

Независимая научно-практическая
конференция «Разработка ПО 2011»

31 октября – 3 ноября, Москва



DDD – эффективный способ работы в условиях системой сложности

Докладчик:

Максим Цепков (M.Tsepkov@custis.ru)

www.custis.ru

CUSTIS®

Предметная область доклада

➔ Заказная разработка:

- outsourcing, inhouse
- решения для нескольких заказчиков

Наш опыт

➔ Внедрение готовых решений с адаптацией

➔ Развитие эксплуатируемых систем



Большие предприятия и большие системы.
Высокая динамика изменений бизнеса.

План доклада

- ▶ Задачи автоматизации
- ▶ Подходы. Водопад vs DDD
- ▶ DDD на практике. Опыт CUSTIS
 - DDD + ООП. Объектная модель
 - DDD и документооборот. Развитие объектной модели
 - DDD и учет. Необъектная модель
- ▶ Заключение

Задача 1: Понять предприятие

- ▶ Большие предприятия обладают системной сложностью.
- ▶ Достаточно снизить сложность декомпозицией не получается (есть много сквозных процессов и взаимодействия).
- ▶ Распределенные знания о предприятии.
 - Топ-менеджмент представляет общие бизнес-модели.
 - Линейные сотрудники знают детали процессов в своей области.
 - Общие модели и подробности противоречат друг другу.
- ▶ Бизнес-модель предприятия сложна и плохо проверяема, понимается только авторами.

Задача 2: Представить ИТ-систему

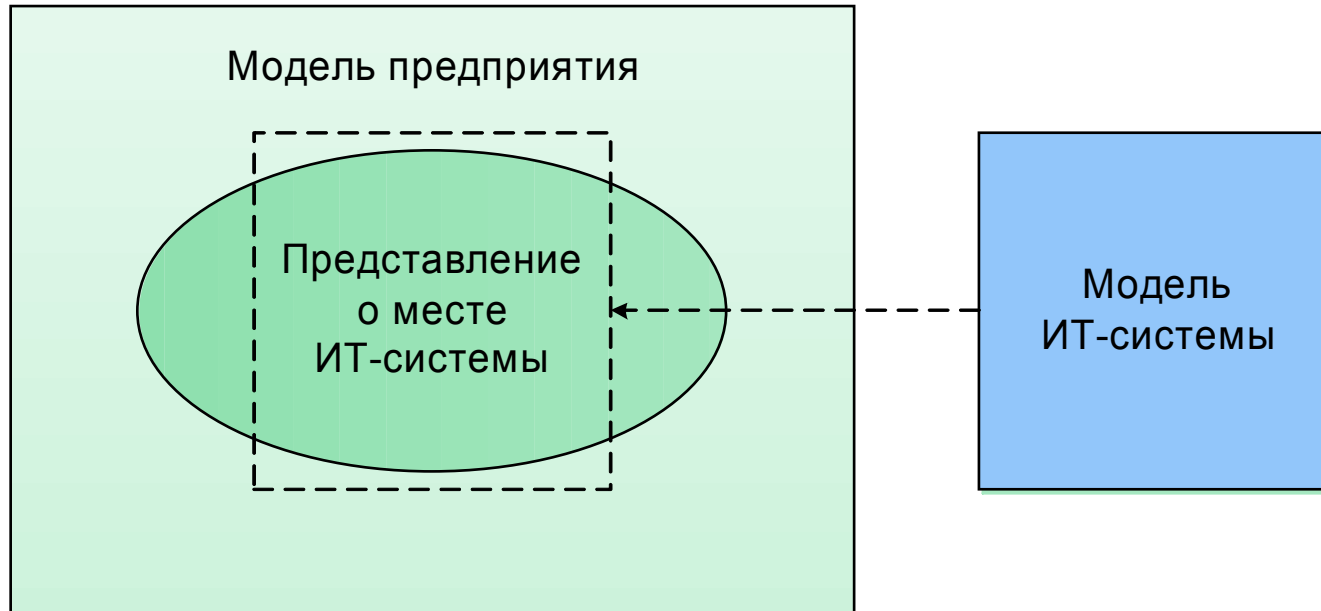
➤ Цели автоматизации:

- эффективное выполнение бизнес-процессов;
- жесткость исполнения (исключить человеческий фактор);
- согласованные действия подразделений при выполнении процессов.

➤ Сложность модели системы сравнима со сложностью предприятия (иначе не работает).

- Декомпозиция нарушит согласованность или эффективность.
- Построение из типовых элементов сохранит человеческий фактор.

Модели надо совместить...

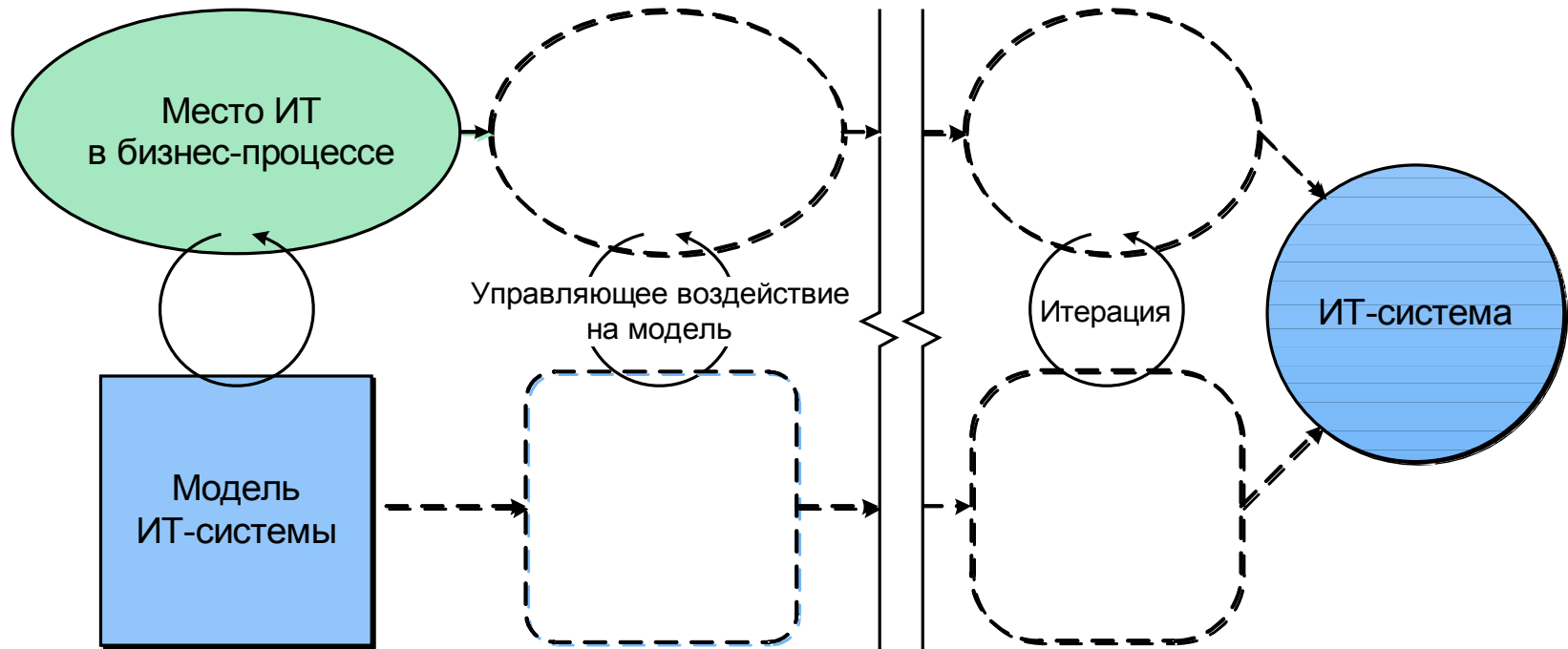


Модель системы не соответствует представлению бизнеса о ее месте в модели предприятия.

«Не то чтобы совсем не попал, но только не попал в шарик...»



Итерационное развитие модели



Требуется активное общение ИТ и бизнеса – изменение модели системы и ее места в бизнесе.

Проблемы водопадной модели

В процессе участвуют две сложные модели:

- **Бизнес-модель предприятия** и место системы в нем – на бизнес-языке.
- **Модель ИТ-системы** – на языке ИТ.

Их надо согласовывать и итеративно изменять.

Альтернатива – DDD

(предметно-ориентированное проектирование)

Вырабатываем **единый язык** (ubiquitous language):

- Он построен на основе терминов предметной области.
- Его понимают ИТ-специалисты и эксперты бизнеса.
- На нем описана модель ИТ-системы и ее место в бизнес-процессах.

Вместо двух сложных моделей, которые нужно согласовывать,
используется одна модель.

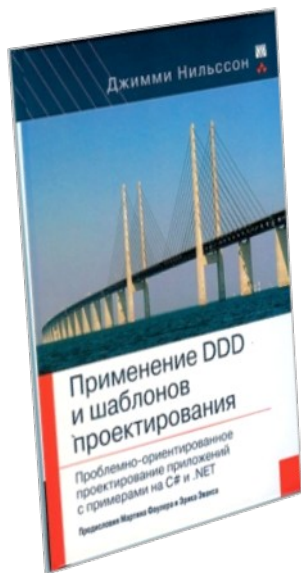


DDD = Единый язык + Модель

Что такое DDD

Концептуальная книга Эрика Эванса

- на английском – в 2003 г.
- на русском – только в 2010 г.



Практическая книга Джимми Нильссона

- на английском – в 2006 г.
- на русском – в 2007 г. (почти сразу!)

Практика применения DDD при разработке корпоративных приложений

Часть 1. DDD + ООП: Объектная модель

Язык и метаязык

- ▶ Единый язык – из предметной области, иначе его не поймут бизнес-эксперты.
- ▶ При построении ИТ-модели эффективно применять общие подходы, не зависящие от области.

Как это совместить?

Общие подходы задают метаязык описания модели:

- из каких элементов состоит модель;
- как эти элементы комбинируются в сложные конструкции;
- как модель визуально представляется в проекте;
- как модель отражается в реализации.

Но сами элементы находятся в предметной области.

Язык ООП

Элементы метаязыка Язык ООП

Элементы языка
и способ их соединения
в сложные конструкции

Объекты с атрибутами
и методами и связи
между ними:
клиенты, накладные

Визуальный образ
для эффективного
представления

Диаграмма классов
и другие диаграммы UML

Способ отражения
модели в реализацию

Типы в программе,
соответствующие
бизнес-объектам

ООП в DDD vs «просто ООП»

➔ ООП в DDD

- Выделяем слой бизнес-логики.
- Проектируем и реализуем его на основе Domain Model.
- Модель согласуется с заказчиком, функционал и интерфейсы из нее следуют.

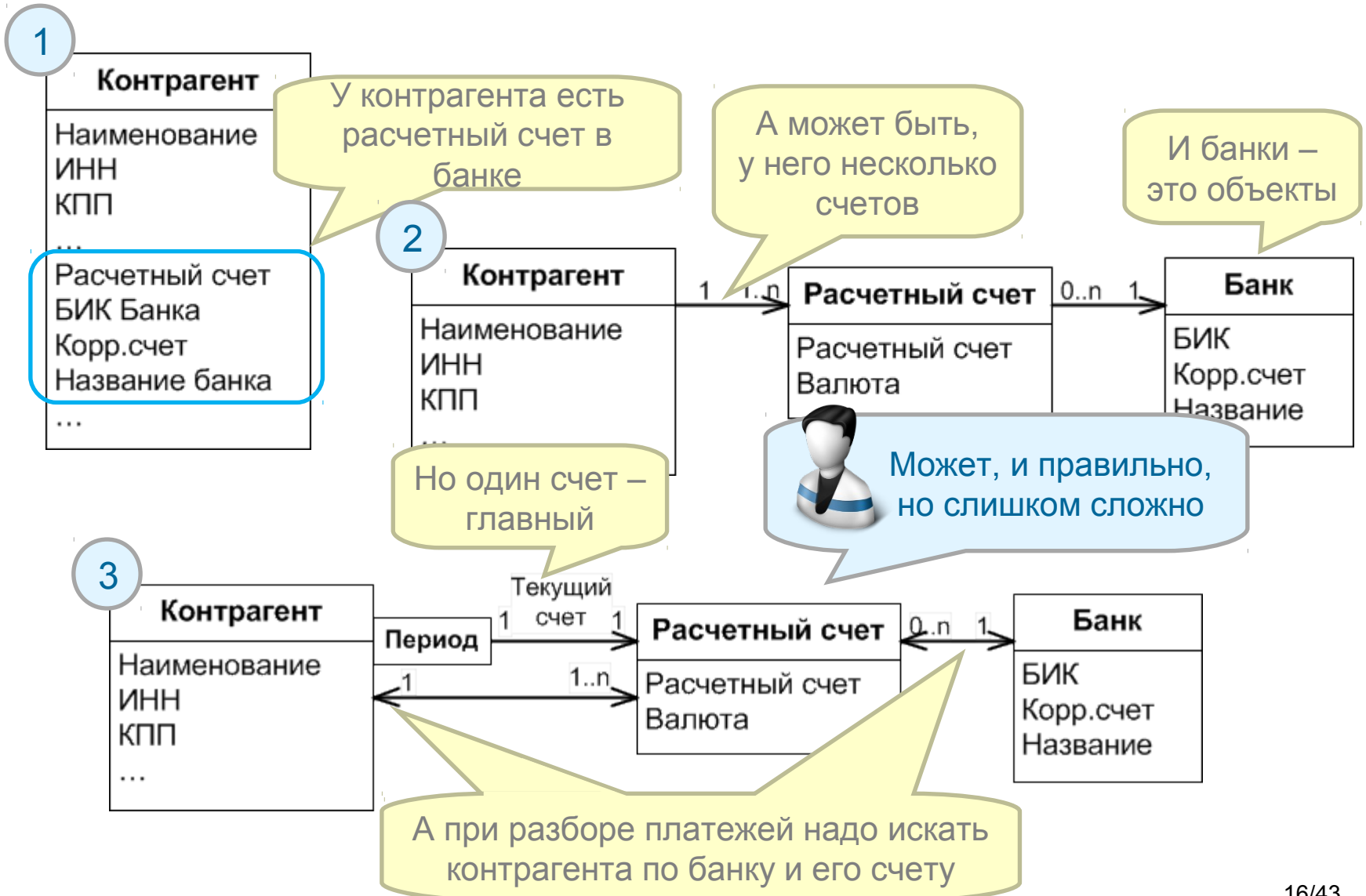
➔ «Просто ООП»

- Объекты могут не иметь отношения к предметной области.
- С заказчиком согласуется только функционал и интерфейсы.
- Пример: приложение на VisualStudio – DataSet + Grid-интерфейсы.

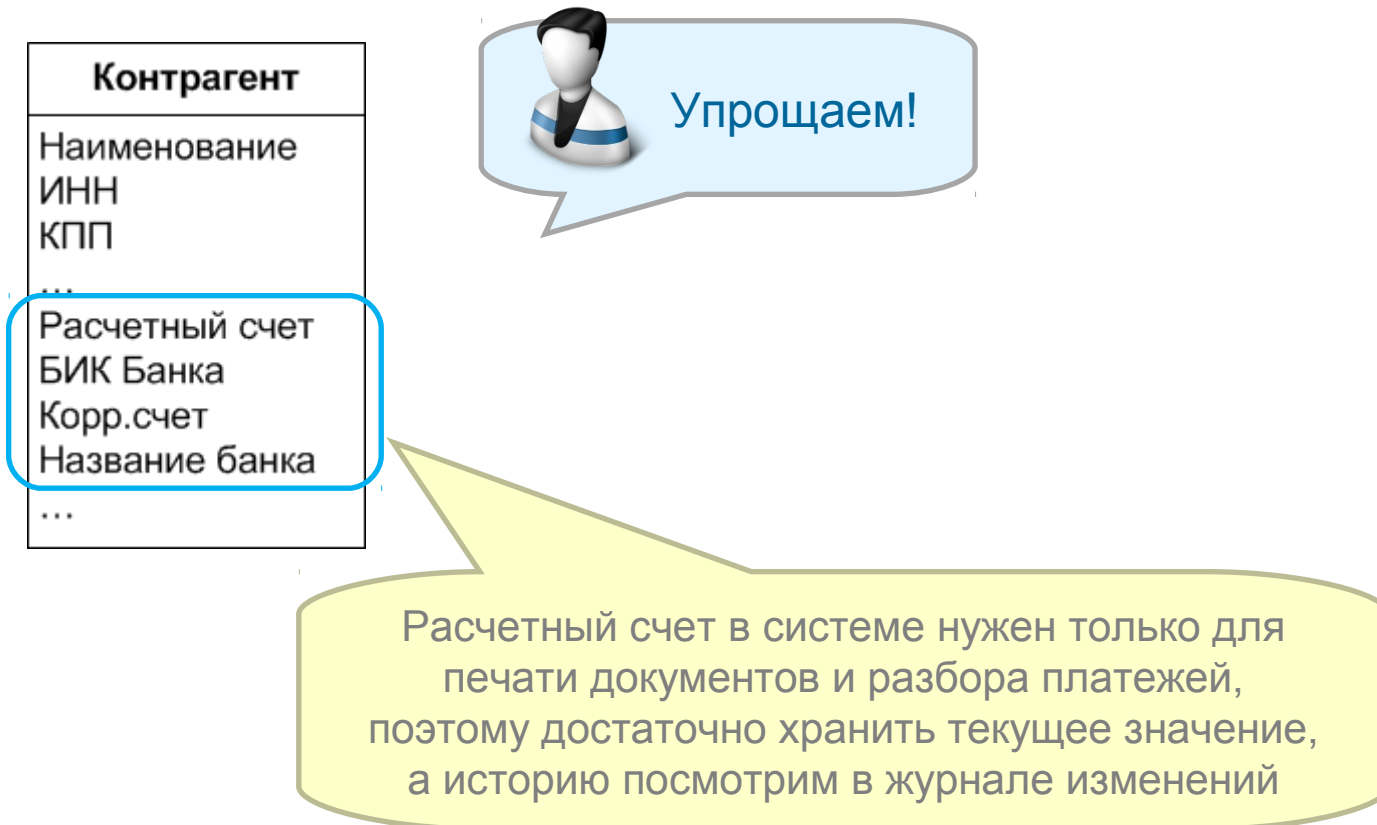
ООП в DDD: Результат

- 😊 Модель предметной области проверяется заказчиком – снижается риск ошибки в постановках.
- 😞 Многие решения вынесены на уровень модели, их изменения надо согласовывать.
- 😞 DDD требует от разработчиков изучения бизнес-области и больше квалификации в целом.
- 😊 Взгляд бизнеса и ИТ на систему одинаков.
- 😊 Пользователи и разработчики понимают друг друга.

Практика: Объектная модель



Практика: объектная модель



Практика: объектная модель

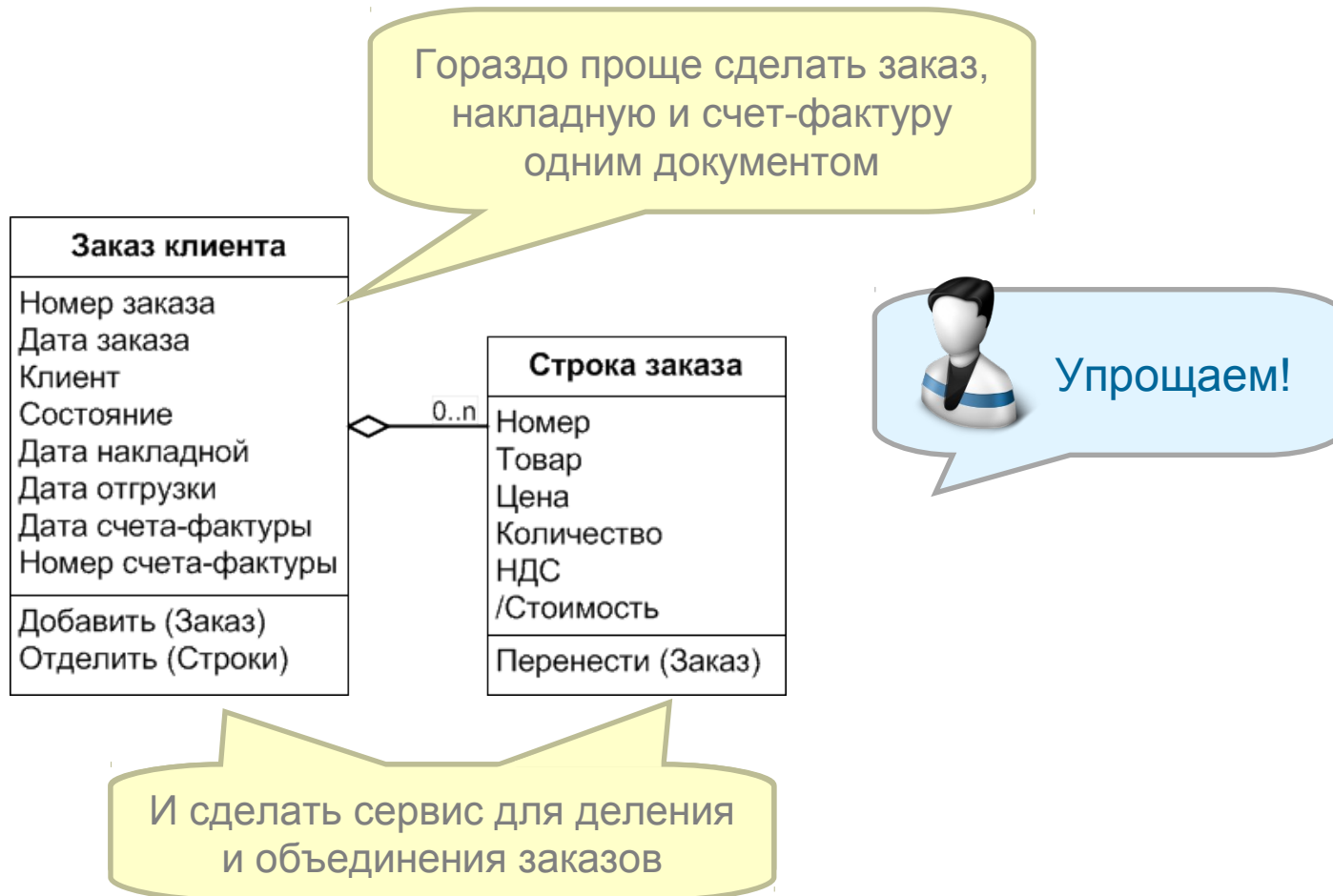
Для отгрузки заказов нужна накладная Или несколько...

А потом клиенту надо выставить счета-фактуры

Если нашли ошибку в документах, например в цене, ее надо править везде



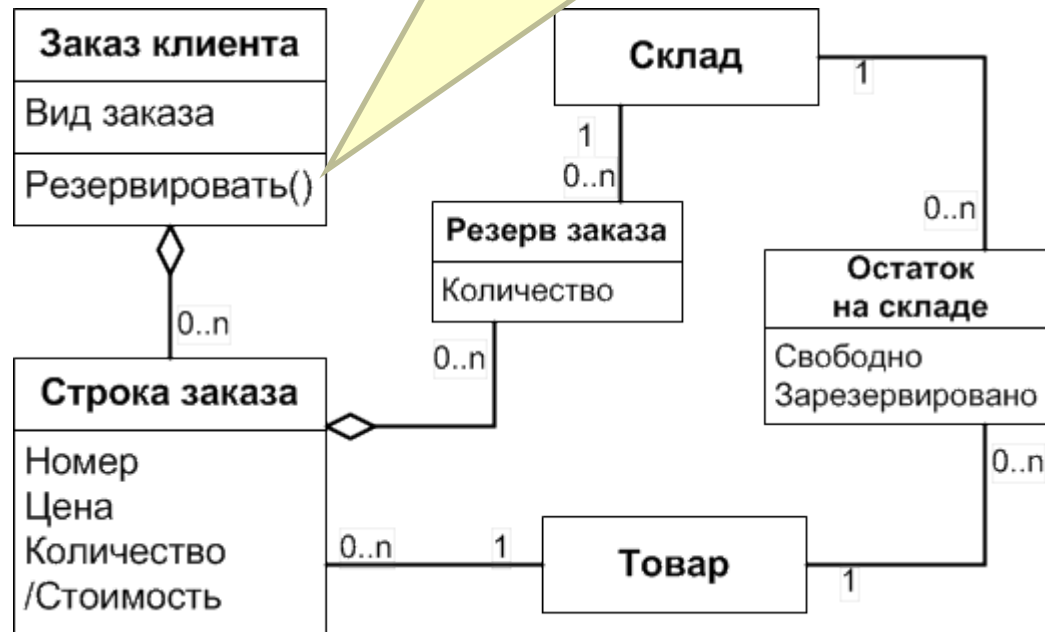
Практика: объектная модель



Практика: объектная модель


Изоляция объектов

Метод резервирования заказа собирает товар по складам, алгоритм зависит от вида заказа



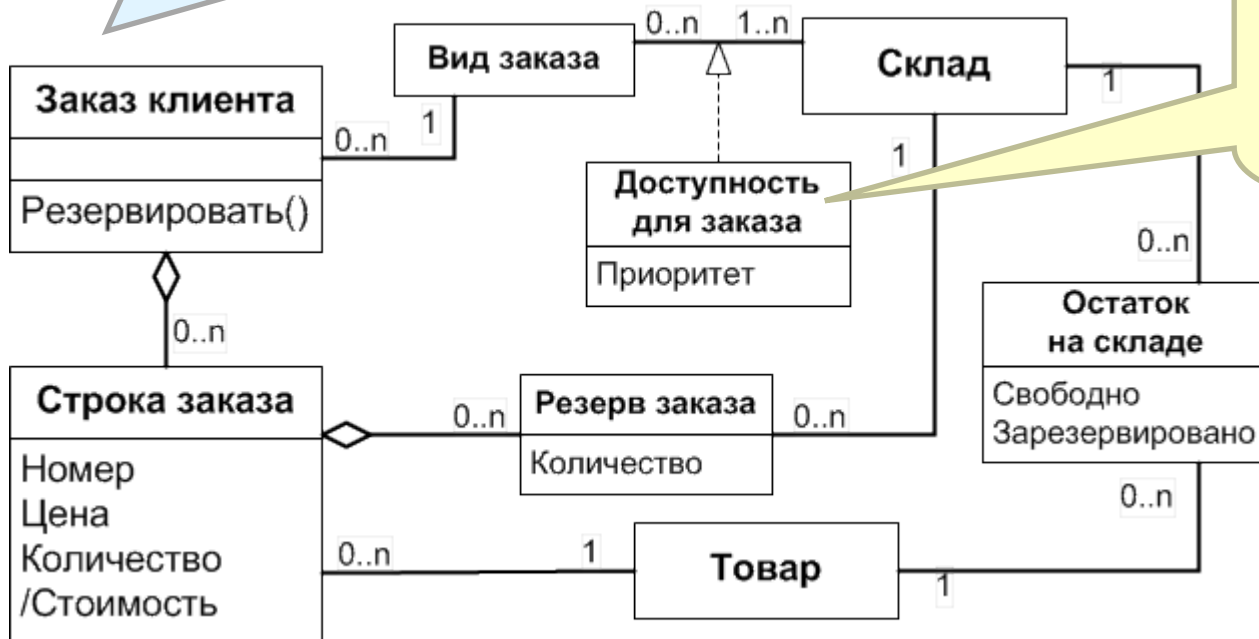
Практика: объектная модель

Изоляция объектов



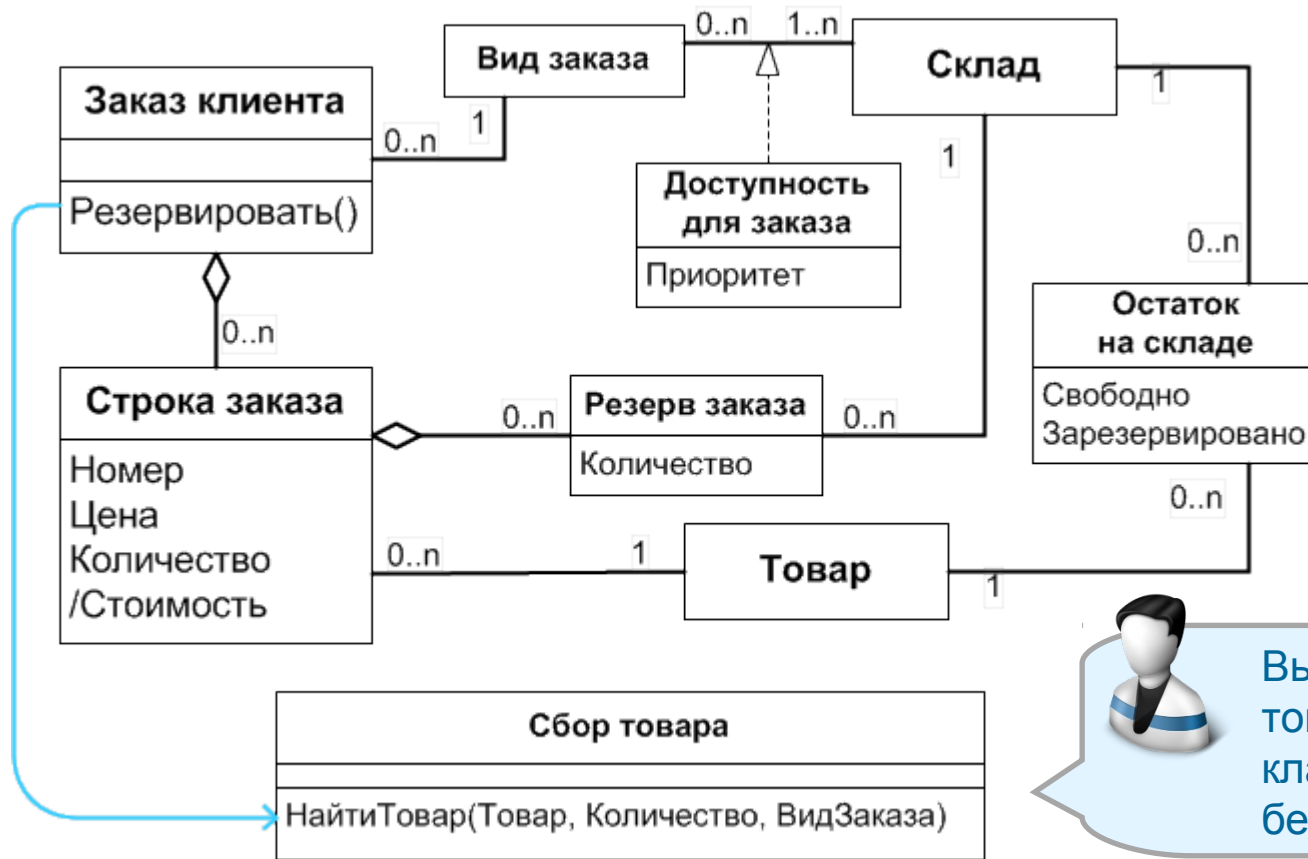
Метод заказа зависит от вспомогательных данных

Делаем данные для гибкой настройки сбора товара



Практика: объектная модель

Изоляция объектов



Как ограничить сложность модели

- ▶ Модель верхнего уровня – основные объекты.
- ▶ Вспомогательные объекты – на схемах фрагментов, они используются только локально.
- ▶ Технические шаблоны в модели не отражаются, достаточно указать использование.

Практика применения DDD при разработке корпоративных приложений

Часть 2. Документооборот –
«почти» объектная модель

Обобщенный документооборот

- ▶ Документ обрабатывается в несколько этапов.
- ▶ На каждом этапе определенные сотрудники могут совершать определенные действия.
- ▶ Для передачи на следующий этап должны выполняться определенные условия.

Модель для документооборота

- ➔ Документу приписываем **состояние**.
- ➔ Состояние определяет этап документооборота:
 - какие действия можно совершать над документом;
 - кто отвечает за обработку документа;
 - кто имеет права на совершение тех или иных действий.
- ➔ Возможные изменения состояний документа образуют **граф переходов**.



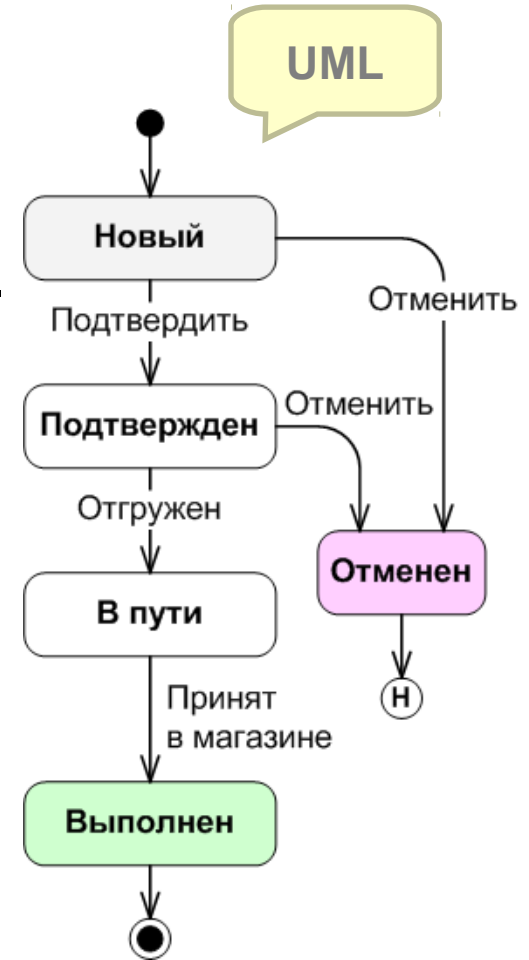
Используем шаблон State Entity.

Язык модели

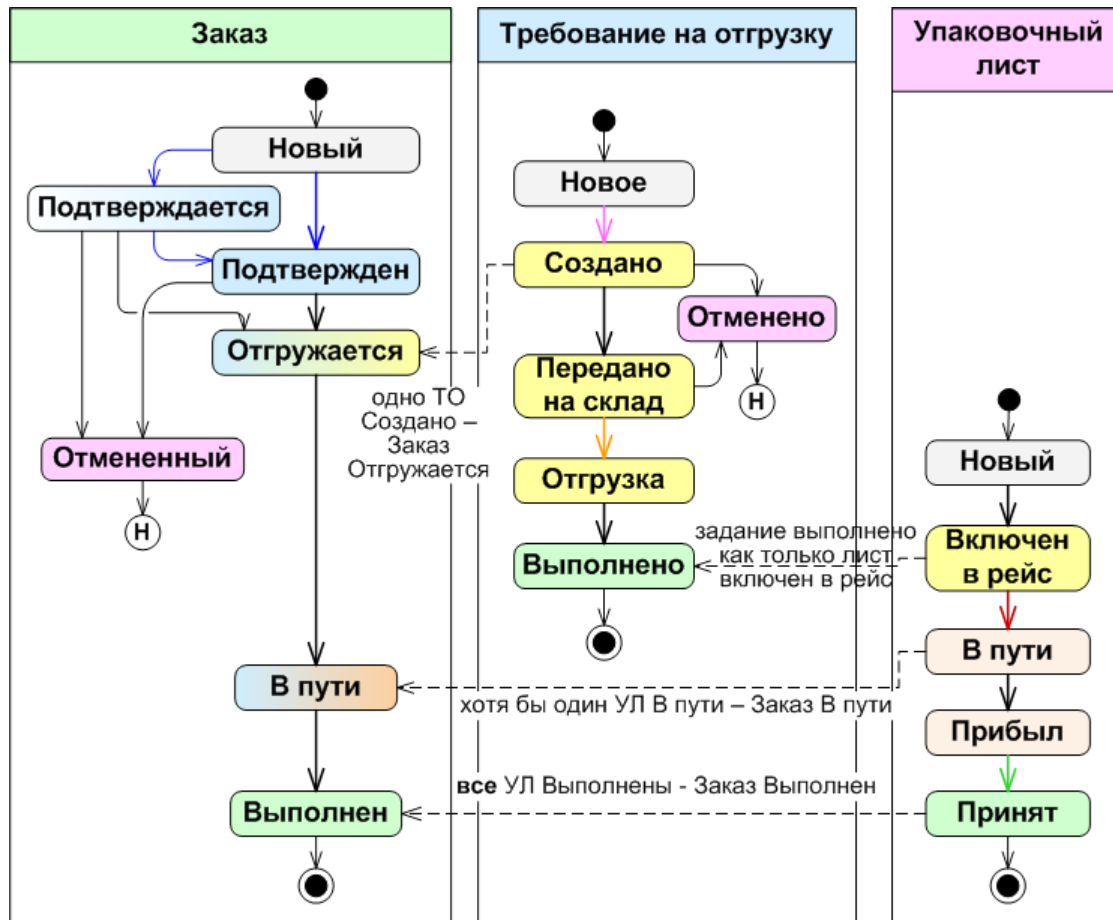
- ➔ Документ – объект предметной области.
- ➔ Действие над ним – вызов его метода.
- ➔ Среди всех методов выделяем переходы и связываем их с состояниями.
- ➔ Граф состояний – State machine diagram.



Язык ООП «с расширениями».
 Названия состояний и переходов –
 на языке бизнеса.



Наглядно представляет сложный документооборот

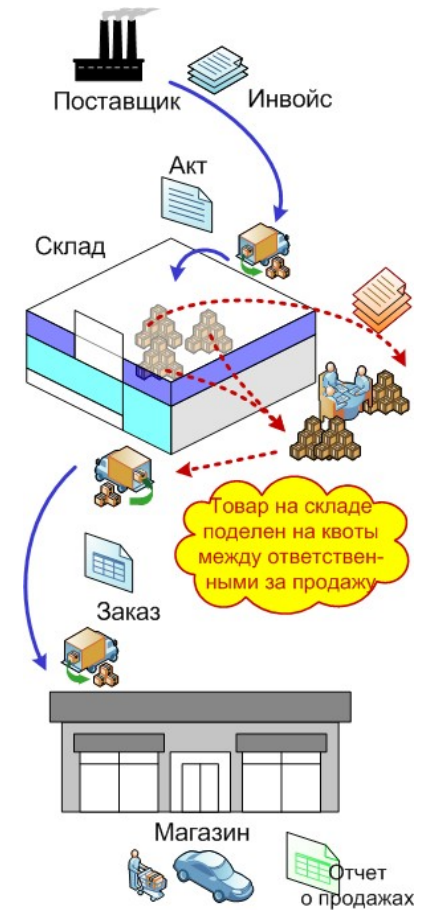


Практика применения DDD при разработке корпоративных приложений

Часть 3. Учет – необъектная модель

Бизнес-задача: учет остатков и потоков товаров, денег, других ресурсов

- ➔ Исполнение документов изменяет учетные показатели.
- ➔ Учетные показатели влияют на обработку документов и решения пользователей.



Нужно в большинстве корпоративных систем, а не только в бухгалтерии.

Примеры показателей и отчетов

- ▶ Показатели: остатки и обороты в разрезе аналитик (товаров, клиентов):
 - остаток на складе по ответственным;
 - поступление товара за период;
 - поставка в магазины за месяц.
- ▶ Отчеты содержат остатки и обороты совместно.

Товар	Было	Пришло	Ушло	Стало
Куртка К12-S	122	40	75	87
Куртка К15-M	187	50	90	147
...				

Язык предметной области

Элементы учета:

- синтетические счета и их аналитические признаки;
- аналитические счета;
- проводки;
- показатели – остатки и обороты на счетах.



Используется общая методология учета,
а не конкретные правила бухгалтерского учета.

Представление учетной модели

Сложность объектного представления учета:

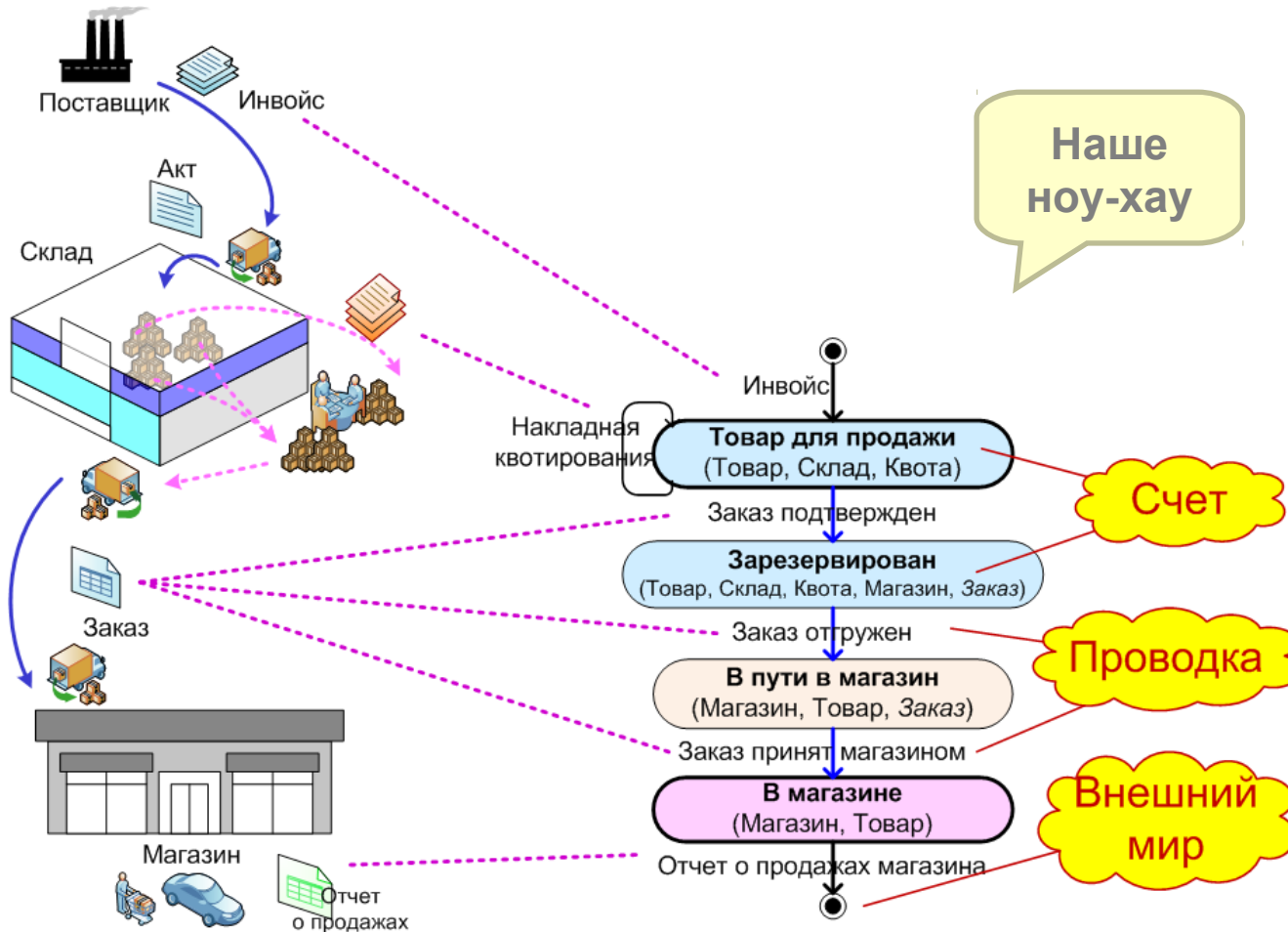
- Нет идентификации единичного объекта.
- Работа идет с показателями, текущее значение которых меняется.
- Изменение числового значения может менять состояние с точки зрения принятия бизнес-решения.
- Часто интерес представляют агрегаты, а не отдельные значения.

Учетная модель – не объектная.



Представление учета оказалось за рамками UML. Для него не придумано эффективного представления.

Диаграммы учета – способ борьбы со сложностью учетной модели



Подробно о диаграммах учета

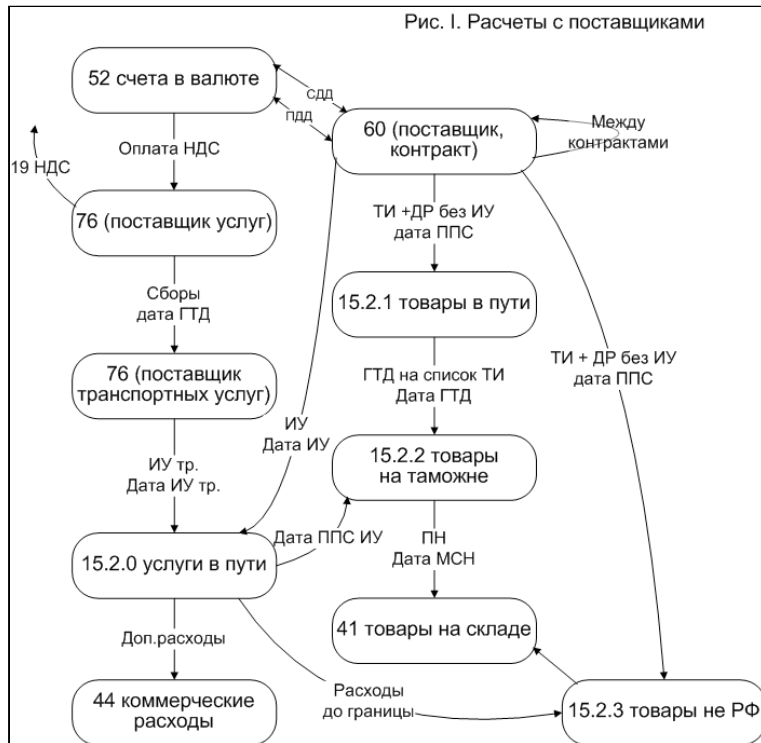
- Презентация и видео «Диаграммы планов счетов – средство моделирования и проектирования учета» ЛАФ-2010 – <http://lib.custis.ru/Accounting-diagrams>
- Презентация и статья «Учет ценных бумаг: Сделать сложное простым» SECR-2010 – <http://lib.custis.ru/Simplify-security-accounting>
- Статья «Диаграммы учета: мост между бухгалтером и разработчиком»
Журнал «Бухгалтер и компьютер» №5 (май) 2011 – http://lib.custis.ru/Когда_всем_понятно



Диаграммы учета для бизнеса

Бухгалтеры могут применять диаграммы учета для разработки учетной политики.

Они нагляднее, чем Excel.



Microsoft Excel - MS90_Отчеты учетных операций_v7.xls

УЧЕТ ОПЕРАЦИЙ С ТОВАРАМИ

№	Учетная операция	договор	договор	договор	договор	договор	договор	договор	договор	договор	договор
13	Зачет оплаты контрагта поставщика товара в рублях	1) Платежка 2) Договор купопродажи товара	02.01.00	07.01.00	05757					Сумма платежа в БК	на Бюджет переводит валют на дату платежа
14	Зачет оплаты контрагта поставщика товара в валюте	1) Платежка 2) Договор купопродажи товара	02.01.00	07.02.00	05757	Валюты = 19	05757			Сумма платежа в 19	на Бюджет переводит валют на дату платежа
16	Зачисление денежных средств по возврату от поставщика в рублях	1) Выписка по счету 2) Договор купопродажи товара	02.01.00	01.01.00	05757	Статус СУЗДС =	05757			Сумма поступления	на СУЗДС переводит валют на дату поступления
17	Зачисление денежных средств по возврату от поставщика в валюте	1) Выписка по счету 2) Договор купопродажи товара	02.01.00	02.01.00	05757	Статус СУЗДС = Валюты =	05757			Сумма поступления	на СУЗДС переводит валют на дату поступления
18	Зачет возврата в рублях на контракт поставщика товара	1) Платежка от поставщика 2) Договор купопродажи товара	02.01.00	07.01.00	05757	Валюты = БК Поставщики + или Виде расчеты = 01	07.01.00			Сумма возврата в БК	на Бюджет переводит валют на дату возврата
19	Зачет возврата в рублях на контракт поставщика товара	1) Платежка от поставщика 2) Договор купопродажи товара	05757	07.01.00	05757		07.01.00			Сумма полученного возврата в БК	на Бюджет переводит валют на дату возврата
20	Клиентская база из перечисленного возврата	1) 77 2) Договор купопродажи товара	01.02.00	01.02.00	05757	Виде расчеты = 24	07.01.00			Сумма комиссии в БК	на Бюджет переводит валют на дату поступления
21	Валютные расходы - оплата в валюте	1) Платежка 2) Договор переводов	02.01.00	01.01.00	05757	Статус СУЗДС =	05757			Сумма платежа RUB	на СУЗДС переводит валют на дату платежа
23	Платеж транспортно-перевозочного в валюте	1) Платежка 2) Договор перевозок	02.01.00	02.01.00	05757	Статус СУЗДС = Валюты = 01	05757			Сумма платежа в БК	на СУЗДС переводит валют на дату платежа
24	Зачет оплаты контрагта доставки в рублях	1) Платежка 2) Договор перевозки	02.01.00	07.01.00	05757	Валюты = БК Поставщики + или	07.01.00			Сумма платежа в БК	на Бюджет переводит валют на дату возврата
25	Зачет оплаты контрагта доставки в валюте	1) Платежка 2) Договор перевозки	02.01.00	07.02.00	05757	Валюты = 19	05757			Сумма платежа в 19	на Бюджет переводит валют на дату платежа
26	Зачет оплаты контрагта доставки в валюте	1) Платежка 2) Договор перевозки	02.01.00	07.02.00	05757	Валюты = БК Поставщики + или	07.02.00	05757		Сумма платежа в БК	на Бюджет переводит валют на дату возврата
27	Зачисление денежных средств по возврату от ТК	1) Выписка по счету 2) Договор перевозки	02.01.00	01.01.00	05757	Статус СУЗДС =	05757			Сумма поступления	на СУЗДС переводит валют на дату поступления
28	Зачисление денежных средств по возврату от ТК	1) Выписка по счету 2) Договор перевозки	02.01.00	02.01.00	05757	Статус СУЗДС = Валюты =	05757			Сумма поступления	на СУЗДС переводит валют на дату поступления
29	Зачисление денежных средств по возврату от ТК	1) Выписка по счету 2) Договор перевозки	02.01.00	02.01.00	05757	Статус СУЗДС = Валюты =	05757			Сумма поступления	на СУЗДС переводит валют на дату поступления
30	Зачет возврата в рублях на контракт поставщика	1) Платежка 2) Договор купопродажи товара	02.01.00	07.01.00	05757	Валюты = БК Поставщики + или	07.01.00	05757		Сумма возврата в БК	на Бюджет переводит валют на дату возврата

Реализация учетной модели

Наш метод

Представить реализацию бизнесу

➤ Взгляд бизнеса:

есть журнал хозяйственных операций,
возникающих при обработке и проведении
документов.

➤ Реализация:

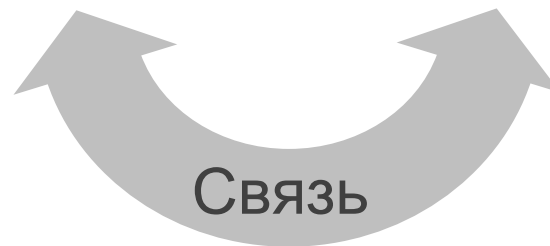
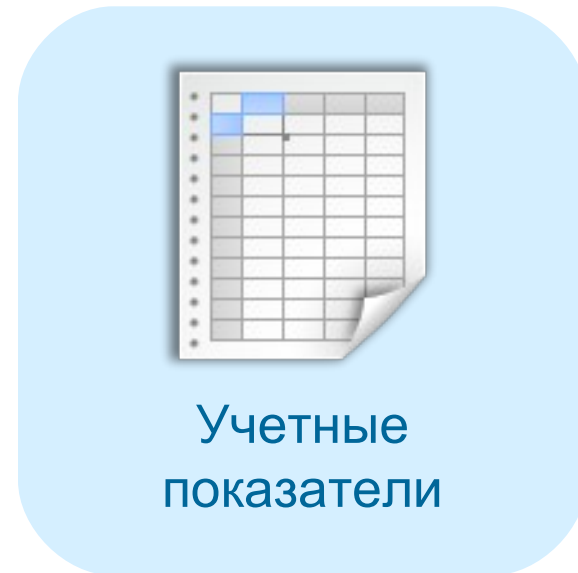
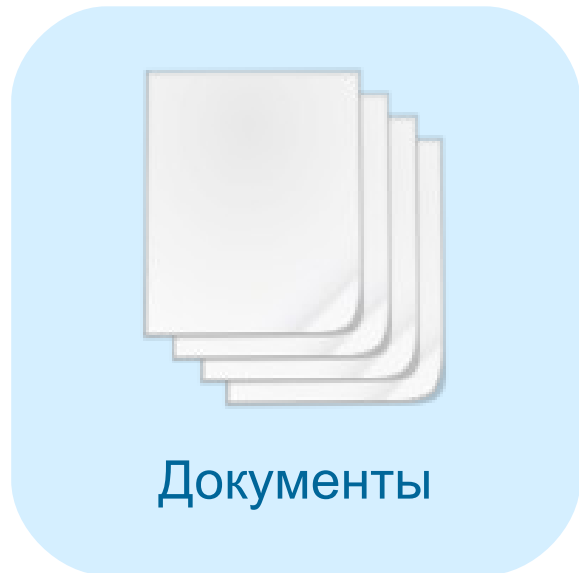
проводки создаются по **событиям** (Event Sourcing,
Фаулер), которые возникают в методах документов.



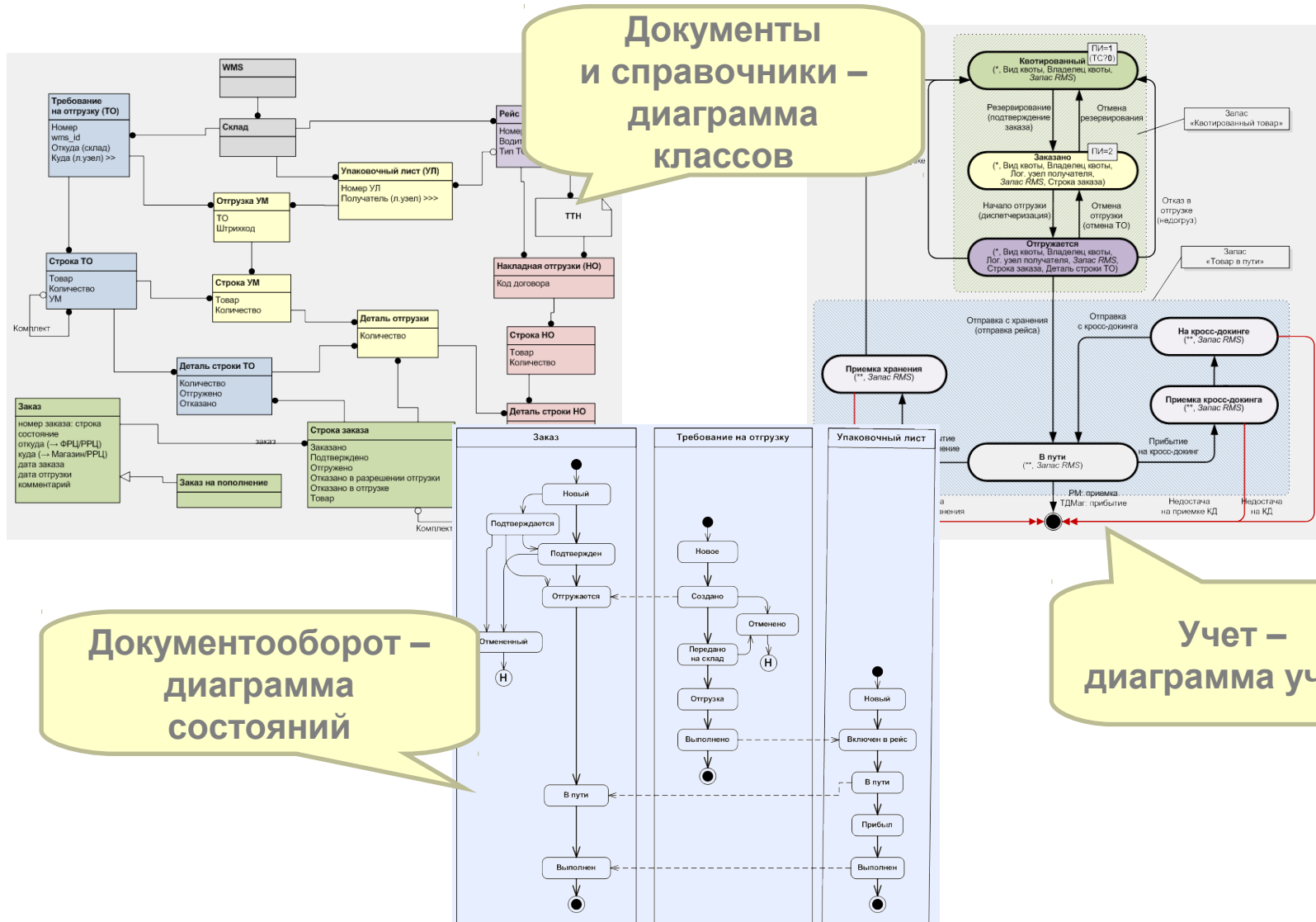
Для прозрачной модели это должно совпадать:
учетные события – суть хозяйственные операции.

Подводя итоги

Единая модель системы



Единая модель системы



Документы и справочники – диаграмма классов

Документооборот – диаграмма состояний

Учет – диаграмма учета

Почему DDD – эффективно?

Единый язык + Единая модель:

- ▶ позволяют эффективно **развивать** сложную систему;
- ▶ обеспечивают понимание всем участникам проекта;
- ▶ избавляют от поддержки нескольких моделей;
- ▶ поддерживают **итеративное проектирование** и разработку;
- ▶ применяются для разных парадигм моделирования.



DDD не убирает системную сложность, а позволяет эффективно работать при ее наличии.

Спасибо!

Вопросы?

Максим Цепков (M.Tsepkov@custis.ru)