

# Содержание

<b>Предисловие</b>	10
Для кого предназначена эта книга	11
Темы, рассмотренные в книге	12
Соглашения	13
<b>Благодарности</b>	14
<b>Об авторе</b>	15
<u>ГЛАВА 1</u>	
<b>Краткий обзор DDD</b>	17
Вреден ли DDD?	18
Хороший, плохой и эффективный дизайн	19
Стратегическое проектирование	23
Тактическое проектирование	24
Процесс обучения и уточнения знаний	25
Поехали!	26
<u>ГЛАВА 2</u>	
<b>Стратегическое проектирование с помощью ОГРАНИЧЕННЫХ КОНТЕКСТОВ и ЕДИНОГО ЯЗЫКА</b>	27
Эксперты проблемной области и бизнес-факторы	34
Типичный пример	37
Необходимость фундаментального стратегического проектирования	41
Ставьте проблемы и обобщайте	45
Разработка ЕДИНОГО ЯЗЫКА	51
Реализация сценариев	55
Далекие перспективы	57
Архитектура	57
Резюме	59
<u>ГЛАВА 3</u>	
<b>Стратегическое проектирование с помощью ПОДОВЛАСТЕЙ</b>	61
Что такое ПОДОВАСТЬ	62
Типы ПОДОВАСТЕЙ	62
Проблема сложности системы	63
Резюме	66

ГЛАВА 4

<b>Стратегическое проектирование на основе СВЯЗЫВАНИЯ КОНТЕКСТОВ</b>	67
Способы связывания контекстов	69
ПАРТНЕРСТВО	70
ОБЩЕЕ ЯДРО	70
КЛИЕНТ-ПОСТАВЩИК	71
КОНФОРМИСТ	71
ПРЕДОХРАНИТЕЛЬНЫЙ УРОВЕНЬ	72
СЛУЖБА С ОТКРЫТЫМ ПРОТОКОЛОМ	73
ОБЩЕДОСТУПНЫЙ ЯЗЫК	73
ОТДЕЛЬНОЕ СУЩЕСТВОВАНИЕ	74
БОЛЬШОЙ КОМ ГРЯЗИ	74
Правильное использование СВЯЗЫВАНИЯ КОНТЕКСТОВ	76
Удаленный вызов процедур по протоколу SOAP	77
Протокол RESTful HTTP	78
РАССЫЛКА СООБЩЕНИЙ	80
Пример СВЯЗЫВАНИЯ КОНТЕКСТОВ	85
Резюме	88

ГЛАВА 5

<b>Тактическое проектирование с помощью АГРЕГАТОВ</b>	89
Зачем нужны АГРЕГАТЫ	90
Эмпирические правила проектирования АГРЕГАТОВ	95
Правило 1. Защищайте бизнес-инварианты в границах АГРЕГАТА	95
Правило 2. Проектируйте маленькие АГРЕГАТЫ	97
Правило 3. Ссылайтесь на другие АГРЕГАТЫ только по идентификаторам	98
Правило 4. Обновляйте другие АГРЕГАТЫ, руководствуясь принципом итоговой согласованности	99
Моделирование агрегатов	102
Тщательно выбирайте абстракции	107
Агрегаты правильного размера	109
Тестируемые модули	111
Резюме	112

ГЛАВА 6

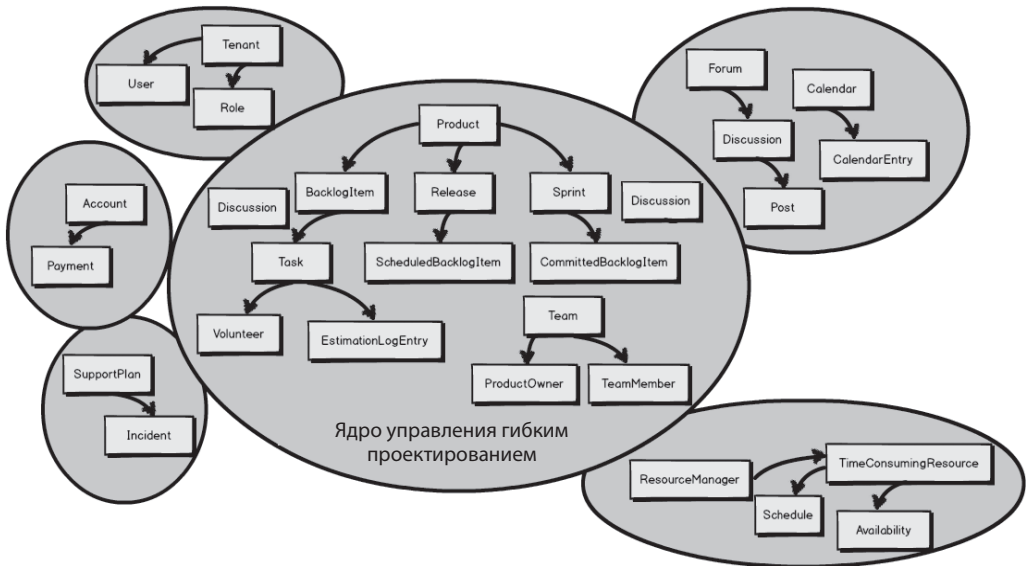
<b>Тактическое проектирование с помощью СОБЫТИЙ ПРЕДМЕТНОЙ ОБЛАСТИ</b>	113
Проектирование, реализация и использование СОБЫТИЙ ПРЕДМЕТНОЙ ОБЛАСТИ	114

ИСТОЧНИКИ СОБЫТИЙ	121
Резюме	123

## ГЛАВА 7

<b>Инструментальные средства для повышения эффективности проектирования</b>	125
СОБЫТИЙНЫЙ ШТУРМ	126
Другие инструменты	138
Применение принципов DDD для гибкого проектирования	138
Начнем с начала	140
Использование SWOT-анализа	140
Всплески и долги моделирования	142
Идентификация задач и оценивание	143
Моделирование с ограничением времени	145
Реализация	146
Взаимодействие с ЭКСПЕРТАМИ ПРЕДМЕТНОЙ ОБЛАСТИ	148
Резюме	149
<b>Библиография</b>	151
<b>Предметный указатель</b>	153

## Стратегическое проектирование с помощью ПОДОБЛАСТЕЙ



Работая над проектом DDD, вы всегда будете сталкиваться с многочисленными ОГРАНИЧЕННЫМИ КОНТЕКСТАМИ. Один из ОГРАНИЧЕННЫХ КОНТЕКСТОВ содержит СМЫСЛОВОЕ ЯДРО, а в других ОГРАНИЧЕННЫХ КОНТЕКСТАХ заключены различные ПОДОБЛАСТИ. В предыдущей главе была показана важность разделения моделей, каждая из которых использует свой ЕДИНЫЙ ЯЗЫК, и формирования многочисленных ОГРАНИЧЕННЫХ КОНТЕКСТОВ. В предыдущей диаграмме есть шесть ОГРАНИЧЕННЫХ КОНТЕКСТОВ и шесть ПОДОБЛАСТЕЙ. Поскольку при разработке модели использовались методы стратегического предметно-ориентированного проектирования, группы разработчиков достигли ее оптимального состава: по одной ПОДОБЛАСТИ в каждом ОГРАНИЧЕННОМ КОНТЕКСТЕ и один ОГРАНИЧЕННЫЙ КОНТЕКСТ на каждую

ПОДОБЛАСТЬ. Иначе говоря, ЯДРО УПРАВЛЕНИЯ ГИБКИМ ПРОЕКТИРОВАНИЕМ представляет собой и один точно определенный ОГРАНИЧЕННЫЙ КОНТЕКСТ, и одну точно определенную ПОДОБЛАСТЬ. В редких ситуациях могут возникнуть многочисленные ПОДОБЛАСТИ в одном ОГРАНИЧЕННОМ КОНТЕКСТЕ, но это не позволяет получить оптимальный результат моделирования.

---

## Что такое ПОДОБЛАСТЬ

Просто говоря, ПОДОБЛАСТЬ — это часть вашей предметной области. Вы можете думать о ПОДОБЛАСТИ как об отдельной логической модели некоторой предметной области. Большинство предметных областей обычно являются слишком большими и сложными, чтобы обсуждать их как единое целое, поэтому мы интересуемся только ПОДОБЛАСТЯМИ, которые должны использоваться в рамках отдельного проекта. ПОДОБЛАСТИ можно использовать для логического разделения предметной области, чтобы понять *пространство состояний* большого и сложного проекта.

Другой способ интерпретации ПОДОБЛАСТИ состоит в том, что она рассматривается как четко очерченная область компетенции, позволяющая найти решение для СМЫСЛОВОГО ЯДРА вашего бизнеса. Это означает, что в конкретной ПОДОБЛАСТИ есть один или несколько ЭКСПЕРТОВ ПРЕДМЕТНОЙ ОБЛАСТИ, которые очень хорошо понимают аспекты бизнеса, ведение которого облегчает данная ПОДОБЛАСТЬ. Кроме того, ПОДОБЛАСТЬ также имеет в некоторой степени стратегическое значение для вашего бизнеса.

Если бы для разработки ПОДОБЛАСТИ использовался подход DDD, то она была бы реализована как точно определенный ОГРАНИЧЕННЫЙ КОНТЕКСТ. ЭКСПЕРТАМИ ПРЕДМЕТНОЙ ОБЛАСТИ, которые специализируются в данной конкретной сфере бизнеса, были бы члены группы, разработавшей данный ОГРАНИЧЕННЫЙ КОНТЕКСТ. Использование DDD для разработки точно определенного ОГРАНИЧЕННОГО КОНТЕКСТА — оптимальный выбор, о котором иногда приходится только мечтать.

---

## Типы ПОДОБЛАСТЕЙ

В рамках проекта есть три простых типа ПОДОБЛАСТЕЙ.

- СМЫСЛОВОЕ ЯДРО. Это область стратегических инвестиций в отдельную, точно определенную модель предметной области путем затрат существенных ресурсов для тщательной разработки ЕДИНОГО ЯЗЫКА в явном ОГРАНИЧЕННОМ КОНТЕКСТЕ. Эта задача имеет очень высокий приоритет для вашей организации, потому что именно ее решение дает вам конкурентное преимущество. Поскольку ваша организация

не может быть лидером во всем, что она делает, ваше СМЫСЛОВОЕ ЯДРО выделяет область ее конкурентного преимущества. Для того чтобы достичь такой глубины знаний, необходимы заинтересованность, сотрудничество и экспериментирование. Именно сюда организация должна вложить самые большие инвестиции, предназначенные для разработки программного обеспечения. Позднее мы рассмотрим средства ускорения и управления выполнением проектов.

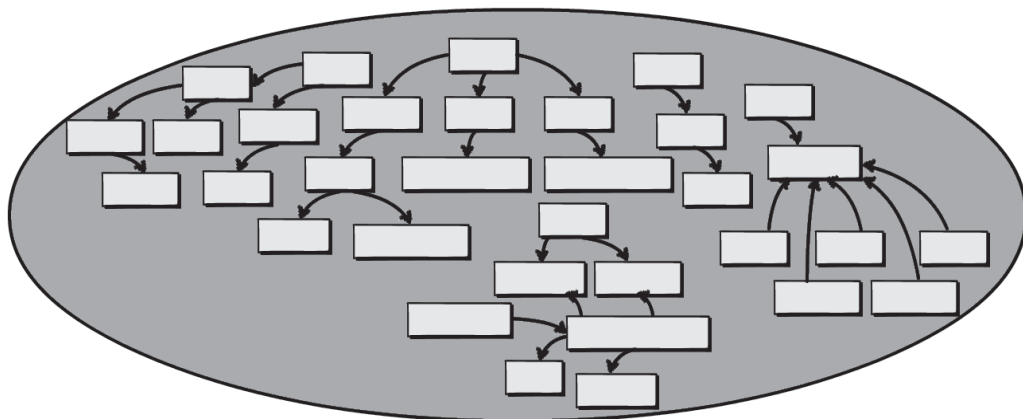
- **ВСПОМОГАТЕЛЬНАЯ ПОДОБЛАСТЬ.** Такая подобласть требует специального моделирования ситуаций, для которых не существует готовых решений. Однако такие ПОДОБЛАСТИ не требуют настолько больших инвестиций, как СМЫСЛОВОЕ ЯДРО. Разработку таких ОГРАНИЧЕННЫХ КОНТЕКСТОВ можно поручить сторонним организациям, чтобы не перепутать их с областями стратегического назначения и не вложить в них слишком большие средства. И все же это важные программные модели, потому что ваше СМЫСЛОВОЕ ЯДРО не может успешно работать без них.
- **УНИВЕРСАЛЬНАЯ ПОДОБЛАСТЬ.** Этот вид решения можно приобрести готовым, но можно также привлечь к его разработке сторонние организации или даже отдельное подразделение вашей организации, в которую не входят элитные разработчики, которые в основном должны заниматься СМЫСЛОВЫМ ЯДРОМ и в меньшей степени ВСПОМОГАТЕЛЬНЫМИ ПОДОБЛАСТЯМИ. Опасайтесь принять УНИВЕРСАЛЬНУЮ ПОДОБЛАСТЬ за СМЫСЛОВОЕ ЯДРО. Не надо вкладывать в нее крупные инвестиции.

Обсуждая проект, разрабатываемый на основе принципов DDD, мы обычно обсуждаем его СМЫСЛОВОЕ ЯДРО.

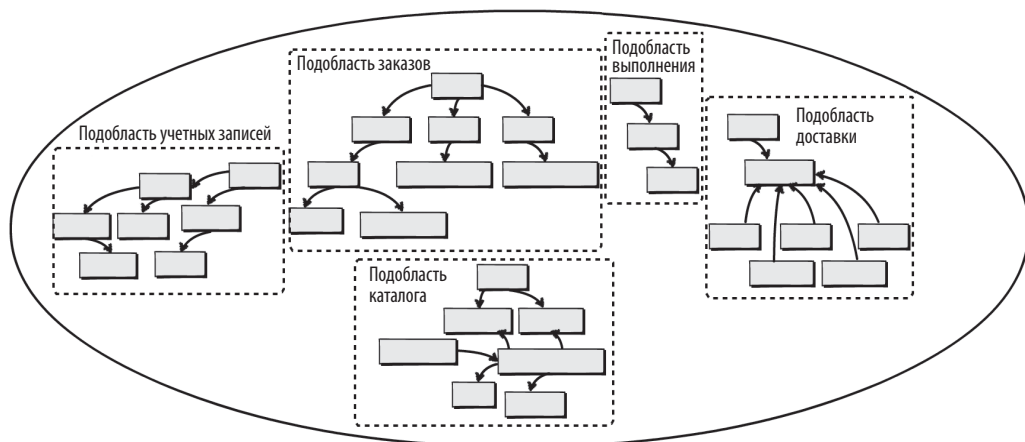
---

## Проблема сложности системы

Некоторые из системных границ в пределах предметной области с высокой вероятностью могут очерчивать унаследованные системы, которые ваша организация создала ранее или купила по лицензии. Вы не сможете намного улучшить унаследованные системы, но все же должны учитывать их, если они влияют на СМЫСЛОВОЕ ЯДРО проекта. Для этого в качестве инструмента для обсуждения вашего ПРОСТРАНСТВА СОСТОЯНИЙ необходимо использовать ПОДОБЛАСТИ.



К сожалению, некоторые унаследованные системы настолько неудобны для предметно-ориентированного проектирования на основе ОГРАНИЧЕННЫХ КОНТЕКСТОВ, что их можно было бы даже назвать *неограниченными* унаследованными системами. Такие унаследованные системы представляют собой то, что я уже назвал БОЛЬШИМ КОМОМ ГРЯЗИ. В действительности система может состоять из многочисленных запутанных моделей, которые следовало бы проектировать и реализовывать по отдельности, но они были смешаны вместе в одно очень сложное и переплетенное месиво.

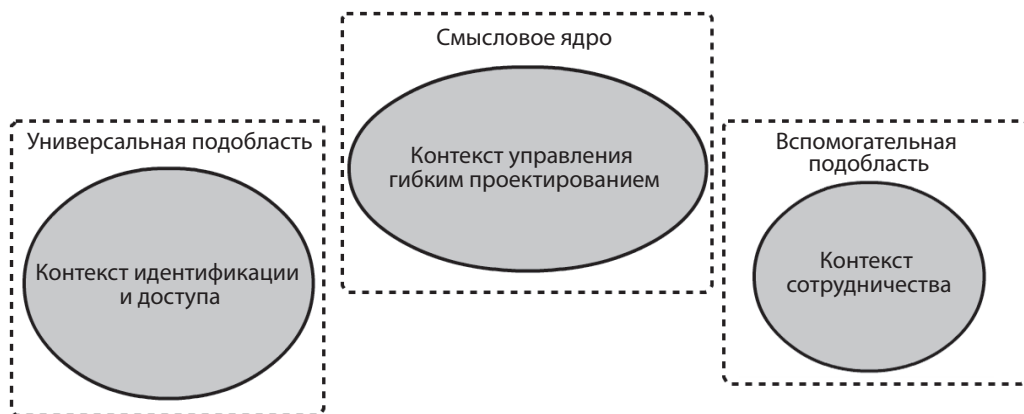


Иначе говоря, когда мы обсуждаем унаследованную систему, то в ней, вероятно, существует несколько, а может, даже много, *логических* моделей предметной области. Каждую из этих логических моделей предметной области можно рассматривать как ПОДОВАСТЬ. На диаграмме каждая логическая ПОДОВАСТЬ в неограниченном унаследованном монолитном БОЛЬШОМ КОМЕ ГРЯЗИ обведена пунктиром. На рисунке показаны пять логических моделей, или ПОДОВАСТЕЙ. Обработка логических ПОДОВАСТЕЙ также помога-

ет справиться со сложностью больших систем. Это очень полезно, поскольку позволяет нам работать с *пространством задач* так, будто оно состоит из многочисленных ОГРАНИЧЕННЫХ КОНТЕКСТОВ, разработанных на основе принципов DDD.

Унаследованная система кажется менее монолитной и запутанной, если мы представим себе разные ЕДИНЫЕ ЯЗЫКИ, хотя бы ради понимания того, как нам их интегрировать в новую систему. Размышления и обсуждения такой унаследованной системы на основе ПОДОВАСТИ помогают нам справиться со сложностью большой запутанной модели. И поскольку мы рассуждаем, используя этот инструмент, мы можем выделить ПОДОВАСТИ, которые имеют большую ценность для бизнеса и необходимы для нашего проекта, и те, которые не заслуживают большого внимания.

Учитывая сказанное, вы можете даже показать СМЫСЛОВОЕ ЯДРО, над которым работаете или собираетесь работать, на той же самой простой диаграмме. Это поможет вам понять связи и зависимости между ПОДОВАСТЯМИ. Но я отложу детали этого обсуждения для КАРТ КОНТЕКСТОВ.



В рамках подхода DDD между ОГРАНИЧЕННЫМИ КОНТЕКСТАМИ и ПОДОВАСТЯМИ должно существовать взаимно однозначное соответствие. Таким образом, если в проекте DDD существует ОГРАНИЧЕННЫЙ КОНТЕКСТ, то в нем следует выделить одну модель ПОДОВАСТИ. Это может оказаться не всегда возможным или практичным, но к этому необходимо стремиться. Это сохранит ваши ОГРАНИЧЕННЫЕ КОНТЕКСТЫ точно определенными и позволит сосредоточиться на основной стратегической инициативе.

Если вы вынуждены создать вторую модель в том же самом ОГРАНИЧЕННОМ КОНТЕКСТЕ (в пределах вашего СМЫСЛОВОГО ЯДРА), то необходимо отделить вторичную модель от вашего ОСНОВНОГО ЯДРА, используя полностью отдельный МОДУЛЬ [IDDD]. (МОДУЛЬ DDD — это, как правило, пакет в языках Scala и Java или пространство имен в языках F# и C#.) Это



позволит ясно выразить, что одна модель — это ядро, а вторая носит вспомогательный характер. Это специфическое использование разделения на ПОДОБЛАСТИ вы можете использовать в своем *пространстве решений*.

---

## Резюме

В этой главе вы узнали:

- что такое ПОДОБЛАСТИ и как они используются в ПРОСТРАНСТВЕ ПРОБЛЕМ и ПРОСТРАНСТВЕ РЕШЕНИЙ;
- чем различаются СМЫСЛОВОЕ ЯДРО, ВСПОМОГАТЕЛЬНАЯ ПОДОБЛАСТЬ и УНИВЕРСАЛЬНАЯ ПОДОБЛАСТЬ;
- как использовать ПОДОБЛАСТИ для интеграции с унаследованным БОЛЬШИМ КОМОМ ГРЯЗИ;
- что взаимно однозначное соответствие между ОГРАНИЧЕННЫМИ КОНТЕКСТАМИ DDD и их ПОДОБЛАСТЯМИ является очень важным;
- как отделить модель ВСПОМОГАТЕЛЬНОЙ ПОДОБЛАСТИ от модели СМЫСЛОВОГО ЯДРА с помощью модуля DDD в ситуациях, когда их непрактично разделять между двумя разными ОГРАНИЧЕННЫМИ КОНТЕКСТАМИ.

Исчерпывающее описание ПОДОБЛАСТЕЙ приведено в главе 2 книги *Implementing Domain-Driven Design* [IDDD].