Universidad Veracruzana

DOMAIN DRIVEN DESING

ENSAYO DESARROLLO DE APLICACIONES

Domínguez Carmona José Javier

Mtro. Juan Luis López Herrera

22/Junio/2022

El Domain-Driven Design (DDD), es un enfoque que se centra en la comprensión y el modelado de problemas en lugar de realizar una implementación técnica. La metodología fue desarrollada por Eric Evans, brinda principios y prácticas para superar los desafíos de desarrollar software en entornos complejos.

La idea central de DDD es colocar al dominio de la aplicación como la parte más importante y se refleja en el diseño y construcción del proyecto. Para lograr lo anterior, el DDD tiene sus bases en una colaboración entre expertos y planificadores de renombre que tuvieron el objetivo de crear un lenguaje compartido común que ayude a comprender mejor los conceptos y requisitos comerciales.

Por otra parte, introduce conceptos clave como tipos estructurados, entidades, grupos, objetos de valor y dominios de contexto. Las entidades son objetos y sistemas definidos, mientras que los valores son propiedades inmutables que representan conceptos importantes en el dominio.

Uno de los principios clave considerado es el establecimiento de límites. Esto implica dividir el sistema en distintas regiones o subregiones, cada una con sus propias reglas y patrones. Esto ayuda a mantener la coherencia y la comprensión dentro de cada contexto y permite el desarrollo independiente de cada parte del sistema.

El diseño basado en el dominio es un enfoque destinado a adaptar el diseño de software a la complejidad del entorno en el que opera. Al centrarse en comprender y modelar el dominio, proporciona una base sólida para una programación de alta calidad, flexible y orientada a los negocios. Al trabajar con profesionales experimentados, los equipos de desarrollo pueden crear soluciones fáciles de mantener, crecer y comprender.

Esta arquitectura tiene como objetivo separar la lógica del dominio de la tecnología, lo que permite que la estructura del dominio sea independiente de la tecnología utilizada en aplicaciones, bases de datos y servicios externos. Esta división permite la agilidad y flexibilidad del sistema para gestionar cambios y pruebas.

Otra parte importante del diseño del dominio es continuar y refinar el modelo del dominio a medida que se aprende más sobre el problema. El diseño del área es transformador y colaborativo, lo que le permite ver nuevas perspectivas y cambiar modelos a medida que se aprende más sobre el área.

Los registros de eventos que ocurren en el dominio se pueden capturar y usar para comunicar y coordinar diferentes partes del sistema. Estos eventos mantienen la coherencia y la coordinación entre las diferentes partes del sistema, lo que permite una mayor flexibilidad y diseños más simples y flexibles.

El enfoque de esta metodología en la inteligencia comercial, la colaboración y la localización proporciona una base sólida para crear sistemas que sean flexibles, adaptables y respondan a las necesidades comerciales específicas. Mediante el uso de conceptos y técnicas de este modelo, los equipos de desarrollo pueden crear un mejor software que resuelva problemas de dominio complejos.

Además de lo anterior, el diseño del área de acuerdo con el dominio del problema fomenta la implementación de pruebas unitarias y pruebas de verificación como una parte importante del desarrollo. Las pruebas ayudan a garantizar que el modelo de dominio permanezca estable y funcional a medida que cambia el sistema. Al escribir pruebas que se enfocan en el comportamiento y las reglas del dominio, puede verificar que el software cumpla con los requisitos previamente establecidos.

Otro aspecto importante de DDD es el enfoque en la mejora continua. A medida que aumenta el conocimiento y la comprensión de nuevas ideas y conceptos, el modelo de dominio debe revisarse y actualizarse. Requiere una habilidad de aprendizaje constante y la voluntad de adaptar y cambiar las arquitecturas de software para comprender mejor las necesidades comerciales.

Vale la pena señalar que Domain Driven Design no es una solución para todas las aplicaciones de software. Su uso es muy útil en dominios y proyectos complejos que requieren una buena comprensión del negocio. En casos simples, todos los principios y prácticas de DDD se aplican más o menos.

En resumen, el diseño basado en dominios es un enfoque destinado a desarrollar software que visualice y resuelva mejor los problemas en un dominio determinado. El énfasis que tiene está direccionado en la comprensión empresarial, la colaboración y la creación de un modelo de dominio coherente proporcionando la base para un desarrollo de software eficiente, fácil de mantener y empresarial. Mediante el uso de los principios y prácticas de DDD, los equipos de desarrollo pueden crear sistemas que sean simples, flexibles y exitosos en entornos complejos o difíciles de manejar.

Bibliografía

Evans, E. (2004). Domain-driven design: tackling complexity in the heart of software. Addison-Wesley Professional.

Millett, S., & Tune, N. (2015). Patterns, principles, and practices of domain-driven design. John Wiley & Sons.