Un patrón es una regla de tres partes que expresa una relación entre un contexto, un problema y una solución. los patrones son generativos, porque no ofrecen soluciones estáticas, sino dinámicas. se menciona que los patrones son herramientas que permiten la reutilización de soluciones exitosas a problemas conocidos y recurrentes.

Se presentan diferentes clasificaciones de patrones propuestas por diferentes autores:

Clasificación de Gamma y su equipo. Clasificar patrones en categorías según el propósito y el diseño orientado a objetos (creación, estructura, comportamiento).

La clasificación de Pressman, que cubre una variedad de patrones relacionados con diferentes aspectos de la ingeniería de software, como análisis, procesos, arquitectura, diseño, datos, diseño de interfaces y aplicaciones web.

Clasificación de Somerville. Clasificar patrones según objetivos específicos, tales como: Diseño orientado a objetos, diseño organizacional, usabilidad, interacción, gestión de configuración de software y arquitectura.

La clasificación de Eriksson y Penker, que distingue tres tipos de patrones según los problemas que abordan: patrones de negocios, patrones arquitectónicos y patrones de diseño

la propuesta de Eriksson y Penker, que proporcionan una estructura para describir y comprender los patrones de negocio.

Para describir cada patrón de negocio, utilizan una plantilla que incluye los siguientes atributos:

Nombre: Denominación del patrón.

Intención: Propósito general del patrón.

Motivación: Fuerzas que conducen a la solución.

Aplicabilidad: Situaciones problema a las que se aplica el patrón y qué resuelve.

Estructura: Representación gráfica del patrón en estándar UML.

Participantes: Elementos que participan del modelo.

Consecuencias: Cómo el patrón ayuda a resolver el objetivo y las situaciones problema.

Ejemplo: Caso concreto de aplicación del patrón.

Patrones relacionados: Otros patrones relacionados o complementarios.

Fuentes / Créditos: Referencias de donde se obtuvo la información del patrón.

Procesos de desarrollo de requerimientos para software: Incluye actividades como la identificación de clases de usuarios, la especificación de requerimientos y la validación de los mismos.

Especificación de requerimientos: Se destaca su importancia para gestionar el cambio y vincular los sistemas de información con la estrategia organizacional.

Esquemas Conceptuales: Se discute su definición desde diversas perspectivas y la importancia de su validación y verificación en el desarrollo de software.

Desarrollo Dirigido por Modelos (DSDM): Introduce este enfoque como una forma de obtener productos de software a partir de modelos independientes de la plataforma de implementación.