

A close-up photograph of a laboratory microplate with a glass pipette dispensing a yellow liquid into one of the wells. The background is slightly blurred, showing other wells and the pipette's tip.

Calidad y Pruebas de Software

Universidad Autónoma de Coahuila

Facultad de Sistemas

Carlos Nassif Trejo García



Contacto

c-trejo@uadec.edu.mx

844 2058966



Redes

<https://github.com/CarlosTrejo2308>

<https://www.linkedin.com/in/carlosnassif/>

Evaluación

- Parcial 1: 25%
Parcial 2: 25%
- Proyecto Final: 25%
Examen Final: 25%

Cada parcial se evaluará de la siguiente manera:

- Asistencia y participación: 10%
Presentación: 15%
Tareas: 15%
Proyecto: 25%
Examen: 35%

Temario

- **U1. Conceptos Básicos de Calidad**

- Presentación
- Definición de Calidad
- Definición de calidad de software
- Quién define la calidad
- Importancia de la calidad
- La calidad y el mundo globalizado
- Calidad de vida
- Calidad total

- **U2. Aseguramiento de la calidad del software (SQA)**

- Relación de la Ingeniería del software con SQA. Definición y propósito del SQA.
- Problemas que resuelve la SQA.
- Calidad del software en el ciclo de vida del mismo.
- Roles y responsabilidades de los equipos de desarrollo.
- Habilidades y capacidades del personal del SQA.
- Actividades del SQA.
- Métodos y herramientas.

Temario

- **U3. Estándares de calidad aplicados al software.**

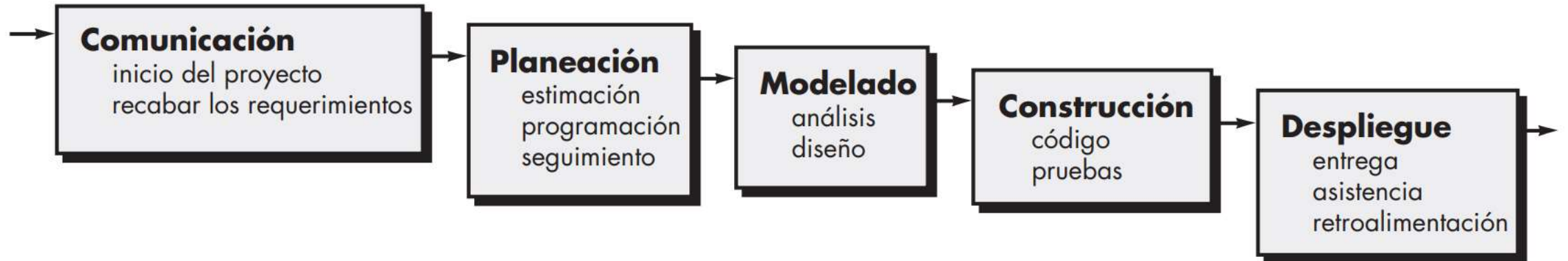
- ISO
- SPICE
- CMM
 - Definición del modelo.
 - Nivel inicial.
 - Nivel repetido.
 - Nivel definido.
 - Nivel administrado.
 - Nivel optimizado.

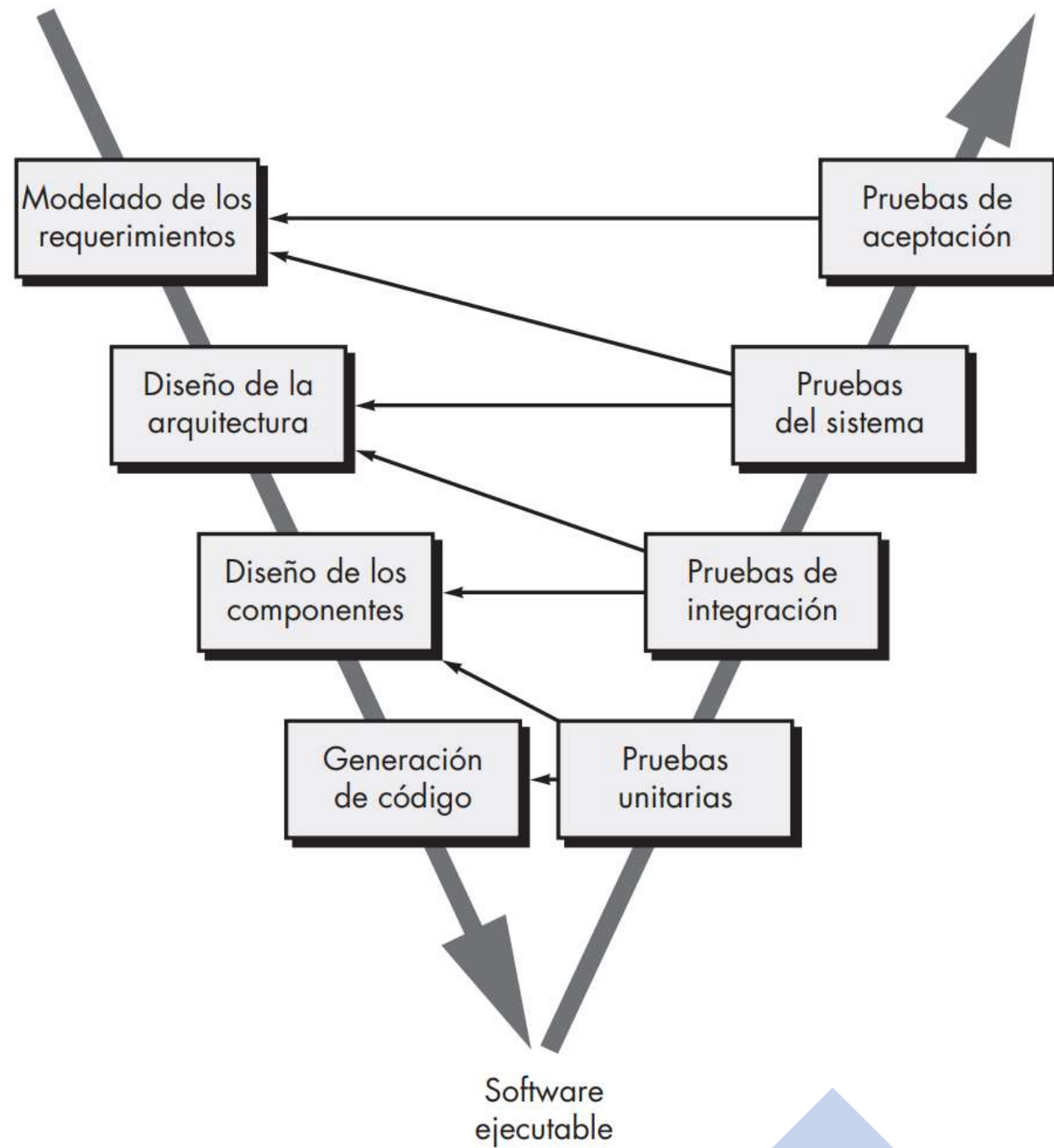
- **U4. Calidad enfocada al desarrollo de software**

- Qué es la calidad del software.
- Cómo obtener calidad de software (métodos, metodologías, estándares).
- Cómo controlar la calidad del software.
- Costo de la calidad del software.
- Nomenclatura y certificación ISO 9001:2000.
- La norma ISO/IEC 9126.
- Análisis de factores que determinan la calidad del software.
- Análisis del proceso del ciclo de vida del software.
- Funciones de evaluación del software.

Un poco de historia

Modelo de la cascada





“Dejemos de desperdiciar \$78 mil millones de dólares al año”

- “las empresas estadounidenses gastan miles de millones de dólares en software que no hace lo que se supone que debe hacer”
- [CIO Magazine: Let's Stop Wasting \\$78 Billion A Year | Linux Today](#)



Código defectuoso

Responsable del hasta el 45% del tiempo que están fuera los sistemas

Se requiere de 3 a 4 defectos por 1,000 líneas de código para que un programa tenga mal desempeño

Los programadores cometen 1 error cada 10 líneas

InformationWeek

Dr. W. Edwards Deming

- Padre de la evolución de la calidad
- Cuatro principios:
 - Quality improvement drives the entire economy
 - The customer always comes first
 - Do not blame the person, fix the system
 - Plan-Do-Check-Act



Quality improvement drives the entire economy

- “Organizations that focused on improving quality would automatically reduce costs while those that focused on reducing cost would automatically reduce quality and actually increase costs as a result”





The
customer
always
comes first

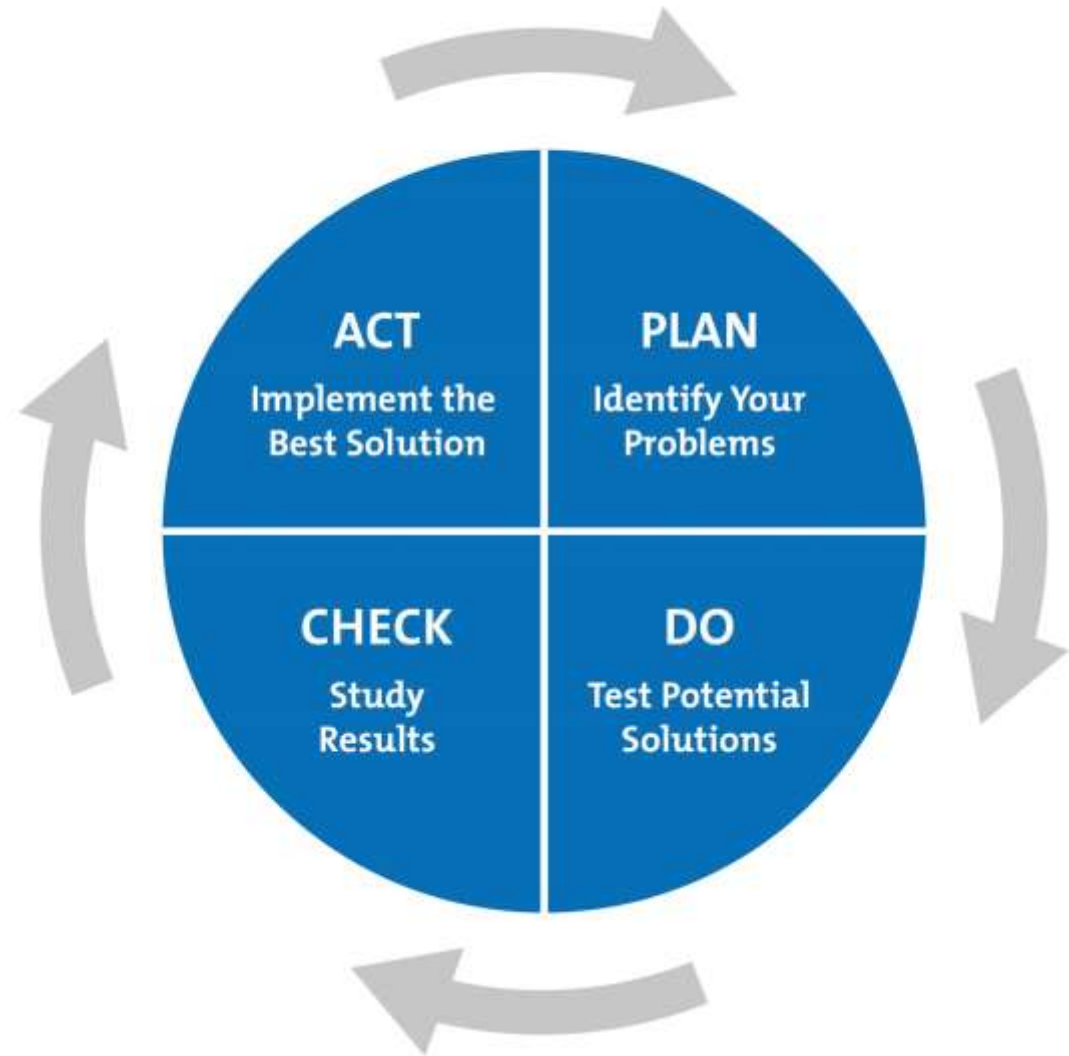
Do Not Blame The Person, Fix The System

Employees are internal
customers



Plan-Do-
Check-Act

Figure 1: The Plan-Do-Check-Act Cycle



¿Qué es calidad?

New Registratio

Choose User Id Enter User ID

Password Enter Password

Confirm Password

Name

Email (Requires verification. Will not be published.)

Country

 Please enter the verification number exactly as shown in left.

calidad¹

Del lat. *qualitas*, *-ātis*, y este calco del gr. ποιότης *poiótēs*.

1. f. Propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor. *Esta tela es de buena calidad.*
2. f. Buena **calidad**, superioridad o excelencia. *La calidad de ese aceite ha conquistado los mercados.*
3. f. Adecuación de un producto o servicio a las características especificadas. *Control de la calidad de un producto.*
4. f. Carácter, genio, índole.
5. f. Condición o requisito que se pone en un contrato.
6. f. Estado de una persona, naturaleza, edad y demás circunstancias y condiciones que se requieren para un cargo o dignidad.
7. f. Nobleza del linaje.
8. f. Importancia o gravedad de algo.
9. f. **pl.** Prendas personales.
10. f. **pl.** Condiciones que se ponen en algunos juegos de naipes.

El Zen y el Arte del mantenimiento de la motocicleta

Calidad... sabes lo que es, pero no sabes lo que es. Pero eso es una contradicción. Algunas cosas son mejores que otras; es decir, tienen más calidad. Pero cuando tratas de decir lo que es la calidad, además de las cosas que la tienen, todo se desvanece... No hay nada de qué hablar. Pero si no puede decirse qué es Calidad, ¿cómo saber lo que es, o incluso saber que existe? Si nadie sabe lo que es, entonces, para todos los propósitos prácticos, no existe en absoluto. Pero para todos los propósitos prácticos, en realidad sí existe. ¿En qué otra cosa se basan las calificaciones? ¿Por qué paga fortunas la gente por algunos artículos y tira otros a la basura? Es obvio que algunas cosas son mejores que otras... pero, ¿en qué son mejores? Y así damos vueltas y más vueltas, ruedas de metal que patinan sin nada en lo que hagan tracción. ¿Qué demonios es la Calidad? ¿Qué es?

David Garvin



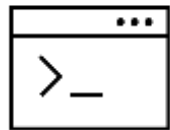
La calidad es un concepto complejo y de facetas múltiples



Usuario: Concibe la calidad en términos de las metas especificadas por el usuario final



Fabricante: Especificaciones originales del producto



Producto: Características inherentes



Valor: Lo que el cliente este dispuesto a pagar por él

Diseño y conformidad

Calidad de diseño: Grado en la que el diseño cumple las funciones y características especificadas en los requerimientos

Calidad de la conformidad: Grado en la que la implementación se apega al diseño y el sistema resultante cumple sus metas

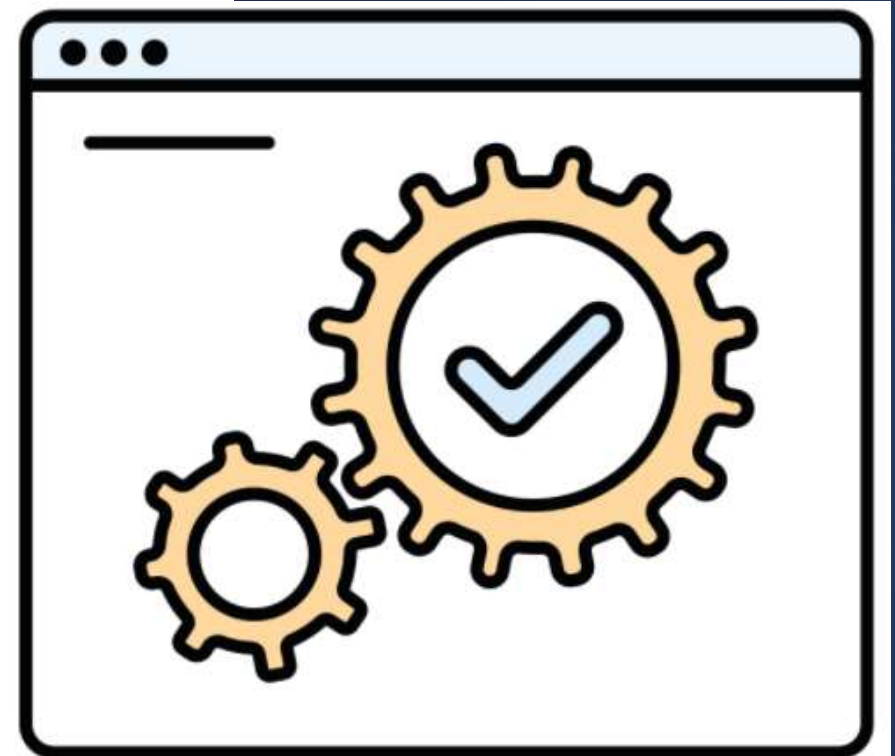
¿Es todo?

- Robert Glass: “si el usuario no está satisfecho, nada de lo demás importa.”

satisfacción del usuario = producto que funciona + buena calidad + entrega dentro del presupuesto y plazo

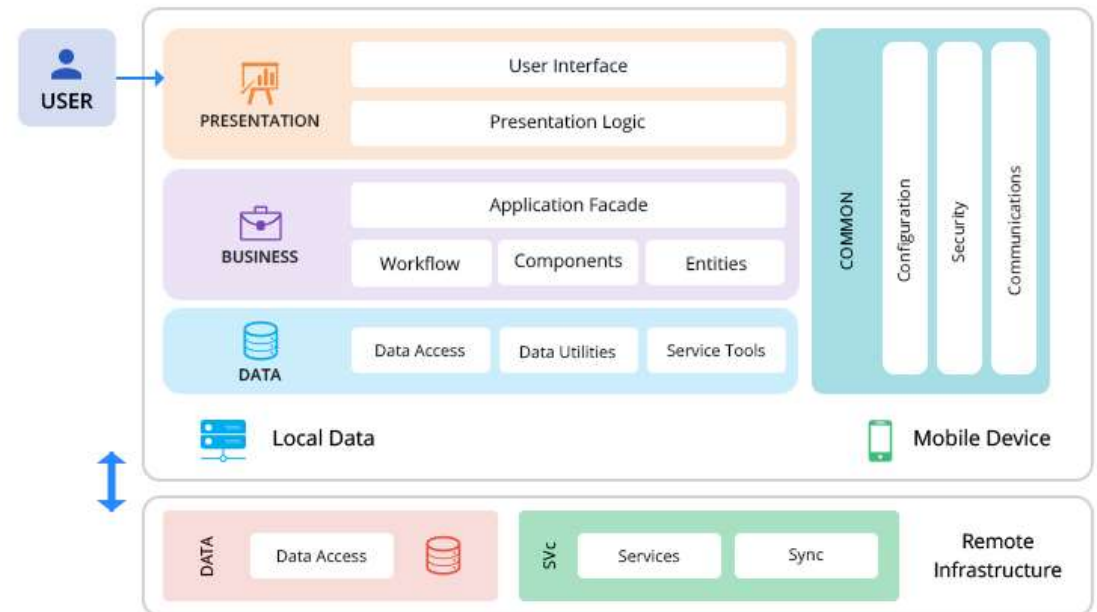
Calidad de Software

- “Proceso eficaz de software que se aplica de manera que crea un producto útil que proporciona valor medible a quienes lo producen y a quienes lo utilizan”



Proceso eficaz de software

- Establece infraestructura
- Analizar y diseñar
- Actividades sombrilla: Cambio y revisiones técnicas



Producto Útil

- Lo que el usuario final desea



Agrega
valor

Mayores utilidades por el
producto del Software

Mayor rentabilidad cuando
apoya un proceso de negocio

Mayor disponibilidad de
información

Dimensiones de calidad de Garvin

Desempeño:
Valor al usuario
final

Características:
Primera vista

Confiabilidad:
Disponibilidad

Conformidad:
Estándares

Durabilidad:
Mantenimiento

Servicios:
Rapidez de
mantenimiento

Estética:
Elegancia

Percepción:
Reputación

Calidad de McCall

Facilidad de recibir mantenimiento
Flexibilidad
Susceptibilidad de someterse a pruebas

Portabilidad
Reusabilidad
Interoperabilidad

REVISIÓN DEL PRODUCTO

TRANSICIÓN DEL PRODUCTO

OPERACIÓN DEL PRODUCTO

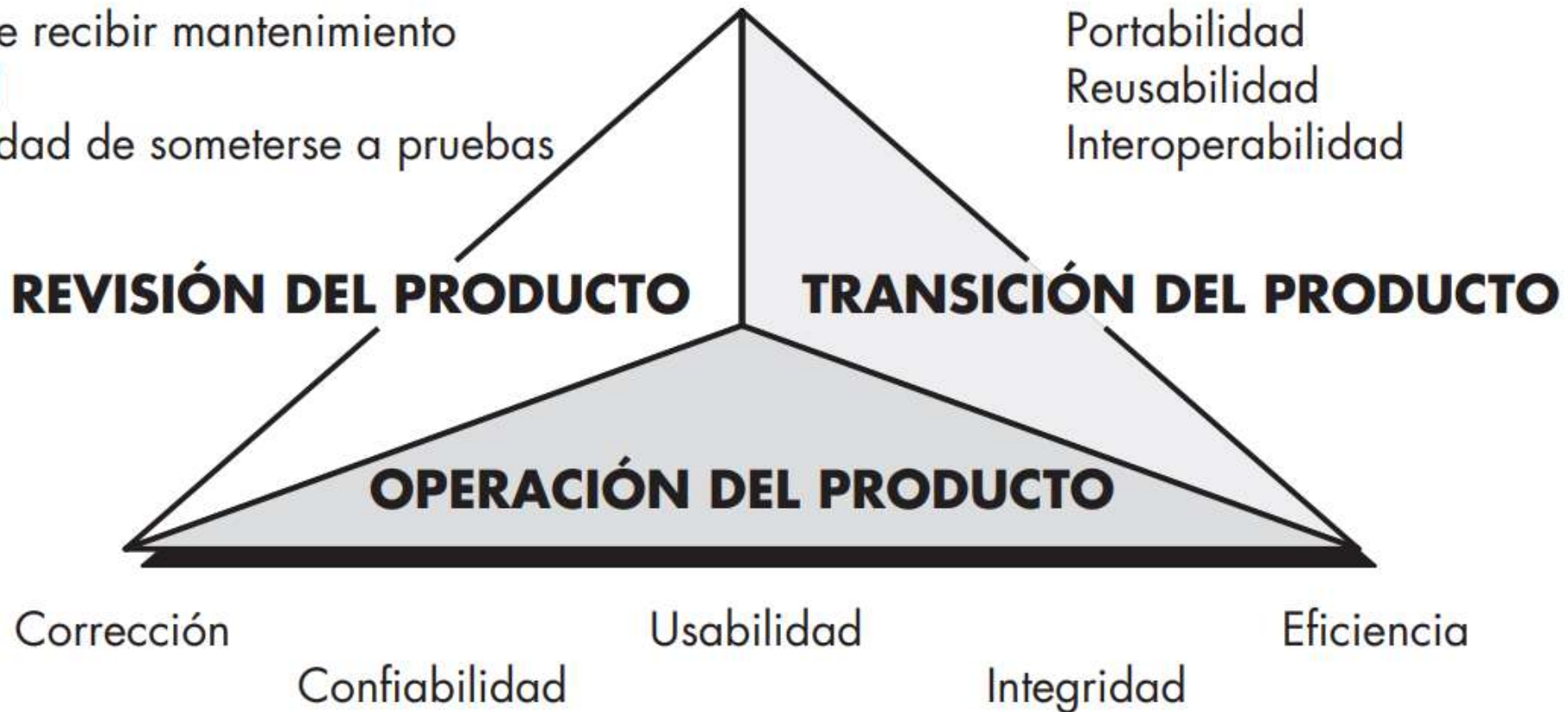
Corrección

Confiabilidad

Usabilidad

Integridad

Eficiencia





- Corrección: Cumple con los objetivos del cliente
- Confiabilidad: Cumple con su función en una precisión deseable
- Usabilidad: Esfuerzo para usar el sistema
- Integridad: Datos y usuarios
- Eficiencia: Recursos utilizados

Facilidad de recibir mantenimiento
Flexibilidad
Susceptibilidad de someterse a pruebas

REVISIÓN DEL PRODUCTO

- Facilidad de recibir mantenimiento: Detectar -> Corregir
- Flexibilidad: Modificar un programa en producción
- Susceptibilidad de someterse a pruebas: Esfuerzo para probar un sistema

- Portabilidad: Usar otro sistema
- Reusabilidad: Modular
- Interoperabilidad: Acoplar un sistema con otro



ISO 9126

<i>Características</i>	<i>Pregunta central</i>
<i>Funcionalidad</i>	¿Las funciones y propiedades satisfacen las necesidades explícitas e implícitas; esto es, el qué . . . ?
<i>Confiabilidad</i>	¿Puede mantener el nivel de rendimiento, bajo ciertas condiciones y por cierto tiempo?
<i>Usabilidad</i>	¿El software es fácil de usar y de aprender?
<i>Eficiencia</i>	¿Es rápido y minimalista en cuanto al uso de recursos?
<i>Mantenibilidad</i>	¿Es fácil de modificar y verificar?
<i>Portatilidad</i>	¿Es fácil de transferir de un ambiente a otro?

Tabla 1. Características de ISO-9126 y aspecto que atiende cada una.

Funcionalidad

Satisface las necesidades por la cual fue diseñado

- Adecuación
- Exactitud
- Interoperabilidad
- Conformidad
- Seguridad

Funcionalidad

- El software presenta resultados acordes a la necesidad a la que fue diseñado
- El software cuenta con funciones apropiadas para efectuar las tareas que fueron especificadas
- El software previene el acceso no autorizado
- El software sigue estándares de la industria
- El software tiene la habilidad de interactuar con otros sistemas

Adecuación

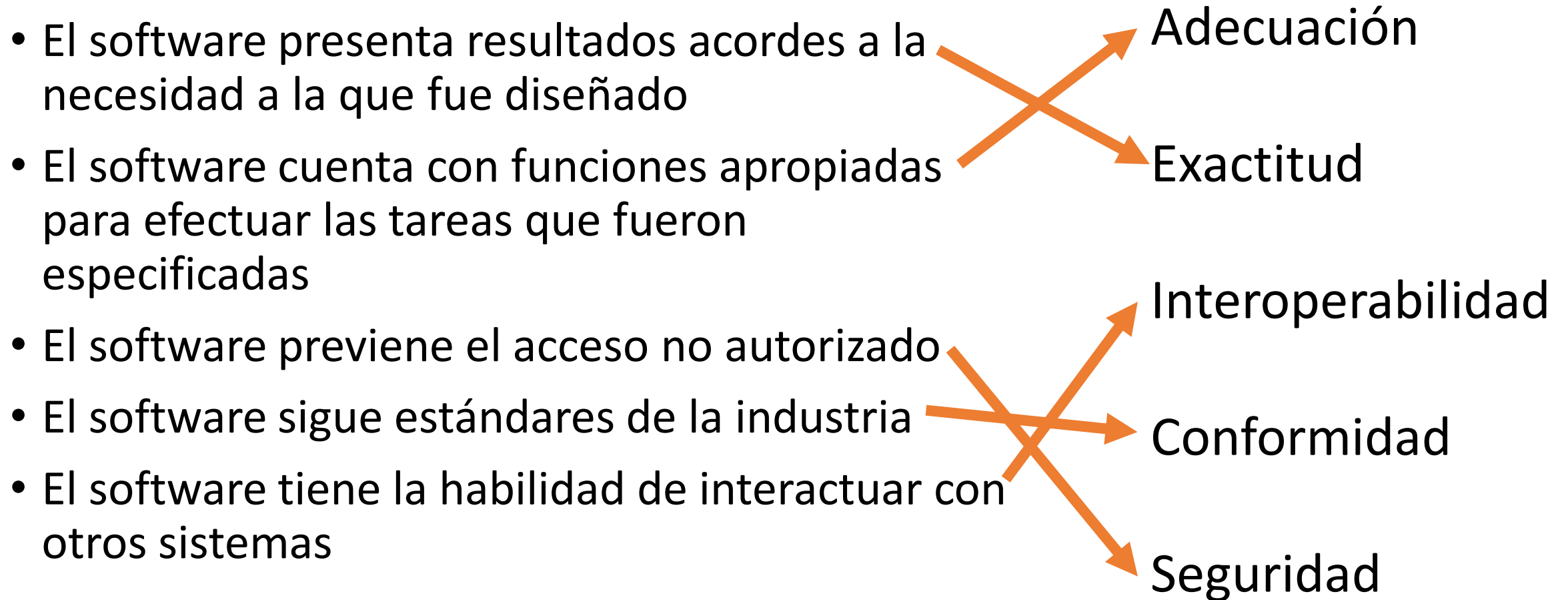
Exactitud

Interoperabilidad

Conformidad

Seguridad

Funcionalidad



Confiabilidad

Capacidad del software de mantener su nivel de ejecución bajo condiciones normales en un periodo de tiempo establecido

- Nivel de madurez
- Tolerancia a fallos
- Recuperación

Confiabilidad

Nivel de madurez
Tolerancia a fallos
Recuperación

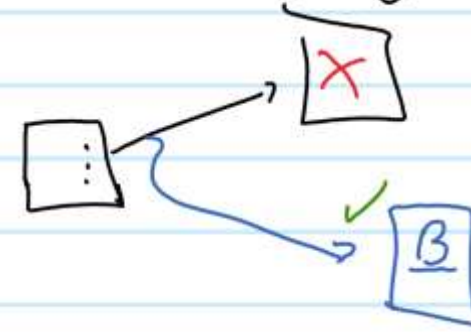
Tiempo medio entre averías



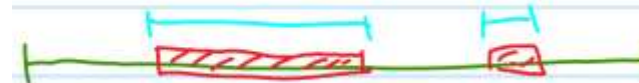
Cantidad de tiempo fuera de servicio



Chaos testing



Tiempo medio hasta la recuperación



Confiabilidad

Madurez

Tiempo medio entre averías



Disponibilidad

Cantidad de tiempo fuera de servicio



Capacidad de recuperación

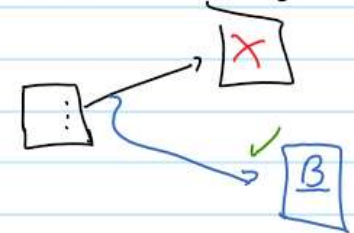
Tiempo medio hasta la recuperación



Area. Resiliencia

Tolerancia a fallos

Chaos testing



Usabilidad

Esfuerzo necesario que deberá invertir el usuario para utilizar el sistema

- Comprensibilidad
- Facilidad de aprender
- Operabilidad

Usabilidad

- Un usuario requiere de una sesión de 3 horas para entender lo que hace una aplicación
- Una aplicación fue desarrollada siguiendo los mismos flujos que otras aplicaciones en el mercado usan
- Métricas de conversión y pasos hasta lograr un objetivo son ejemplos de:

Comprensibilidad
Facilidad de aprender
Operabilidad

Eficiencia

Relación entre el nivel de funcionamiento del software y la cantidad de recursos usados

- Tiempo
- Recursos

Mantenibilidad

Esfuerzo necesario para realizar modificaciones al software

- Capacidad de análisis
- Modificación
- Estabilidad
- Facilidad de prueba

Mantenibilidad

- Un cambio de requerimientos produce un retrabajo en el código y en el modulo de pruebas
- Un cambio en el modulo C produce un cambio inesperado en la base de datos
- Un nuevo requerimiento del cliente produce un retrabajo en las funciones B y D
- Un nuevo requerimiento pide que además de estar disponible en Android, lo este también para iOS

Capacidad de análisis, Modificación, Estabilidad, Facilidad de prueba

Portabilidad

Habilidad del software de ser transferido de un ambiente a otro

- Adaptabilidad
- Facilidad de instalación
- Conformidad
- Capacidad de reemplazo

Portabilidad

- Un editor de videos guarda los videos en un formato único llamado “.vid7”
- Una aplicación sigue estándares de portabilidad
- Un instalador es un ejemplo de
- Un cambio en el requerimiento obliga a que el sistema sea transferido a un nuevo servidor

Adaptabilidad

Facilidad de
instalación

Conformidad

Capacidad de
reemplazo

Portabilidad

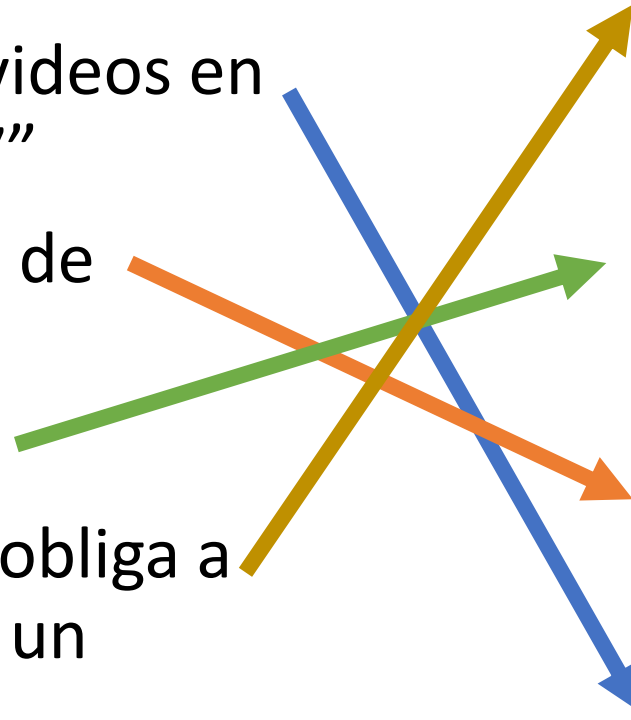
- Un editor de videos guarda los videos en un formato único llamado “.vid7”
- Una aplicación sigue estándares de portabilidad
- Un instalador es un ejemplo de
- Un cambio en el requerimiento obliga a que el sistema sea transferido a un nuevo servidor

Adaptabilidad

Facilidad de instalación

Conformidad

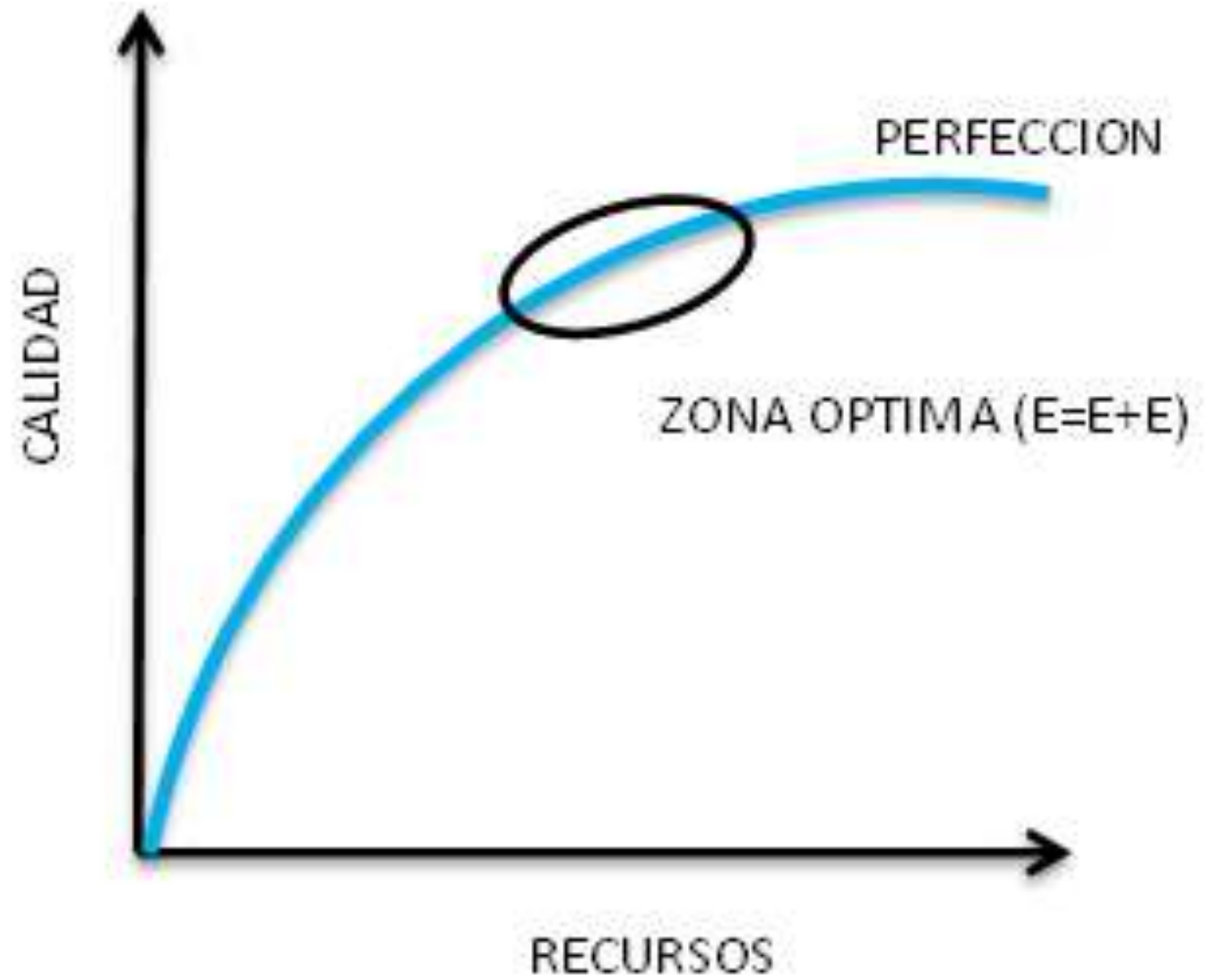
Capacidad de reemplazo





¿Qué es para ti calidad?

“Perfect is the enemy of good”



Costos



Prevención

Actividades de administración

Actividades técnicas:
Requerimientos y diseño

Planear las pruebas

Capacitación



Evaluación

Revisiones técnicas

Recabar datos

Hacer pruebas



Falla

Internos

Externos

Costo de corregir errores



¿Cómo aseguramos la calidad?



Métodos de la ingeniería
de software



Técnicas de administración
de proyectos



Control de calidad



Aseguramiento de la
calidad