



DDD: реализуем проект «Вавилонская башня»

Докладчик:

Максим Цепков (M.Tsepkov@custis.ru)

www.custis.ru

Международная конференция

Software People 2012

10-11 апреля, Москва

О чем этот доклад?

- ▶ Когда-то давно люди не смогли построить Вавилонскую башню – Бог смешал их языки
- ▶ Проекты в IT в чем-то похожи – они рушатся из-за отсутствия понимания между бизнес-экспертами, разработчиками и другими участниками
 - Постановки не отражают потребности
 - Готовая система не соответствует постановкам



DDD – способ решить проблему понимания
и ключ к построению сложных систем

Схема доклада

Теория

DDD – **единый язык проекта** и одна модель системы

- Модели были давно, но две: бизнес-область и система
- Единый язык проекта создает общее поле понятий
- И позволяет работать с одной, общей моделью

Практика

- Что значит «общий язык проекта»
- Какие это дает преимущества
- И какую цену приходится платить

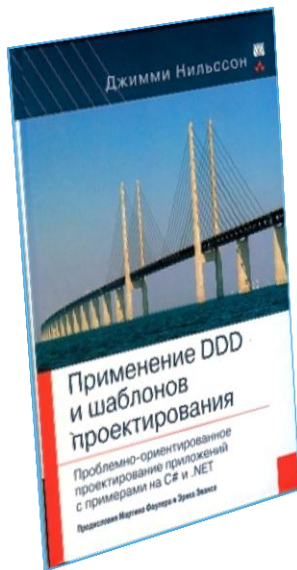


Единый язык на практике –
как это и что в результате?

Немного теории

DDD – как оно начиналось

- 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!)

Единый язык (ubiquitous language)

- 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



Понятия единого языка:

Клиент, Накладная, платеж, Долг –
из предметной области

А где здесь ООП ?



ООП – это парадигма моделирования

Парадигма моделирования определяет

- Элементы языка
- Способ их соединения в сложные конструкции
- Визуальный образ для представления
- Способ отражения модели в код

Объекты с атрибутами и методами

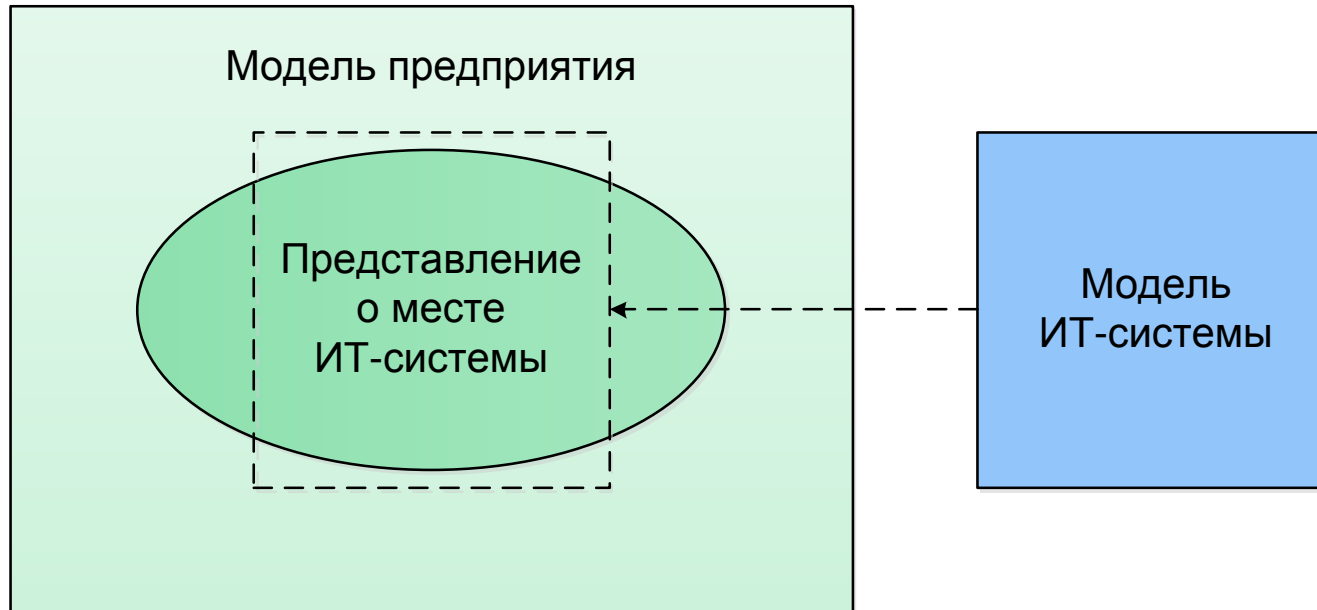
Диаграмма классов и другие UML-диаграммы

Типы, соответствующие бизнес-объектам



Я сосредоточусь на разработке модели, а не на ее реализации

Зачем нужен единый язык?

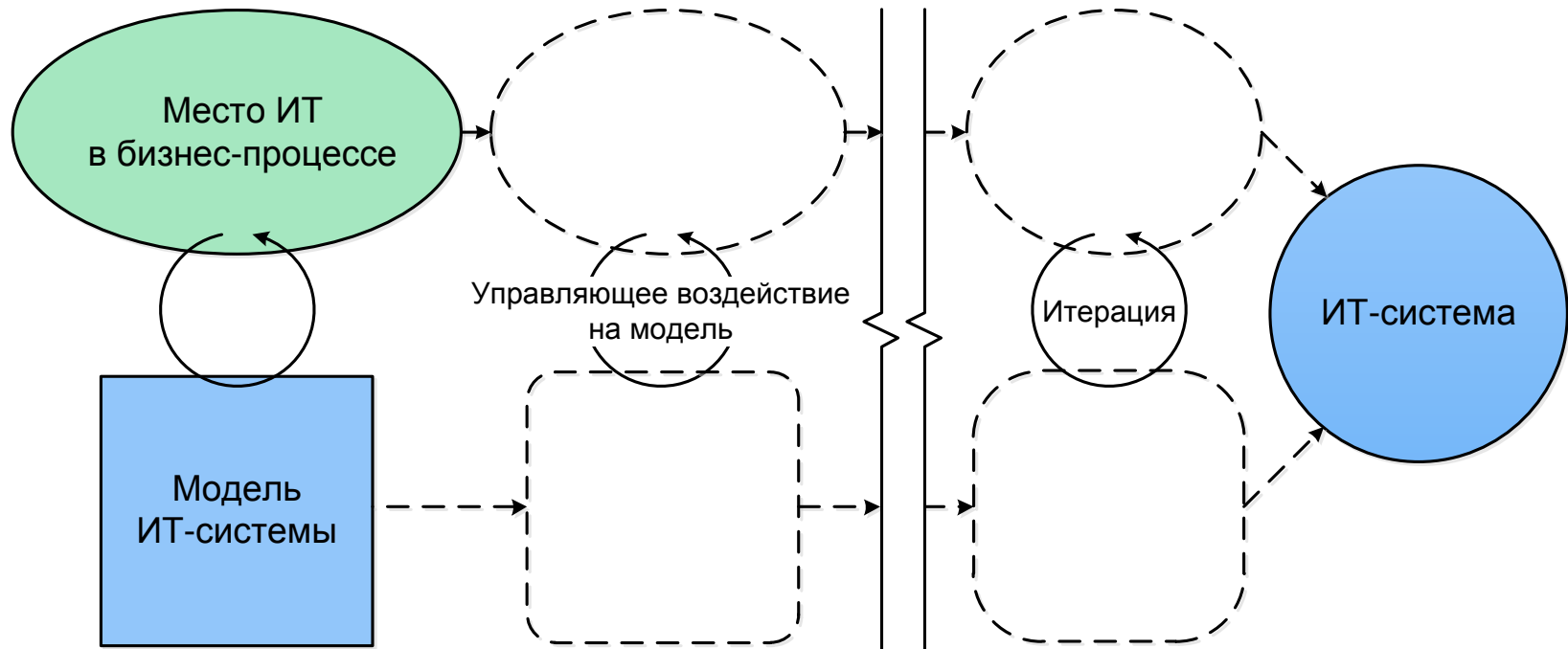


Модель системы не соответствует представлению бизнеса о ее месте в модели предприятия.

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



Итерационное развитие модели



Единый язык позволяет совместить модель системы с представлениями бизнеса о ее месте

Единая модель

- **Аналитик** собирает требования и строит модель – сначала предметной области, затем – системы
- Артефакты модели описывают и систему и ее использование в бизнес-процессах предприятия
- **Разработчик** реализует модель
- Артефакты модели можно проследить в коде



Модель предметной области
становится моделью системы

Что мы достигаем?

- ☺ Верификация постановок бизнес-специалистами
- ☺ Общее понимание требований на стороне бизнеса
- ☺ Обсуждение модели бизнесом и IT, поиск баланса в сложных решениях
- ☺ Перенос моделей из других предметных областей
- ☺ Бизнес представляет потенциальные возможности системы и сложность различных доработок
- ☺ На этапе эксплуатации – эффективное общение бизнес-пользователей и IT без квалифицированных переводчиков-аналитиков

Автоссылки по теме...

SECR-2011 –

[DDD и системная сложность](#)

ADD-2011 –

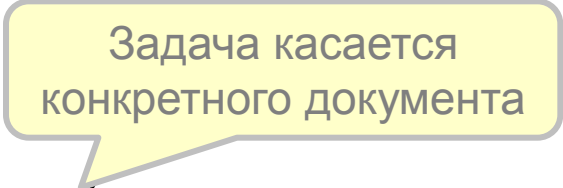
[Необъектные модели предметной области](#)

SoftwarePeople-2011 –

[Три компоненты архитектуры приложений](#)

Пример-1: Виза на документы

Задача



Задача касается
конкретного документа

В процессе обработки документа (сделки, договора, контракта) обеспечить проверку и одобрение определенными сотрудниками или отделами

Выбор проектного решения



В чем проблема?

- ▶ Шаблоны обладают разной гибкостью и сложностью
- ▶ Для выбора нужно понимать требуемый баланс между гибкостью и сложностью решения

Традиционный подход

- ▶ На этапе сбора требований аналитики формулируют задачу для конкретного документа
- ▶ Исходя из этого в системе проектируется решение
- ☹ Выбранное решение отражает текущую ситуацию

Решение надо принимать с учетом потенциального изменения бизнес-процессов

Примеры

- ▶ Проверка сделки казначейством и отделом рисков – не получается выполнять параллельно
- ▶ Юристы отзывали одобрение кредита, а служба безопасности не него опиралась – связи между визами не контролируются системой
- ▶ Настройку виз для одобрения договоров с недвижимостью передали в IT из-за сложности

Решение в рамках DDD

- ▶ Представляем шаблоны, описанные применительно к конкретному документу, показываем разницу
- ▶ Бизнес-специалисты оценивают потенциальную потребность в реализации бизнес-процессов
- 😊 Решение принимается с учетом перспективы

В чем Единый язык?

- ▶ Для решения модель надо описать понятно бизнесу
 - Можно описать обобщенные решения для документов, «состояния» и «визы», и на них ссылаться
 - Можно описывать каждый случай отдельно
- ▶ Термины должны быть понятны Заказчику:

Например, «визированием» могут считать одобрение документа, требующее только просмотра, а если требуется дополнительная работа, то это называется «обработкой» или «проверкой»
- ▶ Общий шаблон надо «перевести» на язык проекта

Результат

- 😊 Мы используем известные шаблоны решений
- 😊 Заказчик оценивает вариант решения не только с точки зрения текущих потребностей, но и из предположений о развитии бизнес-процессов
- 😊 Проектируя изменения бизнес-процессов, заказчик представляет потенциал гибкости системы и принимает решения с учетом этого

Пример-2: Долг клиента

Задача

Ведение взаиморасчетов с клиентами по продажам

- ▶ обеспечивающее ведение управленческих лимитов
- ▶ и отражение расчетов в бухгалтерию

Проблема:

Смещение языков на бизнес-уровне

Менеджеры по продажам: долг – основа для управленческих решений. Увеличивается как только приняли решение об отгрузке, уменьшается когда признали претензию



Бухгалтеры: долг – в соответствии с ПБУ, с учетом оформления документов и прохождения процедур



Следствие - управленческий и бухгалтерский долг имеют систематические различия

ПРОЦЕСС ОТГРУЗКИ ТОВАРА



Проблемы двух пониманий долга

- ☹️ Контроль правильности учета – опаздывает
- ☹️ Менеджеров не получается контролировать через бухгалтерию
- ☹️ Имеются две разные суммы долга, что затрудняет принятие управленческих решений
- ☹️ Для сверки с клиентом и решения проблем менеджеры должны вникать в ПБУ

Решение от DDD

- ▶ Вырабатываем единый язык, описывающий долг в терминах, понятных менеджерам и бухгалтерам
 - ▶ Строим модель, которая показывает управленческий и бухгалтерский долг и их различие
 - ▶ Ситуации, в которых долг различается описываем на едином языке, согласуем со специалистами
 - ▶ Вырабатываем требования по контролю различий, а также по устранению несущественных различий
- 😊 Результат – общий взгляд на предметную область у бизнес-специалистов, описанный в модели, которая реализуется разработчиками

Как описывать учетную модель?

- ▶ Для наглядного описания мы разработали Диаграммы учета
- ▶ Они позволяют представлять учетные модели и делают описание их понятным не только для бухгалтеров, финансистов и аналитиков, но и для других бизнес-специалистов, а также разработчиков
- ▶ При построении моделей можно успешно применять типовые шаблоны, заимствованные из бухгалтерии, например, транзитные счета

Подробнее о диаграммах учета – статья «Диаграммы учета: мост между бухгалтером и разработчиком»

Журнал «Бухгалтер и компьютер» 5-2011 http://lib.custis.ru/Когда_всем_понятно

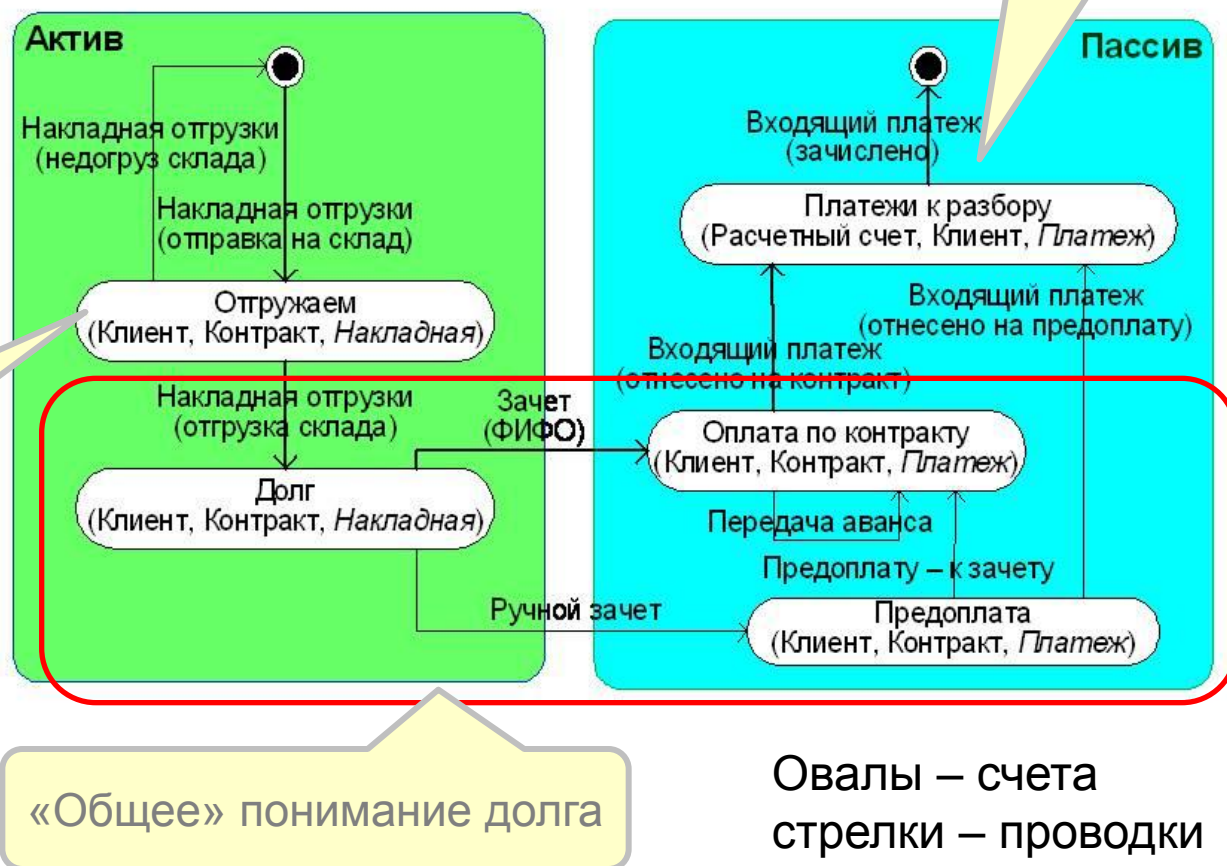
Модель долга клиента



Сверку с клиентом успешно
выполняют менеджеры

Этих платежей **нет**
у менеджеров

Этого долга **нет**
у бухгалтеров



Результат

- 😊 Согласовано понимание предметной области у различных бизнес-специалистов
- 😊 Реализована модель, отвечающая интересам всех заинтересованных сторон проекта

Пример-3: Коммунальный биллинг

Задача

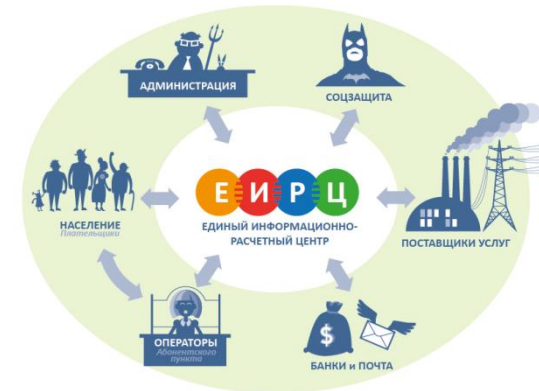


Система расчетного центра
по коммунальным услугам



Проблема: Сложность задачи превышает уровень пользователей

- Система решает сложную учетную задачу, включая изменения в прошлом
- Ее расчеты должны быть понятны широкому кругу пользователей без опыта работы с учетом
 - Населению, получающему квитанции
 - Операторам абонентских пунктов
 - Сотрудникам службы соц.защиты
 - Бухгалтерам поставщиков услуг
 - Сотрудникам городского управления
- Сопровождение системы, включая создание отчетов, должно происходить на местах



Решение от DDD

- ▶ Создается учетная модель с использованием шаблонов, обеспечивающих необходимый учет
- ▶ Модель описывается на доступном пользователям языке, это обеспечивает понимание
- ▶ Реализуются интерфейсы, позволяющие проследить разбор по модели конкретного случая

Решение от DDD



Учетная модель расчетов
сделала систему понятной



В модели использованы
приемы из бухгалтерии,
обеспечившие требуемый учет

Интерфейсы представляют
учет понятно для пользователя

Заключение

Что же обеспечивает DDD?

Единый язык + Единая модель:

- обеспечивают надежную основу для общения всех участников проекта при принятии решений;
- позволяют использовать типовые шаблоны;
- позволяют эффективно развивать сложную систему.

Это требует дополнительных усилий:

- формирование единого языка;
- понимание разработчиками предметной области.

По опыту, результат окупает усилия.



DDD позволяет успешно создавать сложные проектные решения и развивать их

Спасибо!
Вопросы?

Максим Цепков

M.Tsepkov@custis.ru



CUSTIS®