



2. Diseño

2.1 Diseño de procesos propuestos.

2.1.1 Herramientas CASE para diseño.

2.2 Diseño arquitectónico.

2.3 Diseño de datos.

2.4 Diseño de interfaz de usuario.

El diseño de procesos se refiere a la fase en la que se establecen los procedimientos y pasos que se seguirán para crear un sistema. Esto incluye la identificación de requerimientos, la arquitectura del sistema, la planificación de pruebas, la gestión de cambios y la documentación de procesos.

Desempeña un papel fundamental en la creación de procesos automatizados con calidad, la gestión de proyectos exitosa y el cumplimiento de los objetivos del proyecto.

Nota: Adaptarse a las necesidades cambiantes de la industria y la tecnología es esencial para mantenerse relevante en este campo en constante evolución.



Importancia

- **Eficiencia:** Un diseño de procesos bien definido puede mejorar la eficiencia en el desarrollo de la ingeniería de procesos al minimizar la duplicación de esfuerzos y la ambigüedad en las tareas.
- **Calidad:** Un diseño de procesos adecuado permite establecer estrategias con base en mejores prácticas en la ingeniería de procesos y pruebas.
- **Gestión de riesgos:** Los procesos bien diseñados pueden ayudar a identificar y mitigar riesgos tempranamente en el ciclo de ingeniería de procesos.
- **Reproducibilidad:** Un diseño de procesos documentado permite la reproducción de resultados en proyectos futuros.

Métodos y enfoques comunes en el diseño de procesos:

- **Modelo en cascada:** Divide etapas secuenciales como requerimientos, diseño, implementación, pruebas y mantenimiento.
- **Metodologías ágiles:** Enfoques como **Scrum** y **Kanban** promueven la flexibilidad y la colaboración continua.
- **Modelos de procesos de ciclo de vida:** Como el modelo en espiral que incorpora la iteración y la evaluación continua.

Herramientas para el diseño de procesos:

- **Diagramas de flujo:** Utilizados para representar visualmente los pasos y la secuencia de actividades en un proceso.
- **Aplicaciones de modelado:** Herramientas como Microsoft Visio o Lucidchart se utilizan para crear diagramas de procesos y flujos de trabajo.
- **Gestión de proyectos:** Aplicaciones como Trello, Asana o Jira ayudan en la planificación y seguimiento de procesos.

Desafíos en el diseño de procesos:

- **Cambios constantes:** La naturaleza cambiante de los proyectos puede requerir adaptaciones en los procesos.
- **Alineación con los objetivos del negocio:** Es esencial asegurar que los procesos estén alineados a los requerimientos específicos y estratégicos.
- **Cultura organizacional:** “Paradigmas”, la resistencia cultural puede dificultar la implementación de nuevos procesos.



Evolución del diseño de procesos en ingeniería de procesos:

- Con la creciente adopción de metodologías ágiles, el diseño de procesos es más adaptable y centrado en las personas - roles.
- El enfoque en la automatización y la integración continua ha cambiado la forma en que se diseñan los procesos.

2. Diseño

2.1 Diseño de procesos propuestos.

2.1.1 Herramientas CASE para diseño.

2.2 Diseño arquitectónico.

2.3 Diseño de datos.

2.4 Diseño de interfaz de usuario.



Las herramientas CASE (Computer-Aided Software Engineering) son aplicaciones diseñadas para apoyar y facilitar el desarrollo de la ingeniería automatizada. Estas herramientas abarcan una amplia gama de funcionalidades que incluyen modelado, diseño, análisis, documentación, pruebas y gestión de proyectos.

Proporcionando a los equipos de trabajo las capacidades necesarias para planificar, diseñar, gestionar y mantener sistemas con calidad.

Nota: La elección de la herramienta adecuada dependerá de las necesidades y requerimientos específicos del proyecto.

Desafíos en el uso de herramientas CASE:

- **Curva de aprendizaje:** Algunas herramientas pueden tener una curva de aprendizaje que impacta el costo/tiempo.
- **Integración con otros sistemas:** La integración de herramientas CASE con otras tecnologías.

Evolución de las herramientas CASE:

- Las herramientas CASE han evolucionado para adaptarse a las cambiantes metodologías para la generación de procesos automatizados.
- La tendencia es hacia herramientas colaborativas, basadas en la nube para facilitar el trabajo en equipo y la accesibilidad en cualquier lugar.

Tipos de herramientas CASE:

- **Modelado:** Permiten la creación de diagramas y modelos que representan visualmente el diseño de ingeniería de procesos, como diagramas de flujo, diagramas de clases, diagramas de casos de uso, entre otros diagramas. **Ejemplos Microsoft Visio, Lucidchart y Enterprise Architect.**
- **Análisis y diseño:** Ayudan en la planificación y el diseño de ingeniería de procesos, permitiendo definir la arquitectura, estructura y componentes utilizados en el proyecto. **Ejemplos Rational Rose, Sparx Systems, y Visual Paradigm.**
- **Gestión de requisitos:** Ayudan a capturar, rastrear y gestionar los requisitos a lo largo del ciclo de vida del proyecto. **Ejemplos IBM Engineering Requirements Management DOORS y Jama Connect.**
- **Generación de código:** Permiten generar automáticamente código a partir de modelos y diseños previamente definidos. **Ejemplos Eclipse Modeling Framework (EMF) y CodeCharge Studio.**
- **Generación de pruebas:** Facilitan la planificación, ejecución y seguimiento de pruebas. **Ejemplos HP ALM (Application Lifecycle Management) y TestRail.**
- **Gestión de proyectos:** Ayudan en la planificación, seguimiento y gestión de proyectos. **Ejemplos Jira, Trello, ASANA y Microsoft Project.**



Ventajas de las herramientas CASE:

- **Productividad:** Automatizan tareas manuales.
- **Calidad:** Facilitan la detección temprana de errores y eventualidades orientadas al diseño.
- **Colaboración:** Permiten a los equipos trabajar de manera colaborativa y compartir información de manera más efectiva.
- **Documentación:** Facilitan la generación automática de documentación técnica y de usuario.
- **Gestión de cambios:** Ayudan a gestionar y mantener un registro de versiones.

2. Diseño

2.1 Diseño de procesos propuestos.

2.1.1 Herramientas CASE para diseño.

2.2 Diseño arquitectónico.

2.3 Diseño de datos.

2.4 Diseño de interfaz de usuario.



Definición:

El diseño arquitectónico es el proceso de definir la estructura, componentes, módulos y relaciones del sistema.

En esta etapa, se toman decisiones importantes sobre cómo los diferentes elementos del sistema interactuarán entre sí para lograr los objetivos de manera eficiente y efectiva.

Importancia:

- **Claridad, eficiencia, eficacia y comprensión:** Proporciona una representación visual y estructural del sistema, lo que facilita la interpretación para los involucrados en el proyecto.
- **Colaboración:** Permite que múltiples equipos trabajen en un proyecto, dividiendo las actividades en módulos o componentes específicos.
- **Mantenimiento:** Un diseño bien planificado facilita las actualizaciones y correcciones en el ciclo de vida del proyecto.



Elementos clave:

- **Componentes:** Identificación de los elementos principales que compondrán el sistema.
- **Módulos:** División del sistema en unidades funcionales.
- **Conexiones:** Definición de cómo los componentes se comunicarán entre sí.
- **Patrones de diseño:** Utilización de estrategias arbitradas para resolver eventualidades comunes en el diseño.
- **Estándares y convenciones:** Uso de normas y buenas prácticas.

Métodos y enfoques:

- **Modelado arquitectónico:** Utilización de diagramas de arquitectura, como diagramas de componentes, diagramas de despliegue y diagramas de flujo de datos, para representar la estructura del sistema.
- **Metodologías arquitectónicas:** Ejemplos incluyen la arquitectura en capas, la arquitectura orientada a servicios (SOA) y la arquitectura de microservicios.
- **Principios de diseño:** Cumplir con principios como la separación de validaciones, consistencia y el acoplamiento.



Herramientas:

- **UML (Unified Modeling Language):** Un lenguaje de modelado estándar que incluye una variedad de diagramas para representar la arquitectura.
- **Aplicaciones de modelado:** Herramientas como Enterprise Architect, Visual Paradigm y Lucidchart permiten crear diagramas arquitectónicos.
- **Plataformas de diseño de microservicios:** Herramientas como Kubernetes, Podman, Docker.

Desafíos:

- Cambios en los requisitos.
- Complejidad.
- Alineación con objetivos del proyecto.



Evolución:

- Con la adopción de enfoques ágiles, el diseño arquitectónico tiende a ser más iterativo y adaptable.
- La nube y la computación distribuida han influido en la evolución de las arquitecturas, llevando a arquitecturas escalables y resilientes.

2. Diseño

2.1 Diseño de procesos propuestos.

2.1.1 Herramientas CASE para diseño.

2.2 Diseño arquitectónico.

2.3 Diseño de datos.

2.4 Diseño de interfaz de usuario.



Definición:

El diseño de datos se refiere al proceso de planificar y organizar cómo se capturarán, almacenarán, procesarán y gestionarán los datos en un sistema.

Esto incluye la definición de estructuras de datos, esquemas de bases de datos, modelos de datos, políticas de seguridad y estrategias de acceso a los datos.

Importancia:

- **Eficiencia:** Un diseño de datos sólido puede mejorar el rendimiento de una aplicación, permitiendo una recuperación inmediata de la información.
- **Integridad de los datos:** Ayuda a garantizar que los datos se almacenen de manera precisa y consistente.
- **Escalabilidad:** Un diseño adecuado de datos facilita la expansión del sistema a medida que crece la cantidad de información.
- **Seguridad:** Contribuye a proteger los datos sensibles y a garantizar el cumplimiento de regulaciones de privacidad.
- **Interoperabilidad:** Facilita la integración con otros sistemas y aplicaciones.



Elementos clave:

- **Modelo de datos:** Define la estructura lógica de los datos, incluyendo tablas, relaciones y atributos en el caso de bases de datos relacionales.
- **Esquema de bases de datos:** Describe la organización física de los datos en el almacenamiento, como índices, particiones y fragmentación.
- **Políticas de seguridad:** Establece quién puede acceder a los datos y qué acciones pueden realizar.
- **Optimización de consultas:** Considera cómo se accederán y procesarán los datos de manera eficiente.

Métodos y enfoques:

- **Modelado de datos:** Utilización de técnicas como el Modelo Entidad-Relación (ER), el Modelo Relacional y el Modelo Dimensional para representar la estructura de los datos.
- **Normalización:** Proceso de organizar los datos en tablas para reducir la redundancia y mejorar la integridad de los datos.
- **Denormalización:** Estrategia que se utiliza en ocasiones para mejorar el rendimiento al permitir cierta redundancia controlada.
- **Gestión de versiones de datos:** Implementación de estrategias para controlar y gestionar las versiones de los datos considerando su ciclo de vida.



Herramientas:

- **Sistemas de gestión de bases de datos (DBMS):** Herramientas como Oracle DB, MySQL, Microsoft SQL Server y PostgreSQL proporcionan capacidades de diseño y administración de bases de datos.
- **Herramientas de modelado de datos:** Ejemplos incluyen Erwin, IBM Data Architect y MySQL Workbench.
- **Herramientas de ETL (Extract, Transform, Load):** Utilizadas para transformar y cargar datos desde diversas fuentes a sistemas de almacenamiento.

Desafíos:

- **Cambios en los requisitos.**
- **Mantenimiento:** La gestión continua de la integridad y la calidad de los datos puede ser un desafío a lo largo del tiempo.
- **Escalabilidad:** Diseñar para manejar grandes volúmenes de datos.



Evolución:

- Con la creciente adopción de arquitecturas de microservicios y sistemas distribuidos, el diseño de datos debe considerar la descentralización y la replicación de datos.
- El auge del “big & fast data”, ha llevado a la exploración de nuevas tecnologías y enfoques, como bases de datos NoSQL y sistemas de almacenamiento distribuido.

Referencias

- BARDUS, I. & PRYZESENTSEV, O.. (2023). ANALYSIS OF THE PROFESSIONAL ACTIVITY OF A SPECIALIST IN THE DEVELOPMENT OF USER INTERFACES. Scientific papers of Berdiansk State Pedagogical University Series Pedagogical sciences. 1. 199-209. 10.31494/2412-9208-2023-1-1-199-209.
- Lin, Liannan & Zheng, Zitao & Li, Ziqi. (2023). AI Interaction Design Driven Software Engineering: An Exploratory Experimental Teaching Method. 10.1007/978-981-99-2446-2_24.
- Erazo, Lenin & Suquisupa, Steven & Berneo, Alexandra & Cedillo, Priscila. (2023). Model-Driven Engineering Applied to User Interfaces. A Systematic Literature Review. 10.1007/978-3-031-24985-3_42.
- Khaddam, Iyad & Barakat, Hanaa & Vanderdonckt, Jean. (2016). Enactment of User Interface Development Methods in Software Life Cycles.
- Imam, Ayad & Alnsour, Ayman & Allroob, Aysa. (2015). The Definition of Intelligent Computer Aided Software Engineering (I-CASE) Tools. Journal of Information Engineering and Applications. 5. 47-56.
- Arman, Nabil. (2013). Towards E-CASE Tools for Software Engineering. International Journal of Advanced Corporate Learning (IJAC). 6. 16-19. 10.3991/ijac.v6i1.2309.
- Kosavinta, Satakhun & Kanongchaiyos, Pizzanu & Jinuntuya, Pinyo. (2007). Integration of CAD Software with DSS for Engineering and Architectural Project Design. Computer-Aided Design & Applications. 4. 467-476. 10.1080/16864360.2007.10738566.
- Leventhal, Laura & Mynatt, Barbee. (2006). A scarce resource in undergraduate software engineering courses: User interface design materials. 10.1007/BFb0043599.



Referencias (continuación).

- Thompson, J. & Goh, Angela. (1993). CASE Tools in Software Engineering Education. 319-321. 10.1016/B978-0-444-81597-2.50039-5.
- Bode, Stephan & Fischer, Anja & Kühnhauser, Winfried & Riebisch, Matthias. (2009). Software Architectural Design Meets Security Engineering. Proceedings of the International Symposium and Workshop on Engineering of Computer Based Systems. 109-118. 10.1109/ECBS.2009.17.
- Capilla, Rafael & Ali Babar, Muhammad. (2008). On the Role of Architectural Design Decisions in Software Product Line Engineering. 5292. 10.1007/978-3-540-88030-1_18.
- Kinnula, Atte. (1999). Software process engineering in a multi-site environment: an architectural design of a software process engineering system.
- Six, H.-W & Voss, J.. (1992). A software engineering perspective to the design of a user interface framework. 128 - 134. 10.1109/CMPASAC.1992.217591.
- Bhowmick, Twinkle & Koner, Suraj & Saha, Biraj & Ghosh, Debosree & Pramanik, Bablu. (2023). Software Engineering: New Methodologies, Tools, and Best Practices in Software Development. International Journal of Innovative Research in Science, Engineering and Technology. 12. 10.15680/IJRSET.2023.1208077.