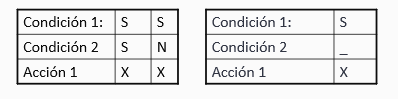
Tablas de Decisión

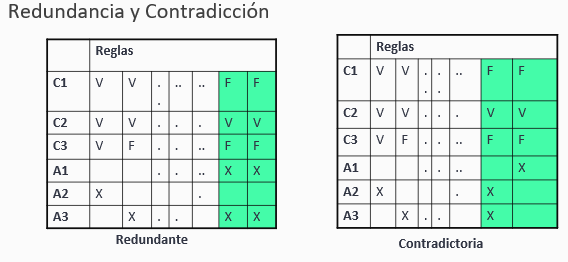
Las tablas de decisión son herramientas para representar reglas lógicas que definen las acciones a tomar según condiciones específicas. Se construyen considerando condiciones simples y acciones simples, expresadas como valores Verdadero o Falso. El proceso implica identificar condiciones, completar la tabla, considerar condiciones opuestas y luego construir reglas.

Las especificaciones pueden ser completas, redundantes o contradictorias. La reducción de complejidad implica combinar reglas similares y eliminar redundancias para simplificar la tabla. El análisis busca eliminar situaciones imposibles, inconsistencias y reducir la tabla al máximo.

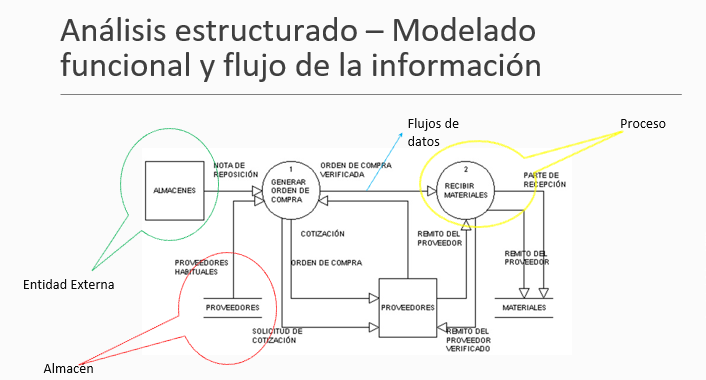


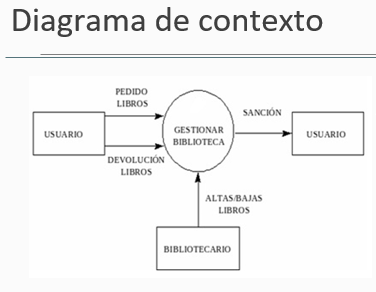
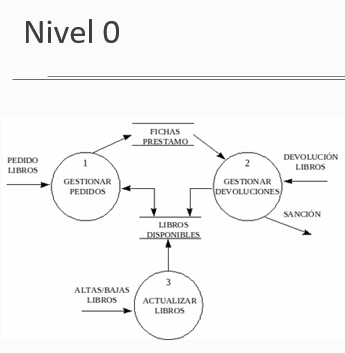


La verificación de la integridad y precisión de las tablas es crucial para evitar problemas como incompletitud, situaciones imposibles, contradicciones y redundancias.



Análisis Estructurado

El análisis estructurado es una metodología que permite comprender los requerimientos de un sistema al identificar cómo se mueven los datos, los procesos que transforman estos datos y sus resultados.

Se basa en el uso de Diagramas de Flujo de Datos (DFD), representaciones visuales que muestran la interacción entre entidades externas, procesos y almacenamientos de datos. El enfoque es jerárquico, comenzando con una lista de actividades de la organización para identificar elementos como entidades externas, flujos de datos, procesos y almacenamientos. Luego se crea un diagrama de contexto que muestra estas interacciones a nivel general, seguido por Diagramas 0 que detallan procesos generales y sus almacenamientos. Posteriormente, se desarrollan diagramas más específicos para cada proceso del Diagrama 0. Esta metodología facilita la comprensión profunda del sistema a construir y su comunicación efectiva a los usuarios.

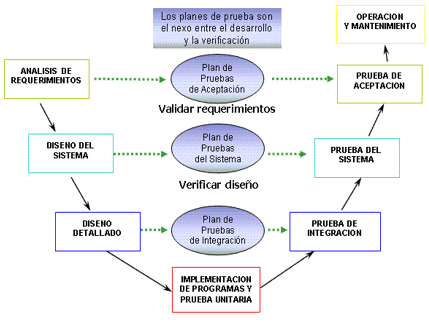


Modelos de proceso

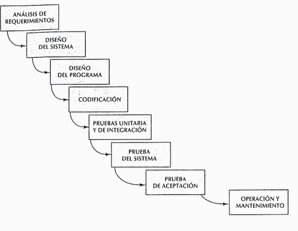
El proceso de software sigue una secuencia de actividades clave: especificación, desarrollo, validación y evolución. La especificación involucra comprender las necesidades y restricciones del sistema, mientras que el desarrollo convierte esas especificaciones en un sistema ejecutable. La validación demuestra que el sistema cumple con las expectativas y requisitos. Por último, la evolución implica el mantenimiento y mejora del software.

Los modelos de proceso de software son representaciones simplificadas de estas actividades y productos, con diferentes enfoques y características. Los modelos prescriptivos definen elementos y flujos de trabajo, mientras que los descriptivos reflejan la realidad del proceso.

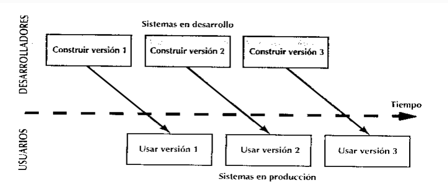
Los ciclos de vida del software describen la vida del producto desde su inicio hasta su implementación, entrega y mantenimiento. Los modelos de proceso pueden ser tradicionales (como el en cascada o en V) o evolutivos (como el en espiral o incremental), cada uno con sus ventajas y desafíos.



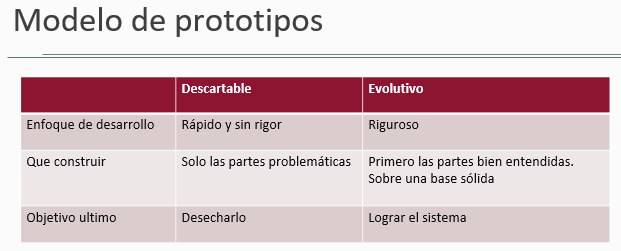
El modelo en cascada sigue etapas secuenciales, pero puede presentar dificultades como la falta de resultados tangibles hasta el final. El modelo en V muestra la relación entre actividades de prueba y diseño. El modelo de prototipos ofrece versiones parciales para comprender mejor el sistema.



Otros modelos como el desarrollo por fases (incremental o iterativo) entregan el sistema por partes o en versiones completas sucesivas. El modelo en espiral combina desarrollo con gestión de riesgos, permitiendo iteraciones y mejoras constantes, pero requiere una evaluación detallada en cada ciclo.



Cada modelo tiene sus ventajas y desafíos, adaptándose a diferentes contextos, necesidades y preferencias de desarrollo.



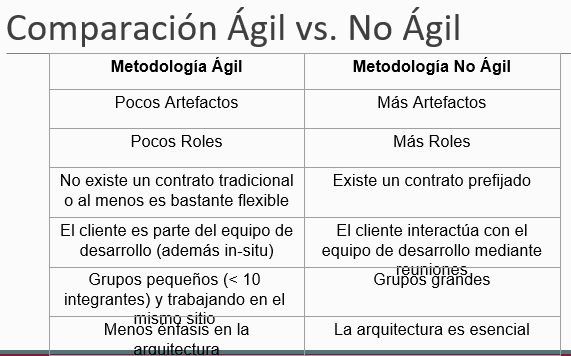
Metodologías agiles

Claro, las metodologías ágiles surgieron como respuesta a las limitaciones y rigidez de los enfoques tradicionales de desarrollo de software. Estos métodos se centran en la flexibilidad, la adaptabilidad a los cambios y la entrega iterativa de software de alta calidad. Algunos de los principios comunes incluyen:

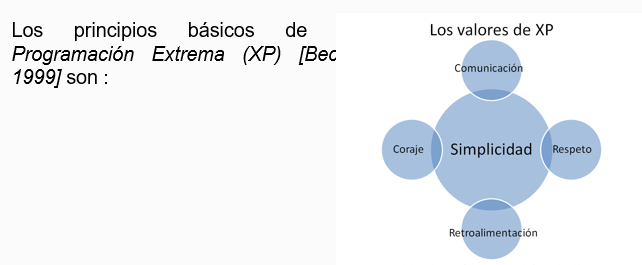
Valor en las personas e interacciones por encima de los procesos y herramientas: Priorizan la comunicación y la colaboración entre individuos sobre los procesos y las herramientas rígidas.

Entregar software operativo en lugar de documentación extensa: Se enfocan en resultados tangibles y funcionales más que en documentar exhaustivamente cada detalle.

Colaboración con el cliente sobre la negociación contractual: Buscan involucrar al cliente en el proceso de desarrollo y adaptarse a los cambios que surjan a lo largo del proyecto.

Responder al cambio en lugar de seguir una planificación estricta: Aceptan y se adaptan a cambios de requisitos incluso tardíos en el desarrollo.

Cada metodología ágil tiene sus propias prácticas y enfoques específicos. Por ejemplo:

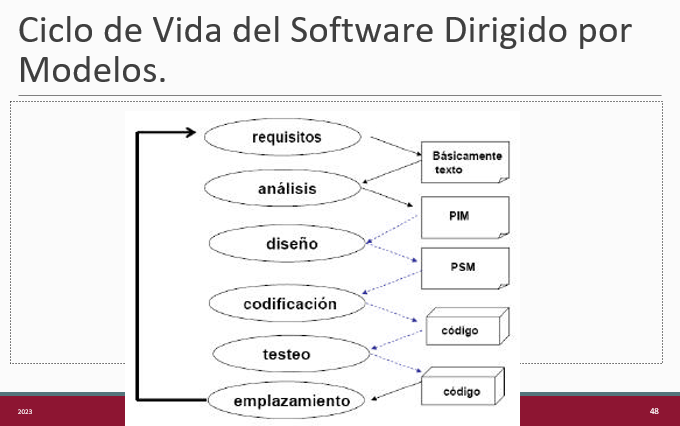
eXtreme Programming (XP): Se centra en la simplicidad, la comunicación y la retroalimentación continua. Promueve prácticas como pruebas unitarias continuas, programación en pares y entrega frecuente de software.

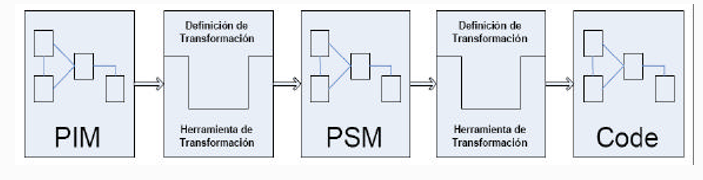
Scrum: Un marco de trabajo iterativo e incremental que se enfoca en entregas regulares y prioriza el trabajo mediante sprints cortos y definidos.

Kanban: Una metodología centrada en la gestión del flujo de trabajo, limitando el trabajo en curso y visualizando el proceso de desarrollo en un tablero.

Desarrollo de Software Basado en Modelos (MDD): Introduce modelos como una parte central del proceso de desarrollo, con modelos independientes de la plataforma y específicos de la plataforma, junto con transformaciones entre ellos.

En general, estas metodologías ágiles buscan optimizar la entrega de software valioso, adaptándose a los cambios y priorizando la calidad y la colaboración en lugar de la rigidez de los procesos tradicionales.





Calidad

Calidad

Es un concepto percibido de distintas formas, pero está definido por normas internacionales como el grado en que un conjunto de características cumple con los requisitos o propiedades que confieren aptitud para satisfacer necesidades expresadas o implícitas.

Calidad en Sistemas de Información

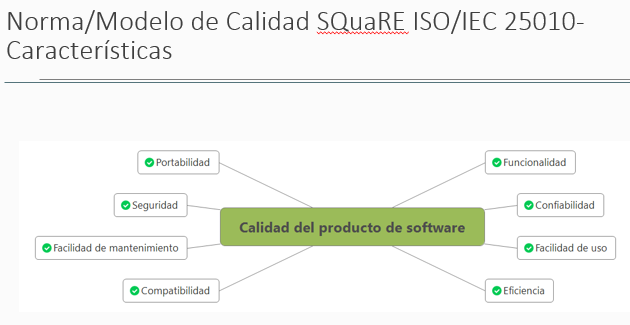
Conjunto de personas, datos, procesos y tecnología que interactúan para recopilar, procesar y proporcionar información.

Su calidad se aprecia en todas sus partes y depende de la organización de herramientas y la obtención de información necesaria para su funcionamiento.

Calidad de Software

Mejora significativa en los últimos años debido a una mayor conciencia de gestión de calidad y técnicas en la industria.

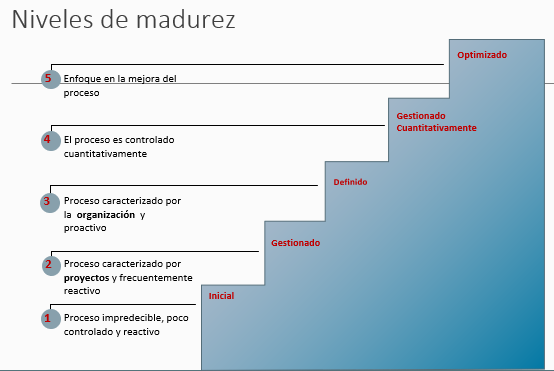
Se divide en calidad del producto y calidad del proceso, siendo estos dos aspectos dependientes entre sí.



Normas y Modelos de Calidad

Calidad de producto de software (ISO/IEC 25000) evalúa características como portabilidad, seguridad, funcionalidad, etc.

CMMI evalúa procesos de una organización y ofrece un marco para desarrollar procesos efectivos.



ISO 9000, familia de normas de gestión de calidad aplicables a cualquier tipo de organización, con el objetivo de mejorarla y obtener certificaciones importantes en el mercado global.

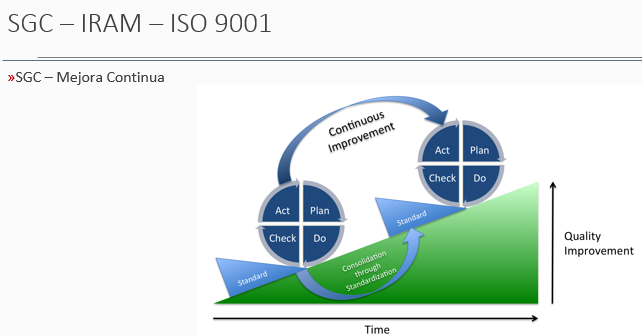
Beneficios de un Sistema de Gestión de Calidad (SGC) como ISO 9001

Asegura cumplir con requisitos legales y del cliente.

Mejora el rendimiento de la organización y la eficiencia operativa.

Mejora la satisfacción del cliente y toma de decisiones.

Optimiza las operaciones y el rendimiento financiero.

Estos estándares y modelos no solo buscan mejorar la calidad de los productos y procesos, sino que también impactan positivamente en la organización en su conjunto.

