

1. Respecto a la ingeniería de software:
 - a) La presión creciente por reducir los tiempos de salida al mercado (time to market) ha motivado la búsqueda de alternativas al modelo de proceso en cascada.
 - b) Al construir y evaluar software conviene despreocuparse de las interacciones de este software con el hardware y con las personas, y concentrarse en la producción de software de calidad.
 - c) El alza de los precios del hardware respecto a los precios del software, como resultado del alza en los precios del petróleo, está teniendo un fuerte impacto en la ingeniería de software.
 - d) La preocupación por la arquitectura de software y por los procesos se contraponen a la preocupación por reducir y manejar los riesgos en los proyectos.

 2. Un programa falla cada vez que, y sólo si, accede al último de 10000 registros de una tabla. ¿Para cuál de los usuarios siguientes resultará menos confiable el programa? Para quien lo usa 30 veces al día
 - a) y accede en cada vez a uno de los primeros 10 registros, seleccionado al azar.
 - b) y accede en cada vez a uno de los últimos 100 registros, seleccionado al azar.
 - c) y accede en cada vez a uno de los primeros 1000 registros, seleccionado al azar.
 - d) y accede en cada vez a dos registros al azar.

 3. Entre las cualidades más relevantes para un software que soporta el pago de servicios y productos mediante el uso de teléfono celular están:
 - a) mantenibilidad, modularidad
 - b) verificabilidad, visibilidad del proceso
 - c) seguridad de la información (security) , facilidad de uso
 - d) seguridad de las personas (safety), legibilidad (del código)

 4. Con respecto a modelos de proceso:
 - a) El modelo de proceso en cascada está orientado al manejo del riesgo de construir el software con una arquitectura que no permita cumplir con (parte de) los requerimientos no funcionales.
 - b) Una desventaja de un modelo en fases con liberaciones parciales es que no favorece la retroalimentación temprana de parte de cliente/usuario
 - c) El modelo de ciclo de vida en espiral de Boehm incorpora los conceptos de riesgo y costo al propio modelo, por lo que solo tiene sentido su utilización en proyectos de bajo riesgo y bajo costo
 - d) El proceso de Especificación Operacional exige contar con una herramienta que permita ejecutar especificaciones, de forma que estas puedan ser evaluadas mediante su ejecución.

 5. Construir un prototipo:
 - a) consiste en construir una parte o un aspecto del producto para evaluar la factibilidad y/o adecuación del enfoque y disminuir o acotar riesgos relacionados con los requerimientos y su implementación
 - b) exige un esfuerzo adicional que sólo se justifica si el prototipo puede evolucionar al producto final (prototipo evolutivo)
 - c) consiste en definir la interfaz de usuario de forma que quienes la vayan a utilizar puedan experimentar su utilización
 - d) resulta adecuado para evaluar aspectos de diseño del software, no así para evaluar o validar requerimientos

 6. Un modelo de proceso definido
 - a) puede ser utilizado para describir un proceso, lo que permite razonar acerca de sus propiedades y características, lo que resulta esencial para encarar la mejora del proceso
 - b) (a), y facilita la incorporación de personal nuevo en una organización de desarrollo de software
 - c) (b), y si el mismo modelo es aplicado en todos los proyectos de una organización, se facilita la rotación de personal entre los distintos proyectos de la organización
 - d) utilizando el esquema ETVX (Entry, Task, Verification, eXit), solo puede ser utilizado como modelo descriptivo, no como prescriptivo, ya que este esquema establece relaciones entre diversos factores abstractos al estilo del modelo de factores de Abdel-Hamid
-

7. El alcance de un proyecto de SW incluye:
- solo las características del producto a construir
 - (a) y eso se llama alcance del producto
 - (b) y se mide contra los requerimientos
 - ninguna de las anteriores
-
8. ¿Cómo acortar la duración total de un proyecto?
- identificando actividades del camino crítico
 - (a) y comprimiendo su duración mediante el agregado de recursos
 - (b) o bien paralelizando actividades con precedencia entre ellas
 - (c) con lo cuál puede ser que aumente el costo por retrabajo
-
9. La organización o estructura más apropiada de un equipo de desarrollo:
- depende en gran medida de la formación y estilos de trabajo de los miembros.
 - (a) y de la cantidad de integrantes
 - (b) y conviene minimizar las líneas de comunicación en proyectos grandes
 - (c) y conviene una estructura democrática en proyectos de gran incertidumbre.
10. La matriz de costos de Wolverton:
- da el costo actual en USD por LOC de distintos tipos de programas
 - da la complejidad de distintos tipos de programas
 - es una tabla en base a datos históricos que me da una buena estimación de los costos, si dispongo de estabilidad tecnológica y características similares de personal.
 - puede ser utilizada como modelo de estimación actualizando los valores y calibrándola con datos de mi organización.
-
11. Los riesgos:
- hay que priorizarlos porque no se pueden atender todos
 - se toman acciones para reducirlos dependiendo del nivel de reducción = severidad / costo de reducción
 - reducirlo implica evitarlo, mitigarlo o aceptarlo si ocurre
 - son generalmente los mismos para todos los proyectos y las actividades de reducción deben ser incluidas en el WBS del proyecto.
-
12. Referente a las reuniones del equipo de desarrollo:
- deben continuarse hasta terminar de tratar todos los puntos en la agenda
 - fracasan si hay conflictos
 - se arma la agenda entre todos al comienzo de la reunión
 - es necesario el rol de secretario
-
13. Los requisitos del sistema y los requisitos del software:
- son lo mismo
 - los requisitos del sistema incluyen los del software
 - los requisitos del software incluyen las personas, los procesos manuales y otros elementos de soporte
 - los requisitos del sistema incluyen las personas, los procesos normales, pero no los elementos de soporte
-
14. Corregir un error de requisitos es mucho más costoso en la fase de:
- Diseño
 - Implementación
 - Verificación
 - Producción
-

15. Los Requisitos No Funcionales:
- son decisiones de diseño
 - son restricciones que limitan las decisiones de diseño
 - incluyen todo lo que el sistema no debe hacer
 - incluyen todo lo relativo al alcance del proyecto
-
16. En la obtención de requisitos:
- solo participan los clientes y miembros del grupo de desarrollo
 - se obtiene el mismo resultado sustituyendo las entrevistas por encuestas
 - en las entrevistas se debe seguir estrictamente la agenda prevista de preguntas
 - si se utiliza la técnica de observación, se deben considerar períodos normales y atípicos.
17. La técnica de tormenta de ideas:
- consiste en exponer ideas lógicamente relacionadas
 - cada idea que se expone no debe ser inmediatamente discutida
 - en la fase de reducción, si la idea no tiene consenso, se elimina
 - se pueden priorizar las ideas tantas veces como se quiera.
-
18. La definición de requisitos y la especificación de requisitos difieren en:
- los términos que utilizan
 - (a) y a quién van dirigidas
 - (b) y el momento del ciclo de vida en que se hacen
 - (c) y el contenido
-
19. La Arquitectura de Software de un sistema
- son los subsistemas que componen el sistema, las interfaces de estos y las reglas de interacción entre ellos
 - (a) y una de las ventajas de documentar la Arquitectura de Software es que se puede mejorar y aumentar la comunicación con los distintos stakeholders
 - (b) y el estilo y estructura elegidos para dicha arquitectura depende fuertemente de los requerimientos no funcionales
 - Los requerimientos no funcionales de un sistema no son contradictorios entre sí, por lo que siempre se satisfacen todos con la arquitectura de software escogida.
-
20. Cambiar la Arquitectura de Software de un sistema ya construido
- Es relativamente sencillo si se usó el estilo en capas para la misma
 - Requiere mucho esfuerzo
 - (b) por lo que es conveniente realizar evaluaciones de la misma antes de implementarla completamente
 - es algo que se debe realizar periódicamente. Aproximadamente cada 8 meses se producen cambios tecnológicos que obligan a cambiar la arquitectura del sistema.

Ejercicio 21

Se desea construir un sistema que permita registrar los pedidos de un restaurante que realiza reparto de comidas a domicilio.

Los pedidos serán realizados por los clientes utilizando el sitio web del Restaurante, y serán luego entregados por los funcionarios del restaurante en el domicilio de aquellos.

Los clientes pueden ingresar a la página web del restaurante. Para realizar cualquiera de las operaciones del sitio, deberán loguearse con su usuario y clave, o de no tenerlo, registrarse previamente como cliente, indicando nombre, domicilio, usuario y clave. El sistema validará que el usuario no esté repetido.

Podrán realizar un pedido consultando allí el menú del día (se despliega el id de comida, la descripción y el precio), seleccionando una o más comidas y registrando la cantidad de cada una que desean.

Al finalizar el pedido, el sistema desplegará el monto total y los clientes podrán seleccionar el pago en efectivo o con tarjeta de crédito. En este último caso, cada transacción de pago es verificada consultando el sistema que brinda el ente emisor de la tarjeta para este cometido. Dado un número de tarjeta, el monto a pagar y la cantidad de cuotas, el sistema verificador valida o rechaza la transacción.

El cliente podrá también imprimir los datos del pedido realizado y el estado (pendiente de pago o la confirmación del pago).

Por último, el personal del restaurante podrá consultar los pedidos pendientes (el id de pedido, su contenido, nombre y domicilio del cliente y si pagó o no).

También podrá marcar los pedidos a medida que son despachados.

Los menús del día son mantenidos a través de otro sistema externo.

Se pide:

- 1) Se desea calcular los Puntos de Función y se pide:
 - a. identificación y análisis de las transacciones (nombre, tipo, cantidad de FTRs y DETs), y
 - b. identificar los archivos involucrados (nombre, campos, tipo, RETs y DETs).
- 2) Identificar y describir los actores del sistema
- 3) Dibujar el diagrama de contexto del sistema
- 4) Realizar la descripción detallada de los casos de uso que son iniciados por el cliente.