Важность доменного дизайна

Domain-Driven Design (DDD) — это подход к разработке программного обеспечения, упрощающий сложность, с которой сталкиваются разработчики, соединяя реализацию с развивающейся моделью.

Если бы мы взяли концепцию, разделили ее на четыре составляющих и перемешали вместе или взяли одну и ту же концепцию и подали ее в виде четырех разных предметов на тарелке, что будет более эффективным? Давайте использовать еду в качестве примера — скажем, миску с чили. Мы знаем, что чили готовят из разных ингредиентов (мясо, соус и бобы), помещают их в кастрюлю и готовят в течение 30–45 минут. Напротив, у нас есть стейк, картофель и овощи на тарелке, готовые к подаче.

Каким из них было бы легче управлять, если бы мы добавляли / убирали продукты: вынимали фасоль и нарезанные кубиками помидоры из миски с чили или убирали овощи с тарелки и добавляли еще один продукт? В этом случае чили представляет управляемый данными дизайн. Слои ингредиентов в последовательности друг с другом в порядке (например, бобы, мясо, помидоры и соус) завершают рецепт.

Давайте сравним это с управляемым доменом дизайном. У вас есть домен (стейк), ограниченный контекст (овощи) и принцип единой ответственности (картофель). Любой из них может быть заменен чем-то другим вне домена, и еда все равно будет считаться полноценным блюдом. Способность добавлять / удалять из организованного программного обеспечения — вот как функционирует DDD, поэтому важно, чтобы разработчики программного обеспечения и владельцы бизнеса использовали его.

Целью DDD является следующее.

1. Предоставить принципы и шаблоны для решения сложных проблем.

2. База комплексных конструкций по модели домена.

3. Инициировать творческое сотрудничество между техническими специалистами и экспертами по предметной области, чтобы итеративно усовершенствовать концептуальную модель, которая решает проблемы предметной области.

Как разработчики, мы взволнованы, чтобы начать проект, начать программировать и создавать программное обеспечение. Тем не менее, мы не можем создавать программное обеспечение без понимания потребностей клиента. DDD уделяет большое внимание не только пониманию того, что хочет клиент, но и работе с ним в качестве партнеров в рамках проекта. Конечная цель — не только написать код или даже создать программное обеспечение, но и решить проблемы!