

Anemic Domain Model, by Martin Fowler

Últimamente, el anti-patrón conocido como modelo de dominio anémico parece tener un impulso particular. Tanto Martin Fowler, como Eric Evans, quienes son fieles defensores del Modelo de Dominio, piensan que esto podría ser un problema.

El principal síntoma de los modelos de dominio anémicos es que, en un primer golpe de vista, parecen reales. Cuentan con objetos, y estos objetos están relacionados. El problema viene cuando se profundiza un poco más en el comportamiento, ya que estos objetos apenas tienen funcionalidad más allá de getters y setters.

Pero, el horror fundamental de éste anti-patrón es que es totalmente contrario a la idea básica del diseño orientado a objetos, el cual se basa en combinar los datos y el proceso. Por otro lado, muchas personas piensan que los objetos anémicos son objetos reales, y esto les hace perder totalmente la esencia del diseño orientado a objetos.

En esencia, el problema de los modelos de dominio es que, además de tener todos los costes y problemas de un modelo de dominio, no producen ningún beneficio.

También merece la pena enfatizar que el comportamiento de los objetos del dominio no debe contradecir el enfoque de usar capas para separar las responsabilidades de presentación y persistencia.

Un gran foco de confusión en todos éste tema es que algunos expertos recomiendan una capa de servicios procedimentales encima del modelo de dominio, para formar una capa de servicios.

En el libro Domain Driven Desing de Eric Evans, se dice que una capa de servicio define los trabajos que el software debe hacer y dirige los objetos de dominio para resolver problemas. La tarea de ésta capa es coordinar diferentes tareas. Además, esta capa no cuenta con estado.

También se dice que la capa de modelo es la encargada de representar los conceptos de negocio. Ésta capa sería el corazón del software de negocio.

La clave está en que la capa de servicios es pequeña, es decir, toda la lógica importante se encuentra en la capa del modelo.

Martin Fowler nos dice que no sabe muy bien por qué este anti-patrón es tan común, aunque sospecha que podría ser porque esa gente no ha trabajado correctamente con un modelo de dominio correcto.

Concluyendo, cuanta más funcionalidad haya en los servicios, más probable es que se esté perdiendo los beneficios del modelo de dominio.