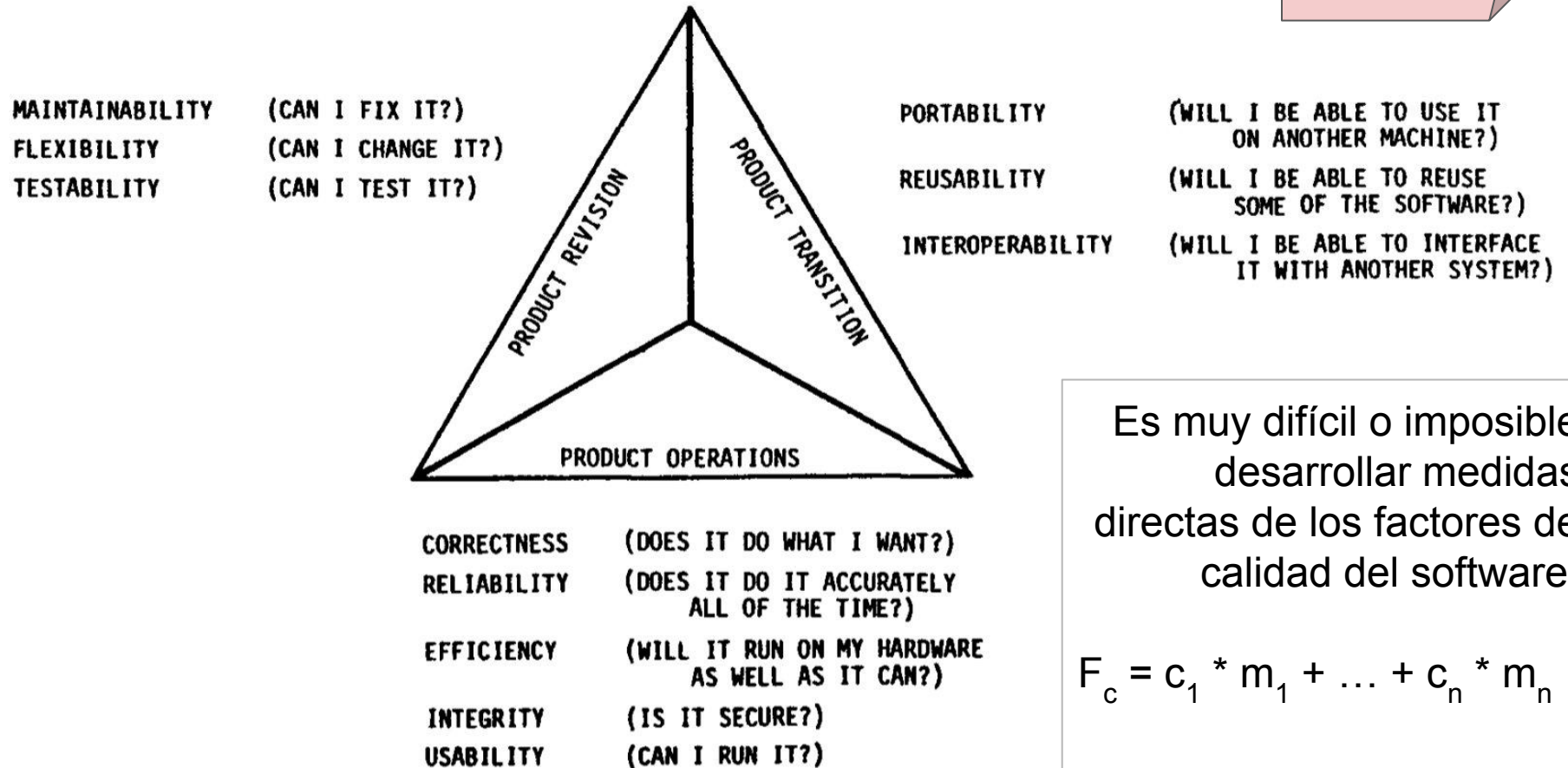


Figure 2. Software Quality Factors

[Cavano, McCall. "A framework for the measurement of software quality." ACM SIGMETRICS Performance Evaluation Review. Vol. 7. No. 3-4. ACM, 1978]

Métricas de calidad del Software - Factores (

contenido
opcional



Es muy difícil o imposible desarrollar medidas directas de los factores de calidad del software:

$$F_c = c_1 * m_1 + ... + c_n * m_n$$

Figure 2. Software Quality Factors

[Cavano, McCall. "A framework for the measurement of software quality." ACM SIGMETRICS Performance Evaluation Review. Vol. 7. No. 3-4. ACM, 1978]

Algunas métricas de calidad del Software

- Correctitud:

$$\text{densidad de defectos} = \frac{\text{Num. de defectos}}{KLOC}$$

- Mantenibilidad:

$$TMEC = \frac{\sum_1^n (TEC_i)}{n}$$

- Integridad:

$$\text{Integridad} = \sum_1^n [1 - \text{amenaza}_i * (1 - \text{seguridad}_i)]$$

- Usabilidad

- Software Usability Measurement Inventory (SUMI)

Integración de las métricas al proceso de IS

Si no medimos, no hay un modo de determinar si estamos mejorando. Si no mejoramos, estamos en una situación desfavorable que nos llevará a la obsolescencia.

- Las métricas proporcionan beneficios en
 - el proceso en un nivel estratégico para controlar y entender características del producto.
 - el producto en un nivel técnico para mejorar el desarrollo y el mantenimiento
 - el proyecto en un nivel tácticos para administrar los recursos en cuanto a la productividad según objetivos
- Cultura de calidad en los equipos de desarrollo
 - Establecer un marco base del cual podamos evaluar las mejoras

Establecimiento de una línea base

Una “baseline” consiste en datos recogidos de proyectos de software anteriores que son útiles para estimar. Deben poseer los siguientes atributos:

1. Los datos deben ser razonablemente precisos
2. Los datos deberían obtenerse de la mayor cantidad de proyectos posibles
3. Las medidas deben ser consistentes, deben tener la misma definición operacional entre proyectos.
4. Las aplicaciones de “baselines” mejoran cuando las usamos para estimar proyectos similares.

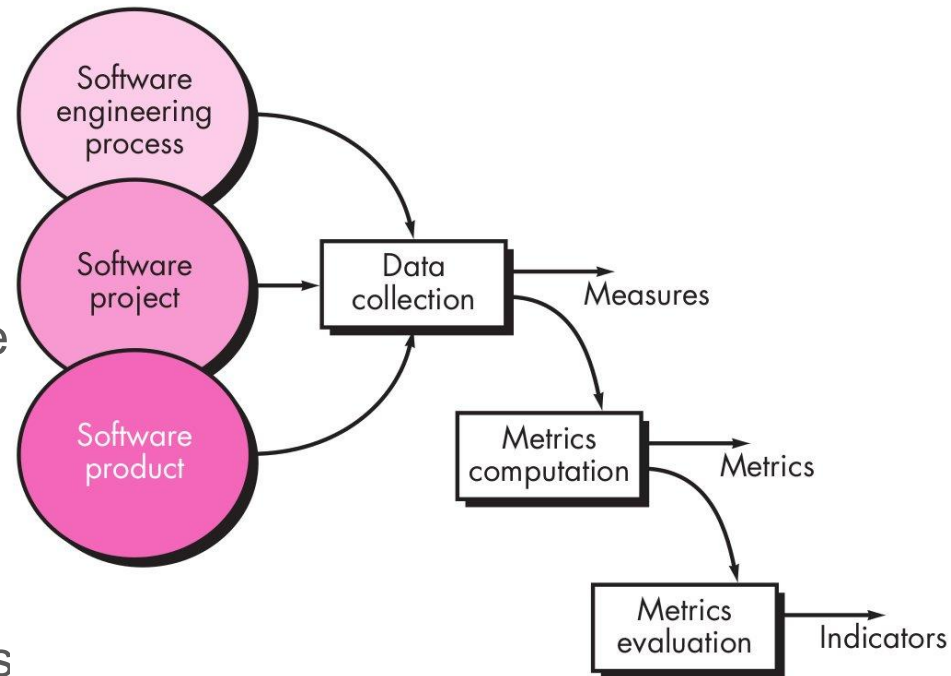
Recolección, cálculo y evaluación de métricas

El proceso de obtener baselines

Idealmente los datos necesarios han sido obtenidos de forma directa. Sino:

- Reconstrucción de datos: requiere de una investigación histórica de proyectos pasados

Luego se calculan las métricas y finalmente se evalúan para obtener las baselines.



[Pressman, Ingeniería de software, un enfoque práctico. 7th edición, p. 586]

Bibliografía

Cavano, McCall. "A framework for the measurement of software quality." ACM SIGMETRICS Performance Evaluation Review. Vol. 7. No. 3-4. ACM, 1978]

Tian, Jeff. Software quality engineering: testing, quality assurance, and quantifiable improvement. John Wiley & Sons, 2005

Pressman, Roger S. Software engineering: a practitioner's approach. 7th edition. Palgrave Macmillan, (2010).

Ian Sommerville. Software Engineering (9th ed.). Addison-Wesley Publishing Company, USA, (2010).

Clements, Paul, Rick Kazman, and Len Bass. "Software Architecture in Practice." (2013).