

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computación



IC6831 Aseguramiento De La Calidad Del Software

Grupo 40

## **Actividad 01- Comprobación De Lectura**

Profesora:

Ericka Solano Fernández

Equipo 1:

Valery Mishel Carvajal Oreamuno – 2022314299

Luis Felipe Calderon Perez – 2021048663

Sergio Chavarría Avilés – 2021077879

Kevin Yadir Calvo Rodriguez – 2023367224

Andrew Denilson Lopez Herrera – 2021062132

Diciembre, 2024

## Índice

<b>Índice.....</b>	<b>1</b>
<b>Capítulo 1.....</b>	<b>2</b>
Pregunta 1.1.....	2
Pregunta 1.2.....	3
Pregunta 1.3.....	4
<b>Capítulo 2.....</b>	<b>6</b>
Pregunta 2.1.....	6
Pregunta 2.2.....	8
Pregunta 2.3.....	9
Pregunta 2.4.....	10
Pregunta 2.5.....	11
Pregunta 2.6.....	13
Pregunta 2.7.....	14

# Capítulo 1

## Pregunta 1.1

Hay tres diferencias principales entre los productos de software y otros productos industriales.

### **(1) Identificar y describir las diferencias.**

- Complejidad del producto: La complejidad de un producto se puede medir basado en los módulos de operaciones que contiene este producto. En el caso de los productos de software, cuesta más asegurar que todos sus módulos estén completos, debido a que pueden contener millones de operaciones distintas. En cambio, en otros productos industriales, no se permiten más que unas miles operaciones distintas.
- Visibilidad del producto: Los productos de software son invisibles, y también lo son sus errores. En productos industriales, se puede notar fácilmente si falta una pieza. En los productos de software podrían faltar piezas durante todo el proceso de desarrollo y no notarse hasta el final.
- Desarrollo del producto y proceso de producción: En los productos de software, la única etapa en la que se pueden detectar problemas es en la etapa de desarrollo. En otros productos industriales, se pueden detectar errores en todas las etapas: desarrollo, planeamiento de producción y producción.

### **(2) Analice las formas en que estas diferencias afectan la SQA.**

Esto afecta al aseguramiento de calidad de software, debido a que se puede evidenciar que la detección de errores es más compleja que en otro tipo de productos. Los demás productos deben verificar menos operaciones, tienen mayor facilidad para detectarlos y tienen más tiempo y métodos para detectarlos. En los sistemas de software se deben verificar millones de operaciones, sus errores son invisibles por lo que no es tan fácil detectarlos y además solo se pueden detectar durante una de las etapas del proceso de producción.

## Pregunta 1.2

Se afirma que no se espera que se hagan las actividades significativas de SQA en la fase de planeación de producción de los productos de software.

### **(1) Discuta esta afirmación**

Esta declaración es verídica, pues en el software es muy caro realizar actividades significativas de SQA en las diferentes fases; por ello se realizan las actividades más significativas en la fase inicial (desarrollo de software), en donde requiere menos esfuerzo y dinero en comparación a las otras 2 fases.

### **(2) Compare el requerimiento de planeación de producción para un nuevo modelo automovilístico con los esfuerzos de la planeación de producción de un nuevo producto de software.**

Comparación entre planeación de producción de nuevo software vs nuevo vehículo	
Nuevo software	Nuevo vehículo
No requiere líneas específicas de producción.	Puede necesitar una línea de producción específica.
Se realiza documentación y manuales del software.	Surgen oportunidades adicionales de inspeccionar el vehículo.
No surgen inspecciones adicionales en el software, debido a su costo.	Esta fase no suele omitirse en el proceso de producción de vehículos.
Esta fase puede ser omitida en el proceso de producción de software.	Los procesos y herramientas son diseñados y preparados en esta fase.

### Pregunta 1.3

Siete aspectos caracterizan el entorno profesional de desarrollo y mantenimiento de software.

#### **(1) Identifica y describe estas características.**

1. Condiciones contractuales: Las actividades están regidas por contratos que definen requisitos funcionales, presupuesto y cronograma del proyecto.
2. Relación cliente-proveedor: El equipo de desarrollo debe cooperar continuamente con el cliente, considerando sus solicitudes de cambios, críticas y aprobaciones.
3. Trabajo en equipo: Es necesario debido a requisitos de tiempo, especializaciones variadas y la necesidad de soporte profesional mutuo.
4. Cooperación y coordinación con otros equipos: Los proyectos suelen requerir colaboración con otros equipos de desarrollo, ya sea internos o externos, incluyendo equipos de hardware y software del cliente o de otros proveedores.
5. Interfaces con otros sistemas de software: Los sistemas deben interactuar con otros paquetes de software, ya sea para recibir datos (interfaces de entrada) o para transmitir información procesada (interfaces de salida).
6. Continuidad a pesar de cambios en el equipo: Es común que miembros del equipo se reemplacen durante el proyecto, lo que requiere adaptar rápidamente a nuevos integrantes sin alterar el cronograma.
7. Mantenimiento a largo plazo: Los clientes esperan que el software sea funcional por largos periodos, lo que implica brindar soporte y mantenimiento durante todo ese tiempo.

#### **(2) ¿Cuáles de estas características ambientales afectan principalmente los esfuerzos profesionales requeridos para llevar a cabo proyectos de desarrollo y mantenimiento de software? Enumera las características y explica por qué se necesita un esfuerzo profesional.**

1. Trabajo en equipo: Es necesario coordinar habilidades técnicas diversas y garantizar que cada miembro cumpla con sus responsabilidades específicas. Esto requiere experiencia técnica para cumplir con las tareas en los plazos establecidos.
2. Interfaces con otros sistemas: Integrar datos entre sistemas requiere experiencia técnica para diseñar y probar interfaces robustas.
3. Mantenimiento a largo plazo: El mantenimiento implica comprender el software a fondo, identificar problemas y solucionarlos sin afectar otras funcionalidades.

Estas características requieren habilidades técnicas avanzadas y experiencia específica en diseño, implementación y resolución de problemas para asegurar la calidad técnica del producto.

#### **(3) ¿Cuáles de estas características ambientales afectan principalmente los esfuerzos gerenciales requeridos para llevar a cabo proyectos de desarrollo y mantenimiento de software? Enumera las características y explica por qué se necesitan estos esfuerzos.**

1. Condiciones contractuales: Es necesario monitorear constantemente para cumplir con los requisitos funcionales, presupuesto y plazos.
2. Relación cliente-proveedor: La comunicación efectiva y la gestión de expectativas del cliente son esenciales para evitar conflictos y garantizar la aprobación de entregables.
3. Continuidad a pesar de cambios en el equipo: La gerencia debe asegurar que las transiciones sean fluidas, manteniendo la productividad y el cumplimiento de plazos.

Estas características demandan planificación, supervisión y gestión del equipo, así como habilidades de comunicación para asegurar el cumplimiento de los objetivos del proyecto y la satisfacción del cliente.

## Capítulo 2

### Pregunta 2.1

Un sistema de software consta de cuatro componentes principales.

#### **(1) Enumere los cuatro componentes de un sistema de software.**

- El programa (el código).
- Procedimientos.
- Documentación.
- Datos necesarios para el funcionamiento del sistema de software.

#### **(2) ¿Cómo contribuye la calidad de cada componente a la calidad del software desarrollado?**

- El programa: Este contribuye permitiendo la funcionalidad correcta del producto software. Sin él, la computadora no sabe qué operaciones realizar
- Procedimientos: Estos contribuyen al definir los métodos del proceso de desarrollo del programa. Un buen proceso de desarrollo, pruebas y despliegue, contribuye de buena forma a la calidad del software.
- Documentación: La documentación del desarrollo contribuye al permitir una cooperación y coordinación eficiente entre los miembros del equipo de desarrollo lo cual da lugar a un mejor producto.
- Datos necesarios para el funcionamiento del sistema de software: Contribuyen al permitir pruebas utilizando datos que incluyen parámetros, listas de códigos que ajustan el software a las especificaciones y archivos de casos de prueba.

#### **(3) ¿Cómo contribuye la calidad de cada componente a la calidad del mantenimiento del software?**

- El programa: La legibilidad y la modularidad del código hace que darle mantenimiento al software sea más sencillo
- Procedimientos: Definen a los responsables para realizar el mantenimiento del software, además de especificar los pasos a realizar para darle un correcto mantenimiento al software.
- Documentación: La documentación del desarrollo permite a los desarrolladores introducir cambios de manera apropiada, al ayudarles a entender lo que hace cada

parte del software. La documentación del mantenimiento provee toda la información necesaria al equipo de mantenimiento para poder ejecutar todas las tareas necesarias para darle mantenimiento al software.

- Datos necesarios para el funcionamiento del sistema de software: Mediante los datos de prueba, permiten probar nuevos cambios introducidos, para así corroborar el funcionamiento esperado de las nuevas funcionalidades y de las implementadas anteriormente.



## Pregunta 2.2

**(1) Defina error de software, defecto de software y falla de software. Explique las diferencias entre estos estados de software no deseados.**

- Error de software: Es un error producido por un programador, ya sea un error gramatical en las líneas de código, o un error lógico en la implementación de los requisitos del cliente.
- Defecto de software: Es un error producido por un error de software que puede provocar un funcionamiento incorrecto dentro del software.
- Falla de software: Es un error producido por un defecto de software que ha sido “activado” y provoca un funcionamiento incorrecto del software en el que fue implementado.

**(2) Sugiera una situación en la que aparece un nuevo tipo de falla de software (“bug”) en un paquete de software que ha estado prestando servicio a 300 clientes por primera vez seis años después de que el paquete de software se vendió por primera vez al público.**

Una aplicación de detección de humedad se programó para iniciar un proceso que prevenga daños a equipos por exceso de humedad. Para esto, se calcula la humedad relativa y, si esta excede el 80%, se activa dicho procedimiento. Este software se prestó a más de 300 clientes en El Cairo, desde hace más de 6 años. Sin embargo, hay un error de software, este procedimiento no se activa si la humedad excede el 80%, sino cuando excede el 90%, esto por un error del programador. Por la poca humedad de El Cairo, este problema nunca se manifiesta, y el sistema funciona sin problemas por todos estos 6 años. Sin embargo, al desplegar este sistema en Singapur, donde la humedad siempre es muy alta, el sistema no funciona, lo que termina provocando daños a los sistemas por una falla de software.

### Pregunta 2.3

#### **(1) Enumere y describa brevemente las distintas causas de errores de software.**

- Incorrecta definición de los requerimientos:

Mala especificación de los requerimientos. Esto suele suceder mayormente por parte del cliente. Se puede dar debido a definición errónea, ausencia de requerimientos, definición incompleta o inclusión de requerimientos innecesarios.

- Fallos en la comunicación cliente - desarrollador:

Mal entendimiento de las instrucciones del cliente, mal entendimiento de los cambios presentados del cliente escritos u orales, mal entendimiento de la respuesta del cliente a los problemas del diseño y falta de atención a los mensajes del cliente.

- Desviaciones de los requisitos de software:

En circunstancias, los desarrolladores se desvían de los requerimientos documentados. Reuso de módulos de software, presión por presupuesto o por cronograma o por inicio de mejoras no aprobadas.

- Errores lógicos de diseño:

Errores lógicos pueden ingresar al sistema cuando los profesionales que diseñan el sistema formulan los requerimientos de software.

- Errores de código:

Mal entendimientos de la documentación del diseño, errores en los lenguajes de programación, en el uso de herramientas de desarrollo, errores en la selección de data y otros.

- No cumplir con documentación e instrucciones de código:

Los miembros del equipo de desarrollo deben apegarse a las reglas de documentación y codificación establecidas para el proyecto. En caso de que no se apeguen a estas reglas, el mantenimiento de este código a futuro se puede complicar y aumentar la cantidad de errores.

- Deficiencias del proceso de prueba:

Estas deficiencias afectan debido a que muchos errores se quedan sin detectar o corregir.

- Errores de procedimiento:

Los procedimientos guían a los desarrolladores con respecto a sus actividades. Son importantes para asegurar el seguimiento de los pasos a través de las distintas actividades del desarrollo.

- Errores de documentación:

Los errores de documentación que afectan el desarrollo y el mantenimiento, son los errores en el diseño de los documentos o en la documentación.

**(2) Clasifique las causas del error según los grupos responsables del error: el personal del cliente, los analistas de sistemas, los programadores, el personal de prueba - o es una responsabilidad compartida que pertenece a más de un grupo?**

<b>Error</b>	<b>Grupo Responsable</b>
Incorrecta definición de los requerimientos	el personal del cliente
Fallos en la comunicación cliente - desarrollador	el personal del cliente - los analistas de sistemas
Desviaciones de los requisitos de software	los programadores
Errores lógicos de diseño	los analistas de sistemas
Errores de código	los programadores
No cumplir con documentación e instrucciones de código	los programadores
Deficiencias del proceso de prueba	el personal de prueba
Errores de procedimiento	los analistas de sistemas - los programadores - el personal de prueba
Errores de documentación	los analistas de sistemas - los programadores - el personal de prueba

#### Pregunta 2.4

**¿Cuáles son las diferencias entre la definición IEEE de SQA y la definición ampliada utilizada en este libro?**

La definición de calidad que da la IEEE es la siguiente:

- “1. El grado en el que un sistema, componente o proceso cumple con los requisitos especificados.
2. El grado en el que un sistema, componente o proceso cumple con las necesidades o expectativas del cliente o usuario.”

La definición que amplía el libro es la siguiente:

“Conformidad con los requisitos funcionales y de rendimiento establecidos explícitamente, los estándares de desarrollo documentados explícitamente y las características implícitas que se esperan de todo software desarrollado profesionalmente.”

La diferencia entre ambas definiciones es la manera en que se mide la calidad del software. La definición de la IEEE mide la calidad del software según las expectativas de los clientes y los desarrolladores. Mientras que, la definición ampliada en el libro, utiliza métricas dadas por la propia definición. De esta manera, se puede conseguir que para una definición cierto software sea de calidad mientras que para la otra definición, el mismo software, no sea de calidad.

### Pregunta 2.5

El señor Johnson es cliente de las tiendas Adams y Lincoln de la cadena Eiffel. Sus registros de compras y devoluciones son los siguientes:

Month	Adams Store		Lincoln Store	
	Purchases (\$000)	Returned goods (\$000)	Purchases (\$000)	Returned goods (\$000)
Jan 2000	100	20	60	5
Feb 2000	120	10	40	–
Mar 2000	10	–	30	10
Apr 2000	80	5	50	10
May 2000	30	–	20	–
Jun 2000	60	20	30	10
Jul 2000	10	–	40	–
Aug 2000	60	5	10	–
Sep 2000	20	–	20	5
Oct 2000	20	5	40	10
Nov 2000	40	–	20	–
Dec 2000	20	–	60	5
Jan 2001	30	10	40	–
Feb 2001	60	5	30	5
Mar 2001	20	5	40	10

(1) ¿En cuál de los meses (enero de 2001, febrero de 2001 o marzo de 2001) el Sr. Johnson califica para el descuento del 5%? ¿Cuál es la suma descontada? Calcule según el procedimiento correcto.

#### Procedimiento Correcto

- Recopila los datos de ventas y devoluciones de mercancías del mes anterior de cada uno de sus clientes de todas las tiendas de la cadena.**
  - Enero:  
Ventas del mes anterior (diciembre 2000) = 80  
Devoluciones del mes anterior (diciembre 2000) = 5
  - Febrero:  
Ventas del mes anterior (enero 2001) = 90  
Devoluciones del mes anterior (enero 2001) = 10
  - Marzo:  
Ventas del mes anterior (febrero 2001) = 60  
Devoluciones del mes anterior (febrero 2001) = 15
- Calcula las compras acumuladas de cada cliente durante los últimos 12 meses en todas las tiendas de la cadena.**  
Compras acumuladas (para enero) = 990  
Compras acumuladas (para febrero) = 900  
Compras acumuladas (para marzo) = 830
- Calcula el porcentaje de devoluciones de los últimos 3 meses de cada cliente en todas las tiendas de la cadena.**

Porcentaje de las devoluciones (para enero) =  $1/10$   
 Porcentaje de las devoluciones (para febrero) =  $1/14$   
 Porcentaje de las devoluciones (para marzo) =  $5/48$

- d. **Prepara una lista de todos los clientes que merecen el 5% de descuento y la distribuye a cada tienda antes de fin de mes.**

Ningún mes aplica, debido a que en ningún mes se pasó del millón de dólares.

**(2) Según los procedimientos erróneos, ¿en cuál de los meses (enero de 2001, febrero de 2001 o marzo de 2001) el Sr. Johnson califica para el descuento del 5% en la tienda Adams y en la tienda Lincoln? ¿Cuál es la suma descontada?**

- a. **Recopila los datos de ventas del año anterior para cada uno de los clientes de todas las tiendas de la cadena.**

Jan 2000	100	20	60	5
Feb 2000	120	10	40	—
Mar 2000	10	—	30	10
Apr 2000	80	5	50	10
May 2000	30	—	20	—
Jun 2000	60	20	30	10
Jul 2000	10	—	40	—
Aug 2000	60	5	10	—
Sep 2000	20	—	20	5
Oct 2000	20	5	40	10
Nov 2000	40	—	20	—
Dec 2000	20	—	60	5

- b. **Calcula las compras acumuladas de cada cliente durante el año anterior en todas las tiendas de la cadena.**

Compras acumuladas = 990

- c. **Prepara una lista de todos los clientes cuyas compras superan el millón de dólares y la distribuye a todas las tiendas.**

No aplica, debido a que las compras no superan el millón de colones.

## Pregunta 2.6

Acorde a la definición de SQA del IEEE, el control de calidad (CC) no se equipará con el aseguramiento de la calidad (AC).

### (1) ¿En qué aspectos el CC se diferencia del AC?

Diferencias entre CC y AC	
El control de calidad es un conjunto de actividades diseñadas para evaluar la calidad de lo desarrollado antes que el producto sea público.	En cambio el aseguramiento de la calidad minimiza el costo de garantizar la calidad por distintas actividades durante los procesos de desarrollo y fabricación.
Son una pequeña parte del aseguramiento de la calidad.	Reducen la tasa de productos rechazados.
Retiene o desecha cualquier producto que no cumpla con la calidad.	Reduce los costos de garantizar la calidad en muchos casos.

### (2) ¿Por qué el CC puede considerarse parte del AC?

Porque, aunque sirven a objetivos distintos, el aseguramiento de la calidad cuando no logra cumplir su objetivo, entra en acción el control de calidad, para evitar que el producto que se envía al público merme la confianza de los clientes con la entidad fabricante.

## Pregunta 2.7

Examina las definiciones de SQA y los objetivos de las actividades de SQA.

### **(1) ¿Existe correspondencia entre las dos definiciones?**

Sí, existe correspondencia entre las definiciones de Software Quality Assurance (SQA) y los objetivos de las actividades de SQA. La definición de SQA se centra en un conjunto planificado y sistemático de actividades necesarias para asegurar que el software cumpla con los requisitos funcionales y no funcionales. Mientras que, los objetivos de las actividades de SQA están diseñados para garantizar que los estándares de calidad sean alcanzados y mantenidos a lo largo de todo el ciclo de vida del software.

### **(2) Si la respuesta es sí, muestra cómo los objetivos de las actividades de SQA están orientados a implementar los conceptos de SQA.**

- Prevención de defectos: Diseñando procesos y metodologías para minimizar la aparición de errores durante las etapas tempranas del desarrollo. Esto se alinea con el concepto de calidad como prevención.
- Evaluación y mejora continua: Monitoreando el cumplimiento de estándares, realizando auditorías y aplicando métricas de calidad para detectar áreas de mejora, lo cual implementa el concepto de control de calidad.
- Cumplimiento de requisitos: Asegurando que el software cumpla con las especificaciones funcionales, de documentación y de datos necesarios, lo que refleja el objetivo principal del SQA de entregar un producto conforme a lo acordado.
- Gestión del mantenimiento: Garantizando la calidad en procesos de mantenimiento para preservar el funcionamiento adecuado del software a lo largo del tiempo.