

Universidad de las Fuerzas Armadas ESPE

Fundamentos de Software

NRC 1951

Integrantes: Gerald Astudillo, Henry Chalcualan, Lenin Erazo, Henry Suin.

Actividad: responder las siguientes preguntas utilizando referencias

1. La clasificación de las metodologías de desarrollo de software es:

Estructuradas/Orientadas a objetos.

Orientadas a procesos/Orientadas a datos.

Mixtas.

Justificación: La clasificación "Estructuradas/Orientadas a objetos" agrupa dos enfoques fundamentales en el desarrollo de software. Las metodologías estructuradas siguen un proceso secuencial y predefinido de fases (planificación, análisis, diseño, etc.), mientras que las orientadas a objetos se centran en la modularidad y reutilización, utilizando objetos que combinan datos y métodos. Ambas metodologías representan diferentes paradigmas en la creación de software, lo que justifica su clasificación separada. (Sommerville, 2016)

2. Las metodologías orientadas a procesos se fundamentan en:

El modelo básico de entrada-salida.

El modelo básico del proceso.

El modelo básico de entrada-proceso-salida.

Justificación: Las metodologías orientadas a procesos se fundamentan en el modelo básico de entrada-proceso-salida, ya que este modelo refleja cómo los procesos de software reciben entradas (información, datos), las procesan a través de actividades específicas, y generan salidas (resultados o productos). Este enfoque ayuda a estructurar y gestionar el flujo de trabajo en el desarrollo de software, garantizando una producción eficiente y controlada. (Sommerville, 2016)

3. El proceso unificado de desarrollo se basa en características que son:

Centrado en el diseño/Centrado en la arquitectura/Dirigido por casos de uso.

Centrado en la arquitectura/Dirigido por casos de uso/Iterativo e incremental.

Centrado en la arquitectura/Dirigido por casos de uso/Exige poca documentación.

Justificación: El proceso unificado de desarrollo se basa en características como: centrado en la arquitectura, dirigido por casos de uso, e iterativo e incremental. Este enfoque se enfoca en crear una arquitectura sólida desde el principio, utiliza casos de uso para definir los requisitos y las funcionalidades del sistema, y sigue un ciclo iterativo e incremental que permite desarrollar y mejorar el software de forma continua, adaptándose a cambios y retroalimentación constante. (Kruchten, P. 2004)

4. Las metodologías orientadas a datos se concentran en:

Entrada-salida.

Actividades de análisis.

Procesos de entrada y salidas.

Justificación: Las metodologías orientadas a datos se concentran en las actividades de análisis, ya que su enfoque principal es modelar, analizar y transformar los datos a lo largo del ciclo de vida del software. El análisis de datos es fundamental para entender cómo se estructuran y manipulan los datos en el sistema, lo que permite una gestión eficiente de la información (Avison, D., & Fitzgerald, G., 2006)

5. Las metodologías orientadas a objetos se fundamentan en:

Datos y procesos.

Procesos de entrada y salidas.

Conjunto de objetos.

Justificación: Las metodologías orientadas a objetos se fundamentan en el conjunto de objetos, ya que este enfoque organiza el software en unidades autónomas llamadas objetos, que combinan tanto datos como los procesos (métodos) que operan sobre esos datos. Este modelo promueve la reutilización y

modularidad, elementos clave en el desarrollo de software eficiente y flexible (Booch, G., Rumbaugh, J., & Jacobson, I., 2005)

6. El diseño y la implementación de las metodologías orientadas a objetos está centrado en:

El comportamiento y la estructura.

Los procesos y los datos.

Los objetos.

Justificación: El diseño y la implementación de las metodologías orientadas a objetos están centrados en los objetos, ya que estos son la unidad fundamental de este enfoque. Los objetos encapsulan tanto los datos como los métodos que operan sobre esos datos, permitiendo una mayor modularidad, reutilización y mantenimiento eficiente del software. (Booch, G., Rumbaugh, J., & Jacobson, I., 2005)

7. Las metodologías ágiles se clasifican en:

Agile Unified Process-AUP.

Rational Unified Process-RUP.

Metodología de Edward Yourdon.

Justificación: Las metodologías ágiles se clasifican en *Agile Unified Process (AUP)*, ya que esta es una adaptación ágil del Rational Unified Process (RUP), enfocada en la flexibilidad, la colaboración continua con el cliente y la entrega incremental de software funcional. AUP mantiene los principios de RUP pero con un enfoque más ágil, permitiendo ciclos más cortos de desarrollo. (Ambler, 2005)

8. Las metodologías orientadas a datos se clasifican en:

Jackson Structured Programming-JSP-AUP.

Jackson Structured Design –JSD-RUP.

Desarrollo de sistemas estructurados-Warnierr-Orr.

Justificación: Las metodologías orientadas a datos se clasifican en *Desarrollo de sistemas estructurados-Warnier-Orr*, ya que esta metodología pone énfasis en

el análisis y diseño de los datos en el sistema, utilizando un enfoque estructurado para la gestión y manipulación de los mismos. Esta clasificación se basa en el procesamiento de la información de manera sistemática y organizada, lo cual es característico de las metodologías orientadas a datos. (Warnier & Orr, 1975)

9. Las metodologías ágiles más conocidas son:

OMT, OOSE, PU.

XP, SCRUM, AUP.

Método de BOOCH.

Justificación: Las metodologías ágiles más conocidas incluyen Scrum, Extreme Programming (XP) y Agile Unified Process (AUP). Estas metodologías se destacan por su enfoque en la flexibilidad, la colaboración y la entrega incremental de productos de software. (Monserrate, 2024)

10. Las metodologías orientadas a objetos se clasifican en:

Object Modelling Technique-OMT/OOSE/PU/BOOCH.

Jackson Structured Design –JSD-RUP.

Metodología de Edward Yourdon

Justificación: Las metodologías orientadas a objetos incluyen técnicas como la Object Modeling Technique (OMT), la Object-Oriented Software Engineering (OOSE) y el método Booch. Estas metodologías fueron desarrolladas por expertos en ingeniería de software: James Rumbaugh, Ivar Jacobson desarrolló y Grady Booch (Universidad Abierta y a Distancia de México, 2024)

Referencias

Sommerville, I. (2016). *Ingeniería de software* (10.^a ed.). Pearson Educación.

Kruchten, P. (2004). *The Rational Unified Process: An introduction* (3.^a ed.). Addison-Wesley.

Avison, D., & Fitzgerald, G. (2006). *Information systems development: Methodologies, techniques and tools* (4.^a ed.). McGraw-Hill.

Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *The unified modeling language user guide* (2.^a ed.). Addison-Wesley.

Warnier, J. D., & Orr, W. A. (1975). *Structured systems analysis: An approach to software development*. McGraw-Hill.

Universidad Abierta y a Distancia de México. (01 de 10 de 2024). *unadmexico*. Obtenido de Desarrollo de Software 2do semestre Programa de la unidad didáctica:

https://dmd.unadmexico.mx/contenidos/DCEIT/BLOQUE1/DS/02/DDOO/U3/descargables/DDOO_Unidad_3.pdf?utm_source=chatgpt.com

Monserate, M. (31 de marzo de 2024). *axialcrm*. Obtenido de Metodologías ágiles: conoce Scrum, Lean, Kanban, XP y más:

<https://www.axialcrm.com/es/soporte-y-recursos/que-es-metodologia-agile/>