

- 
- 1) Respecto a la ingeniería de software
- La ingeniería de software se ha visto impactada por las presiones por reducir los plazos para salir al mercado ("time to market"), lo que ha llevado a buscar alternativas al modelo de proceso en cascada tales como el desarrollo en fases.
  - La relación de la ingeniería de software con la ingeniería de sistemas se basa en el hecho de que la ingeniería de software se ocupa de la construcción de sistemas compuestos por elementos de software que interactúan entre sí.
  - La reducción relativa del costo del software respecto al costo del hardware continúa teniendo hoy un fuerte impacto en la evolución de la ingeniería de software tal como en sus comienzos
  - "Desarrollo de Software", "Ingeniería de Software" y "Programación" corresponden a tres formas de expresar esencialmente el mismo significado.
- 
- 2) Respecto a modelos de proceso
- Los modelos "Desarrollo en Fases", "Especificación Operacional" y "Transformacional", son muy poco utilizados en la industria
  - El modelo de proceso en espiral es especialmente adecuado para proyectos que involucran poco riesgo
  - El esquema ETVX (Entry, Task, Verification, eXit), por su estructura, no permite especificar procesos del tipo "en fases", ya que fue concebido para la especificación de procesos "en cascada"
  - Los modelos propuestos por Abdel-Hamid vistos en el curso (Factores que inciden en la productividad y Estructura de desarrollo de software) son de tipo descriptivo y son útiles para entender la relación entre diversos factores y su incidencia en el desarrollo de software
- 
- 3) La gestión de riesgos de un proyecto:
- incluye identificar riesgos técnicos asociados al producto
  - solo incluye identificar riesgos del proceso
  - incluye evaluar la severidad de un riesgo y tomar siempre acciones correctivas
  - se realiza una vez al comienzo de un proyecto.
- 
- 4) Dados un requerimiento que pide imprimir la información de usuarios, que consiste en nombre de usuario, password, y un campo interno para marcar borrados lógicos, ¿cuál es la contribución de las transacciones y cuál la de los datos? (No tomar en cuenta ni mensajes ni errores)
- |                                  |               |
|----------------------------------|---------------|
| a) Transacción: 1 ILF con 2 DETs | Datos: 2 DETs |
| b) Transacción: 1 ILF con 3 DETs | Datos: 2 DETs |
| c) Transacción: 1 ILF con 2 DETs | Datos: 3 DETs |
| d) Transacción: 1 ILF con 3 DETs | Datos: 3 DETs |
- 
- 5) El diagrama de estados de UML
- es una técnica válida para especificar requerimientos
  - da una vista estática del sistema
  - no permite modelar paralelismo de estados
  - todas las anteriores
- 
- 6) Al elegir una técnica para especificar requerimientos se debe elegir:
- la que sea completa
  - una que sea dominada a priori tanto por los desarrolladores como por los clientes.
  - teniendo en cuenta las características del proyecto
  - la más adecuada y no mezclarla con otras
- 
- 7) Según [Scharer '90] existe una brecha en la comunicación, en la obtención de los requerimientos. Los desarrolladores dicen que los usuarios:
- no son capaces de mantener el cronograma
  - no conocen el negocio
  - pretenden indicarles cómo hacer su trabajo
  - les exigen una dedicación exclusiva a costa de sus otras obligaciones
- 
- 8) El proceso de diseño tiene distintos niveles de abstracción, siendo el de definición de la Arquitectura
- de las últimas actividades en realizarse, ya que en etapas tempranas del diseño difícilmente se cuenta con el conocimiento necesario para poder definirla en forma correcta.
  - en el que se deben concentrar los esfuerzos del equipo de diseño para identificar una correcta modularización del sistema que cumpla los requerimientos establecidos.
  - a) y es importante tener en cuenta las características (ventajas y desventajas) de los estilos de Arquitectura existentes para poder evaluar su adecuación a la aplicación en desarrollo.
  - b) y a esa altura del proceso, no es importante aún la definición de interfaces entre módulos, ya que posiblemente los cambios que todavía sufra la aplicación obliguen a re-definirlas varias veces.
-

- 9) Los métodos de evaluación de arquitecturas se pueden usar para
- conocer si la arquitectura cumple con los requisitos de calidad establecidos para el producto
  - (a) y la evaluación temprana permite introducir mejoras reduciendo el impacto de los cambios**
  - (a) y las evaluaciones no tempranas de la arquitectura, por ejemplo, luego de finalizado el producto, no tienen ningún beneficio.
  - Ninguna de las anteriores

- 10) Entre las técnicas vistas para mejorar el diseño se encuentra el diseño por contrato, que establece:
- un contrato entre un proveedor (servidor) y un consumidor (cliente) de un servicio, el cual define obligaciones para el cliente que son los requisitos del servidor, en forma de precondiciones
  - a) y beneficios para el cliente que representan el compromiso de los servicios otorgados por el servidor, en forma de postcondiciones**
  - a) y si no se cumplen las precondiciones establecidas el servidor debería elevar una excepción para que sea tratada por el cliente
  - Ninguna de las anteriores

- 11)
- El estilo de arquitectura de tubos y filtros es razonable usarlo cuando la parte central del sistema a construir es la transformación sucesiva de datos de entrada
  - (a) y en este estilo cada filtro debe ser independiente del resto y no debe conocer la identidad de los filtros que están antes o después de él.
  - (b) y el estilo de arquitectura "invocación implícita" tiene como característica que facilita la reutilización de componentes**
  - El patrón Model-View-Controller da una solución para el acceso a una base de datos relacional desde un sistema orientado a objetos.

- 12)
- Para generar casos de prueba utilizando la técnica de caja blanca se necesita del código
  - (a) y si se realizan pruebas solamente basadas en caja negra parte del código puede no ser ejecutado
  - (b) y una de las técnicas de caja negra, vista en el curso, es la técnica de partición en clases de equivalencia**
  - (c) y otra técnica de caja negra es el criterio de cubrimiento de decisión

- 13) Dentro de una empresa existen posibles conflictos entre los equipos de verificación y de desarrollo. Algunas de las posibles soluciones vistas en el curso son las siguientes:
- Tener el mismo jefe y gerente en ambos grupos (desarrollo y verificación). De esta forma, si surge un problema el jefe y/o gerente estará en conocimiento de todos los detalles del mismo
  - (a) y realizar evaluaciones personales
  - (b) y tener voluntad de mejora tanto personal como grupal
  - Trabajar en equipo, teniendo roles distintos pero igual objetivo**

- 14) Dado el método `explosionLista` indique el conjunto de datos de prueba con menor cardinalidad y que cumple con el criterio de decisión de las opciones listadas a continuación.

```
void explosionLista (lista : Lista) {
    int a = 0;
    while lista.tieneMasElementos() and a < 2{
        lista.borrarPrimerElemento(); //reduce la cant. de elem. en la lista en 1
        a = a+1;
    }
}
```

- CDP = { lista = vacía }
- CDP = { lista = vacía ; lista = {e11, e12, e13, e14} }**
- CDP = { lista = vacía ; lista = {e11, e12} ; lista = {e11, e12, e13, e14, e15, e16} ; lista = {e11} }
- CDP = { lista = {e11, e12} ; lista = {e11, e12, e13, e14, e15, e16} ; lista = {e11} }

- 15) La verificación estática (análisis)
- a) Son efectivas en la detección temprana de defectos
  - b) (a) y sirven para verificar cualquier producto (requerimientos, diseño, código, etc.)
  - c) (b) y están sujetas a los errores de nuestro razonamiento**
  - d) (c) y se usan para validar junto con el cliente

- 16) La planificación de la liberación de un sistema de software

- a) Conviene comenzarla cuando el producto está terminado, una vez que está claro en qué consiste realmente el producto a liberar.
- b) **Conviene comenzarla al comienzo del proyecto, para incluir sus particularidades en el alcance del proyecto y completarla y ajustarla antes de proceder a la liberación en sí.**
- c) Conviene completarla junto con la especificación de requerimientos, una vez que está claro qué características debe tener el producto a liberar
- d) Conviene completarla junto con la definición de la arquitectura, para tener claro cuáles son los principales componentes que se van a liberar.

---

17) Respecto al mantenimiento

- a) Conviene establecer un calendario de liberación de versiones, para poder evaluar de manera adecuada los atributos de calidad del resultado de las modificaciones
- b) (a) y conviene conformar un Comité de Control de Cambios que reúna la visión de Cliente, Usuarios y Desarrolladores, responsable por evaluar, priorizar y autorizar los cambios al sistema en explotación
- c) **(b) y frecuentemente el análisis de impacto insume buena parte del esfuerzo total aplicado a cada cambio**
- d) Conviene que los desarrolladores atiendan directamente los requerimientos de los usuarios, realicen los cambios y los pongan en explotación tan pronto como les resulte posible

---

18) La construcción y refinamiento de un modelo de calidad de un producto incluye, según la propuesta de R. Geoff Dromey, los siguientes pasos:

1. identificar los atributos de calidad de alto nivel que describen las necesidades prioritarias de la aplicación
2. identificar las componentes del producto
3. identificar y clasificar las propiedades de calidad más tangibles y significativas de cada componente
4. proponer un conjunto de axiomas para vincular propiedades y atributos
5. evaluar el modelo, identificar sus debilidades, y refinarlo o desecharlo y comenzar nuevamente

Esta forma de trabajo permite:

- a) Establecer un criterio para incluir determinada propiedad que permita su verificación
- b) (a) y que el foco se desplace según los productos y los proyectos
- c) **(b) y construir modelos de calidad para los productos clave de la construcción de software: especificación de requerimientos, diseño, implementación**
- d) (c) y obtener un modelo de calidad consistente con el de ISO/IEC 9126

---

19) El análisis post-mortem de un proyecto:

- a) se lleva a cabo en aquellos casos en los que un proyecto se cancela de forma prematura, sin alcanzar su objetivo
- b) conviene realizarlo sistemáticamente al fin de cada proyecto para evaluar qué anduvo bien, mal e identificar lecciones aprendidas
- c) **(b) es un componente esencial de la mejora del proceso**
- d) (b) y en general conviene realizarlo pasado un tiempo de la finalización para tener datos del resultado en explotación, sin lo cual es aventurado evaluar los resultados

---

20) En la mejora de procesos

- a) Una buena práctica consiste en fijar líneas base o de referencia, por ejemplo de productividad, densidad de defectos
- b) (a) y plantear objetivos de mejora relacionados con obtener mejoras en determinados valores respecto a esas líneas base
- c) **(b) y cumplir los objetivos de mejora significa una mejora en la capacidad del proceso**
- d) la aplicación de un proceso maduro garantiza la obtención de resultados idénticos al ser aplicado repetidas veces en diversos proyectos y contextos

---

21) Ejercicio de Diseño y Verificación (20 puntos)

Se planea construir un sistema que recibe como entrada una lista de números enteros positivos (desde el cero inclusive hasta el cien inclusive) escritos alfabéticamente y devuelve la lista de los números primos que estaban en la lista de entrada. La lista resultante no contiene números primos repetidos y se devuelve escrito alfabéticamente pero al revés. Además, la lista resultante está ordenada de menor a mayor (según el número que representa).

Ejemplo:

Si la entrada es: {"ocho", "once", "ocho", "cuatro", "pepe", "34", "tres", "once"}

Se espera que la salida sea: {"sert", "ecno"}

- a) Suponiendo que se usa el patrón MVC, diseñar la parte del Model del sistema usando tubos y filtros. Describir qué recibe y qué devuelve cada uno de los filtros y en qué orden deben ser ejecutados para cumplir con los requerimientos. Todos los filtros tienen como entrada un archivo de texto y como salida devuelven un archivo de texto. El primer filtro tiene como entrada el archivo de texto que representa la lista de números ingresada por el

usuario del sistema, donde cada línea del archivo representa un número. El último filtro debe tener como salida el resultado esperado, donde cada línea es la representación de cada número primo al reverso. En el ejemplo esta salida corresponde a un archivo de texto con dos líneas: sert y eno.

La ejecución coordinada y concurrente de los filtros queda por fuera de lo que hay que diseñar.

- b) Independientemente de si el filtro está o no en la solución de la parte a) brindar las clases de equivalencia necesarias para probar satisfactoriamente un filtro que recibe un archivo de texto con números (uno por línea) escritos en texto y devuelve un archivo de texto con los números en formato número. En caso de ser necesario se puede complementar la especificación de dicho filtro. Por motivos de costos de las pruebas no puede haber más de 20 clases de equivalencia.
- c) Considerando que se agrega ahora el requerimiento que la lista de números de entrada también puede ser numérica (y no en texto) proponga cambios al diseño descrito en a). En caso de aparecer nuevos filtros se deben describir.
- d) Proponga una estrategia de integración (dentro de las vistas en el curso) de todo el sistema considerando las siguientes partes: Model, View y Controller. Para esta versión del producto hay tres View distintas. Se debe aclarar si con la estrategia elegida se tienen que usar stubs y/o drivers. Se debe considerar una estrategia de integración también dentro del Model, es decir, para los tubos y filtros.

Para simplificar el ejercicio se asume que las letras alfabéticas son sólo minúsculas.

---

## 22) Ejercicio de WBS

El ministerio XX desea unificar e interconectar el software que soporta el funcionamiento de tres secciones, que actualmente están utilizando un software con tecnología obsoleta y que no cubre todas sus necesidades. El jefe del sector informática ha decidido tercerizar este desarrollo y ha iniciado conversaciones con la empresa que Ud. dirige para que lleve a cabo la tarea.

No existe una especificación de requerimientos, el relevamiento de los mismos forma parte del proyecto encomendado.

La mayoría de los funcionarios de las tres secciones están descontentos con el software actual.

Algunas secciones no lo usan porque dicen que “no les sirve”. Ud. indaga y se entera de que es porque otra de las secciones no ingresa los datos que debería y éstos no quedan registrados en la base. Actualmente se pasan datos en papel.

El director del sector informático sabe lo que quiere, pero hace poco que ingresó al ministerio y no cuenta con demasiado ascendiente sobre los funcionarios ni sobre sus superiores.

Los usuarios de cada sección quieren que el nuevo software soporte toda la operativa de su sección.

Algunos usuarios son muy particulares, otros desconfiados y no saben si se van a adaptar a un software y una forma de trabajo diferente. El conocimiento de informática de los distintos usuarios es muy variado.

- a) Haga la lista de riesgos que presenta este proyecto y proponga una estrategia de reducción o mitigación, o un plan de contingencia (indicando de qué se trata en cada caso) para cada uno.
- b) Realice un WBS de las actividades necesarias para llevar esta tarea a buen puerto.