**Ejercicio 1:**

Marque con una cruz la respuesta correcta. Valor total 22 puntos, cada respuesta incorrecta resta 1,1 puntos.

1. El/Los uso/s fundamental/es de las métricas del proyecto es/son: (+1)
   1. Optimizar la planificación.
   2. Evaluar la calidad de los productos en un momento determinado.
   3. Para asegurar que el proceso definido está siendo considerado.
   4. Para implementar mejoras de manera de lograr los objetivos de negocio.
   5. Todos los anteriores.
2. ¿Cuándo ocurre la primera aplicación de métricas de proyecto en la mayoría de los proyectos de desarrollo de software? (1)
   1. Durante el análisis del dominio.
   2. Durante la documentación de la especificación de requerimientos.
   3. Durante la estimación
   4. Durante las entrevistas con los clientes/stakeholders.
   5. Ninguna de las anteriores.(desde la planificación)
3. El costo de la calidad en software es: (1) Ni puta idea
   1. Insignificante comparado con el costo de programación.
   2. Igual o a lo sumo muy poco más costoso que el relevamiento.
   3. Proporcionalmente superior al resto de las actividades de software.(La vi y puse ésta de una, pero ni idea por qué)
   4. Un cuarto del presupuesto total de cualquier proyecto de software.
4. El estándar IEEE/EIA 12207, considera las actividades de V&V como parte de: (1)
   1. Proceso de Desarrollo.
   2. Proceso de Gestión.
   3. Proceso de Mejora.
   4. Proceso de Aseguramiento de Calidad.
   5. Proceso de Resolución de Problemas
5. Los errores en el software, son introducidos por: (1)
   1. Los seres humanos que desarrollan software.
   2. Los seres humanos y las computadoras utilizadas para desarrollar software.
   3. Los seres humanos, las computadoras y el software utilizado para programar.
   4. Los seres humanos que interactúan con el software desarrollado.
6. El Plan de Aceptación del Software debe generarse una vez finalizado: (1)
   1. El análisis de requisitos.
   2. El Diseño del Sistema.
   3. La programación del sistema.
   4. En cualquier momento.
7. Un caso de prueba está conformado por: (1)
   1. Las entradas propuestas y las salidas esperadas.
   2. Las entradas propuestas, las salidas esperadas, y el resultado de las pruebas.
   3. Las entradas propuestas, las salidas esperadas, y la definición de lo que se está probando.
   4. Las entradas propuestas, las salidas esperadas, la definición de lo que se está probando, el resultado de las pruebas.
8. Si un producto de software pasa sus procesos de Verificación (no se encuentran más defectos), entonces podemos garantizar que además es Válido (no es necesario Validarlo).
   1. Verdadero.
   2. Falso.
9. Los modelos prescriptivos de proceso se denominan de esa manera porque prescriben: (1)
   1. Actividades del marco de trabajo.
   2. Acciones de ingeniería de software.
   3. Tareas y productos de trabajo.
   4. Aseguramiento de calidad
   5. Mecanismos de control de cambios.
   6. Todas las anteriores.

“Prescriptivos” porque prescriben un conjunto de elementos del proceso: actividades estructurales, acciones de ingeniería de software, tareas, productos del trabajo, aseguramiento de la calidad y mecanismos de control del cambio para cada proyecto. Pressman

1. “Al tener en cuenta fuertemente los procesos de prueba, el Ciclo de Vida de Desarrollo de Software a utilizar siempre tiene que ser una variante del ‘Modelo en V’”.
   1. Verdadero.
   2. Falso.

**Ejercicio 2:**

Un buen software debe tener los siguientes atributos: Mantenibilidad (o Mantenimiento), Confiabilidad, Eficiencia y Usabilidad (Aceptabilidad). (25 puntos)

1. Describa brevemente estos atributos.

* **Mantenimiento**: El software debe escribirse de tal forma que pueda evolucionar para satisfacer las necesidades cambiantes de los clientes. Éste es un atributo crítico porque el cambio del software es un requerimiento inevitable de un entorno empresarial variable.
* **Confiabilidad y seguridad**: La confiabilidad del software incluye un rango de características que abarcan fiabilidad, seguridad y protección. El software confiable no tiene que causar daño físico ni económico, en caso de falla del sistema. Los usuarios malintencionados no deben tener posibilidad de acceder al sistema o dañarlo.
* **Eficiencia**: El software no tiene que desperdiciar los recursos del sistema, como la memoria y los ciclos del procesador. Por lo tanto, la eficiencia incluye capacidad de respuesta, tiempo de procesamiento, utilización de memoria, etcétera.
* **Aceptabilidad**: El software debe ser aceptable al tipo de usuarios para quienes se diseña. Esto significa que necesita ser comprensible, utilizable y compatible con otros sistemas que ellos usan.

1. Defina al menos una métrica para cada uno de estos atributos, que le permita evaluar el desempeño del software construido.

* **Mantenimiento**: TMC (tiempo medio de cambio)
* **Confiabilidad**: tiempo medio entre fallas/caídas, número medio de fallas por día (semana, mes, año).
* **Eficiencia**: uso medio de memoria, tiempo medio de respuesta, número de medio de transacciones por hora.
* **Aceptabilidad**: tiempo medio requerido para ser eficiente en la utilización del sistema, número medio de cuadros de ayuda por pantalla gráfica presentada al usuario.

**Ejercicio 3:**

La IEEE define a la Ingeniería de Software como “la aplicación de un enfoque sistemático, disciplinado y cuantificable al -desarrollo, operación y mantenimiento del software”. Explique esta definición. (18 puntos)

Esta definición refleja cómo la idea base de la ingeniería se aplica a la “industria” del software. Mediante un enfoque sistemática y disciplinado permite que *todo* el proceso de desarrollo de software pueda realizarse siguiendo lineamientos básicos que permiten guiar esta actividad dentro del marco de cada proyecto y organización. Además, mediante mediciones cuantificables otorga un enfoque más *objetivo* de cómo se está haciendo cada una de las actividades involucradas para luego, en base a parámetros y criterios preestablecidos, poder realizar mejoras que contribuyan con el objetivo principal de la industria del software: desarrollar sistemas que cumplan con los requisitos planteados, bajo los cotos estimados/acordados y en el tiempo pactado.

**Ejercicio 4:**

1. Dados los siguientes requerimientos, verifique si son no-ambiguos, verificables, claros (concisos), correctos, atómicos, independientes, abstractos (libres de implementación), consistentes, necesarios y no redundantes. (17 puntos)
2. En el caso de no cumplir con alguna de estas características, se le solicita que lo reescriba para que estos requerimientos cumplan con las características de una buena definición. (18 puntos)

* Req. 1: El sistema deberá implementarse usando ASP.
  + Requerimiento organizacional enfocado posiblemente en la *usabilidad* y derivado de políticas y certificaciones de calidad.
  + Nota: si no existe nada de eso, sería un requerimiento no abstracto y no necesario.
* Req. 2: El sistema no soportará el ingreso de contraseñas con menos de 8 caracteres y más de 15 caracteres.
  + **No** sería **necesario**.
  + Si el cliente lo pediría explícitamente sería tenido en cuenta como un requerimiento no funcional.
  + “El sistema debería validar que las contraseñas contengan entre 8 y 15 caracteres”.
* Req. 3: En la pantalla de "Vuelos almacenados", el usuario sólo puede ver un registro.
  + Es una **restricción**.
* Req. 4: La facilidad de búsqueda deberá permitir que el usuario pueda encontrar una reserva basada en su apellido, fecha, ciudad destino, etc.
  + **Incompleto**, **incorrecto**.
  + “El sistema debe permitir buscar reservas”.
  + “El sistema debe permitir buscar reservas mediante apellido, fecha y cuidad destino”.
* Req. 5: El código del aeropuerto debe ser ingresado por el usuario.
  + **No** es **necesario**.
  + **Ambiguo** (el código se ingresa para realizar una búsqueda, el código es necesario para comprar un pasaje).
  + Eliminar.
* Req. 6: Algunas veces los usuarios ingresarán el código del aeropuerto, el cual el sistema deberá entenderlos, pero algunas veces la ciudad más cercana al aeropuerto puede reemplazarlo, por lo que el usuario no necesita conocer qué código aeropuerto sino sólo la ciudad, y el sistema deberá entenderlo.
  + Es **ambiguo** y **no** es **conciso** (**no claro**).
  + Se debe relevar nuevamente para conocer qué es lo que se requiere realmente.
* Req. 7 (en un sistema para el alquiler de autos a nivel nacional): Los precios de alquiler de los autos deberán mostrarse con todas las tasas impositivas (incluyendo el 2,5% de ingresos brutos de la provincia).
  + Me parece **ambiguo**. ¿Todas las provincias tienen el mismo porcentaje de Ingresos Brutos? ¿O se refiere a un caso particular de una provincia?
  + **Incompleto** (¿Cuáles son las tasas impositivas? ¿De qué país?)
* Req. 8a: La lista de vuelos disponibles deberá incluir la fecha de vuelo, el nro. y el horario de arribo.
* Req. 8b: Éstos deberían ser ordenado por precios.
  + **Ambiguo** (siempre ordenados por precio o podría ser una opción adicional).
  + Eliminar.
  + “El sistema debería permitir ordenar los vuelos según el precio”.
* Req. 9: El sistema deberá proveer la funcionalidad para reservar vuelos, comprar los pasajes, reservar un hotel, reservar un auto y proveer también información sobre atracciones turísticas.
  + **No** es **atómico**.
  + Deben separarse en varios requerimientos.
* Req. 10: Todos los requerimientos especificados deben ser implementados.
  + No es **necesario**.
  + No es una función del sistema.
  + Eliminar.
* Req. 11: La información debe almacenarse en archivos de texto plano.
  + **No** es **abstracto**.
  + Dice cómo y no qué.
  + Eliminar.
* Req. 12a: Las fechas deberán mostrarse en formato DD/MM/AAAA.
* Req. 12b: Para los vuelos a Estados Unidos, las fechas deberán mostrarse en formato MM/DD/AAAA.
  + **No** es **consistente** (se contradicen con el anterior).
  + 12a podría ser un requerimiento no funcional del sistema enfocado en la *usabilidad* (puede que interactúe con otro sistema ya existente que maneja las fechas con ese formato).
  + Eliminar 12b.
* Req. 13a: Pagos por PayPal deberán estar disponibles.
  + Se debería redactar diferente “El sistema debería permitir pagos mediante PayPal.”
* Req. 13b: Sólo se aceptan pagos con tarjeta de crédito internacionales.
  + **No** es **consistente**. (contradice al anterior; ¿solo tarjetas de crédito internacionales o PayPal (método de pago)?)
* Req. 14a: El sistema deberá desplegar un calendario pop-up cada vez que se necesite ingresar una fecha.
* Req. 14b: Un calendario debería estar disponible para ayudar al ingreso de la fecha de vuelo.
  + Son **redundantes**. El 14a **no** es **abstracto**.
  + El 14b, como está, estaría bien. Eliminar 14a.
* Req. 15a: El país de destino no necesita ser visualizado para vuelos dentro de Argentina.
* Req. 15b: Para todos los vuelos el sistema mostrará el país de destino.
  + **No** son **consistentes**.
  + Volver a consultar para saber cuál es el requerimiento real.

**Ejercicio 5:**

Es usted el jefe de un proyecto de desarrollo de un sistema satelital en el cual ha estimado tener 30 KLDC. El software a desarrollar es de alta complejidad, pero afortunadamente podremos contar con personal de muy alta calificación y experiencia específica en el tipo de software a desarrollar. Teniendo en cuenta que el costo del salario mensual promedio de cada persona es de $ 20500 mensuales:

1. Calcule el esfuerzo de desarrollo de este sistema, tiempo, personal y costo teniendo en cuenta los valores dados.
2. Para el caso anterior, ¿sería más rentable en tiempo y dinero, emplear personas de nivel medio cuyo salario es de $ 13500 mensuales? Justifique su respuesta.

Nota: En este proyecto podemos considerar que el valor del FAE es:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Proyecto software | **a** | **e** | **c** | **d** |
| **Orgánico** | 3.2 | 1.05 | 2.5 | 0.38 |
| **Semi-acoplado** | 3.0 | 1.12 | 2.5 | 0.35 |
| **Empotrado** | 2.8 | 1.2 | 2.5 | 0.32 |



1. Dice personas de nivel medio, no que tengan menor experiencia. Podría calcularse los valores con