

Capítulo 6 “Análisis de los requerimientos de información Parte II”

1. ¿Cuáles son los cuatro tipos de información que busca el analista en la elaboración de prototipos?

R.- Las reacciones del usuario hacia el nuevo sistema, las sugerencias, las innovaciones, y la revisión planeada.

2. ¿Qué significa el término *prototipo corregido*?

R.- Se refiere a un sistema que funciona pero que se corrige simultáneamente, y requiere que se optimicen la mayoría de sus componentes.

3. Defina un prototipo que es un modelo a escala no funcional.

R.- Es un prototipo en el que posiblemente se incluyan las entradas y salidas, pero no se incluye el procesamiento de los datos, ya que esta codificación es demasiado extensa o costosa.

4. Proporcione un ejemplo de un prototipo que es un primer modelo a escala completa.

R.- El prototipo de un nuevo modelo de automóvil.

5. Defina lo que significa un prototipo que es un modelo con algunas, pero no todas, las características principales.

R.- Es un prototipo donde se presentan al usuario final, algunas de las características principales del sistema, y las mismas se mantienen a lo largo del desarrollo hasta llegar a implementar el 100% de las características o funciones.

6. Haga una lista de las ventajas y desventajas de usar la elaboración de prototipos para *reemplazar* el ciclo de vida del desarrollo tradicional de sistemas.

R.- Algunas de las ventajas son:

- **Se reduce el tiempo entre el levantamiento de requerimientos y la entrega de un sistema funcional.**
- **Ayuda a identificar con mayor precisión los requerimientos del usuario.**

Alguna desventajas:

- **Riesgo de desarrollar un sistema antes de poder entender totalmente los problemas o requerimientos.**
- **El sistema desarrollado podría ser adecuado para un grupo específico de personas, pero inadecuado para las necesidades más globales.**

7. Describa cómo se puede usar la elaboración de prototipos para aumentar el ciclo de vida del desarrollo tradicional de sistemas.

R.- La elaboración de prototipos puede utilizarse como un método adicional y especializado para ayudar a la etapa de levantamiento de requerimientos.

8. ¿Cuáles son los criterios para decidir si se debe hacer un prototipo de un sistema?

R.- Se deben estimar los costos asociados al desarrollo del prototipo. Si los costos de los programadores, analistas y del equipo quedan dentro del presupuesto, entonces se puede proceder con la elaboración del prototipo.

9. Mencione cuatro lineamientos que el analista debe observar en el desarrollo de un prototipo.

R.- Los cuatro lineamientos a considerar son:

- Trabajar en módulos manejables.
- Construir rápidamente el prototipo.
- Modificar el prototipo en iteraciones sucesivas.
- Poner énfasis en la interfaz de usuario.

10. ¿Cuáles son los dos problemas principales identificados en la elaboración de prototipos?

R.- Por un lado, es difícil manejar la elaboración de prototipos como un proyecto dentro del desarrollo de sistemas más grandes. Y por otro lado, debe evitarse que los usuarios finales tomen al prototipo como si fuera un sistema final.

11. Mencione las tres ventajas principales de utilizar la elaboración de prototipos.

R.- Las tres ventajas principales son:

- Posibilidad de modificar el sistema en las etapas iniciales del desarrollo
- Oportunidad de suspender el desarrollo de un sistema que no es funcional
- Posibilidad de desarrollar un sistema que se aproxime más a satisfacer las necesidades y expectativas de los usuarios.

12. ¿Cómo puede un prototipo de un sitio Web interactivo facilitar el proceso de la elaboración de prototipos? Conteste en un párrafo.

R.- Al permitir a los usuarios finales utilizar en cualquier momento el sistema, incluso desde ubicaciones remotas, y capturar inmediatamente sus observaciones o sugerencias sobre el mismo, en algún formulario integrado en la plataforma de pruebas.

13. ¿Cuáles son las tres formas en que un usuario puede ser de ayuda en el proceso de la elaboración de prototipos?

R.- Las tres formas en que un usuario colabora en la elaboración del prototipo son:

- Experimentando con el prototipo.
- Dando reacciones sinceras sobre el prototipo.
- Sugiriendo adiciones o eliminaciones al prototipo.

14. Defina lo que significa RAD.

R.- El desarrollo rápido de aplicaciones (RAD) es un enfoque orientado a objetos para el desarrollo de sistemas que incluye un método de desarrollo, así como herramientas de software.

15. ¿Cuáles son las tres fases del RAD?

R.- Las tres fases son:

- Planeación de requerimientos
- Taller de diseño del RAD
- Implementación

16. Defina la programación extrema.

R.- La programación extrema es un enfoque de desarrollo de software que toma lo que generalmente conocemos como “prácticas de desarrollo de software aceptables” y las lleva al extremo.

17. ¿Cuáles son los cuatro valores que deben compartir el equipo de desarrollo y los clientes de negocios cuando se toma un enfoque de programación extrema?

R.- Los cuatro valores son la comunicación, la simpleza, la retroalimentación, y la valentía.

18. ¿Cuáles son los cinco principios básicos de la programación extrema?

R.- Los cinco principios son: proporcionar una retroalimentación rápida, adoptar la sencillez, cambiar progresivamente, aceptar el cambio, y alentar un trabajo de calidad.

19. ¿Cuáles son las cuatro prácticas principales del enfoque de desarrollo de XP?

R.- Las cuatro prácticas principales son:

- Liberación limitada
- Semana de trabajo de 40 horas
- Tener al cliente en el sitio
- Filosofía de programación en parejas

20. Delinee los pasos típicos en un episodio de desarrollo de XP.

R.- Los pasos típicos en el desarrollo de XP son la exploración, la planeación, las iteraciones a la primera versión, la puesta en producción, y el mantenimiento.

21. ¿Qué es una historia de usuario? ¿Es principalmente escrita o hablada? Elija su opción, luego apoye su respuesta con un ejemplo.

R.- Una historia de usuario es un listado de las diferentes acciones que son posibles en un escenario específico. El desarrollo de la misma es principalmente hablada entre el desarrollador y el usuario, pero debe quedar escrita para usarla de referencia en el desarrollo del sistema.

22. Mencione las herramientas de software que pueden ayudar al desarrollador a hacer una variedad de pruebas de código.

R.-

- Pruebas unitarias de código: SUnit y Junit.
- Probadores unitarios automatizados, probadores de aceptación y probadores de GUI: JUnit, ComUnit, VUnit, NUnit, httpUnit y Rational Visual Test Tools.
- Medición del sistema y desempeño de componentes: Jmeter, JUnitPerf, PerfMon, TrueTime, RealTime y Microsoft Visual Studio Analyzer.
- Control del código fuente: CVS, Visual Source Safe y PVCS.
- Entornos de desarrollo: IBM VisualAge, Microsoft Visual Studio .NET y JBuilder.

23. ¿Cuáles son las seis lecciones tomadas de la experiencia con los esfuerzos del desarrollo de XP?

R.- Las seis lecciones importantes son:

- La liberación limitada permite que los sistemas evolucionen.
- La programación en parejas incrementa la calidad global.

- Los clientes en el sitio son beneficiosos tanto para el negocio como para el equipo de XP.
- La semana de trabajo de 40 horas mejora la eficiencia.
- Los recursos y actividades equilibrados dan soporte a los objetivos del proyecto.
- Los valores de XP (comunicación, sencillez, retroalimentación, y valentía) son importantes para su éxito.

24. Compare y contraste el modelado ágil con el enfoque de XP.

R.- El modelado ágil tiene un enfoque muy similar a la programación extrema, con el elemento adicional de que agrega la “humildad” a los valores. También agrega algunos principios como “modelar con un propósito”, “el software es su meta principal” y “viajar con poco equipaje”.

25. ¿Qué es melé?

R.- Melé es un término asociado al modelado ágil. Se refiere principalmente al trabajo en equipo y las actividades que son requeridas para realizar este tipo de desarrollo.