SOFTWARE PEOPLE

CUSTIS®

DDD: проблемы и решения

при отражении модели предметной области в код

Максим Цепков,
главный архитектор,
группа компаний CUSTIS

CUSTIS®

О чем этот доклад?

В применении любой методологии есть этапы

- Смотрим примеры – в них все хорошо
- Применяем, решая возникающие проблемы
- Осмысливаем опыт решения, развивая метод

DDD – не исключение

В докладе речь пойдет о нашем опыте
решения проблем, возникающих
при применении DDD в сложных проектах

CUSTIS®

План доклада

1. Идея и внутренние противоречия DDD
2. Проблемы при использовании DDD
3. Технологические рельсы – наш способ решения
4. Адаптация процесса разработки
5. Достигнутые преимущества

CUSTIS®

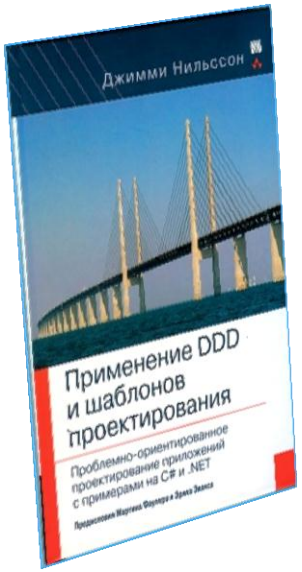
Идея
и внутренние противоречия DDD

CUSTIS®

DDD – как оно начиналось

Концептуальная книга Эрика Эванса

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



Практическая книга Джимми Нильссона

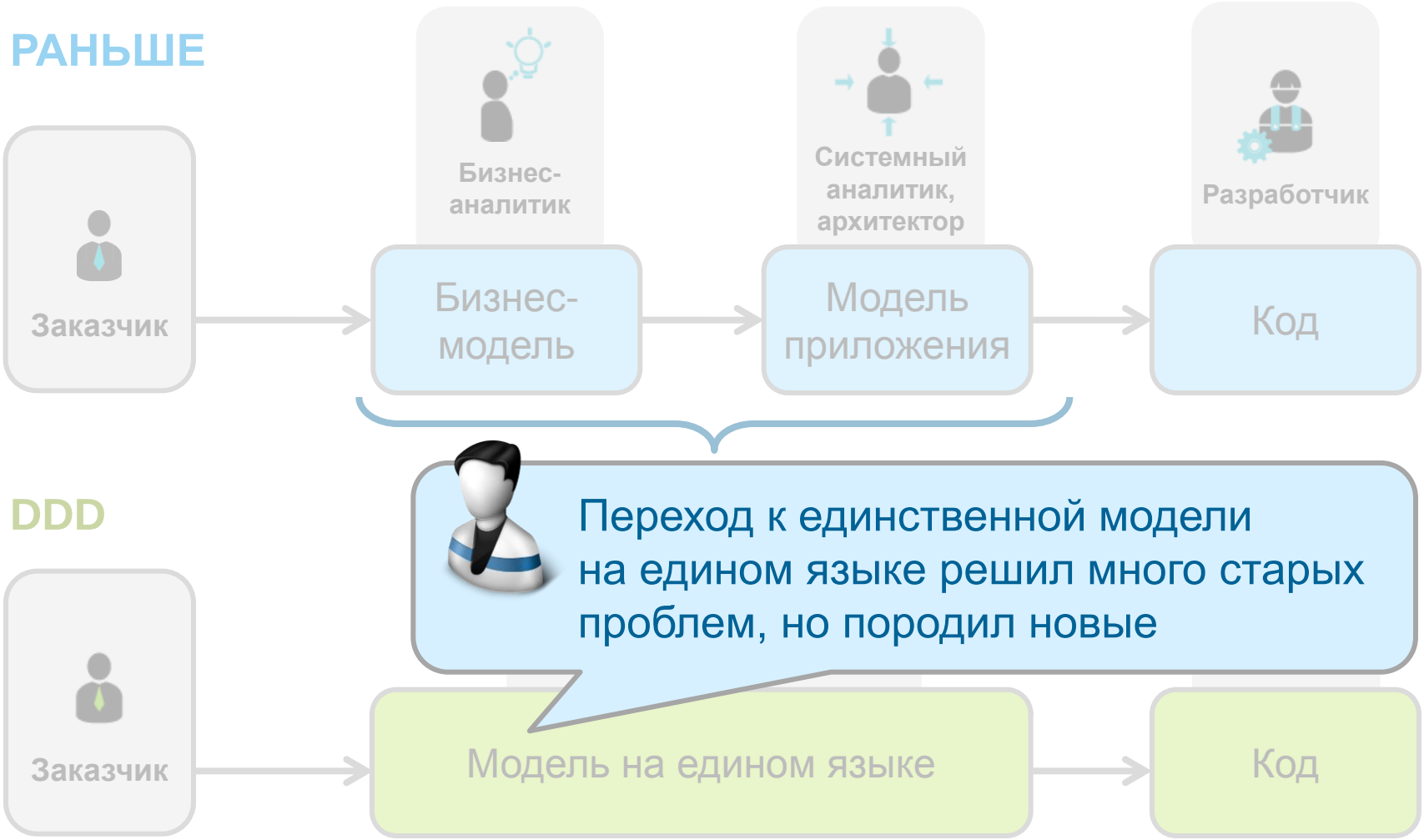
- на английском – в 2006 г.
- на русском – в 2007 г. (почти сразу!)

CUSTIS®

Основная идея DDD

CUSTIS®

Основная идея DDD



CUSTIS®

Плюсы модели на едином языке

- Единое поле коммуникаций для всех участников проекта
- Верификация модели заказчиком и выявление наиболее критичных ошибок на этапе постановок
- Простота внесения изменений в требования: их не надо протаскивать через несколько моделей
- Гибкость реагирования на ограничения, выявляемые при разработке: их можно передать заказчику посредством модели и найти решения

CUSTIS®

Подробнее о DDD

О преимуществах единственной модели на едином языке я рассказывал в докладах на конференциях:

- «DDD: Реализуем проект Вавилонская башня» (Software People – 2012)
- «DDD — эффективный способ работы в условиях системной сложности» (SECR – 2011)

CUSTIS®

Проблемы при использовании одной модели

- Необходимость понимания модели заказчиком существенно ограничивает моделирование
- Технические подробности и сложные формальные методы становятся недопустимы
- А без них модель не может служить проектом для реализации, нужно дополнительное проектирование

Единственная модель на едином языке –
и преимущество DDD, и источник проблем

CUSTIS®

Стандартные пути решения проблем

- ООП – общепринятый и (на простом уровне) интуитивно понятный метод
- Модель, сделанная в объектной парадигме, может быть представлена заказчику и согласована с ним
- Современные языки поддерживают ООП, модель хорошо реализуется с использованием шаблонов Domain model и Rich objects

Этот способ **работает,** но **ограниченно.** Простого ООП не хватает, а изучать сложный заказчики не считают правильным

CUSTIS®

Простое применение DDD

- На основе требований создаем объектную модель
  - структура классов
  - поведение объектов
  - интерфейсы
- Согласуем ее с заказчиком
- Реализуем в коде на основе шаблонов Domain model и Rich objects

Что получается?

CUSTIS®

Проблемы при использовании DDD
Практические аспекты

CUSTIS®

Большие и сложные объекты

- Бизнес-объект включает все аспекты деятельности
  - Клиент – как субъект делового мира, партнер по сделкам, взаимоотношения юридические и управленческие
  - Заказ – полный жизненный цикл от начальных переговоров до исполнения, включая юридическое и другое оформление
- Бизнес-объекты тесно связаны между собой

☹ При отражении в IT-объекты получается очень похоже на антипаттерн Big Object

☹ Сложно понимать, развивать, тестировать

CUSTIS®

Техническое усложнение модели

- Принципы ООП побуждают к декомпозиции, что увеличивает число объектов
- Применение шаблонов (стратегии и др.)
- Технические и инфраструктурные атрибуты и классы
- Ограничения на объекты со стороны фреймворков и обходные пути для их преодоления

☹ Сложная модель не понимается заказчиком, простая – не отражает реализацию

CUSTIS®

Ограниченная область DDD-модели

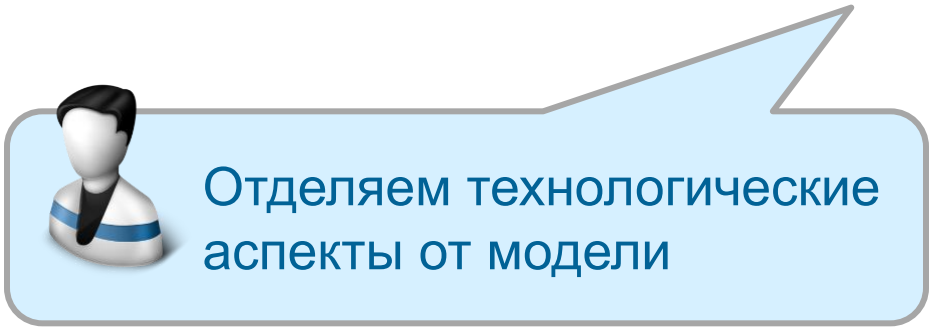
- Подход разрабатывался для слоя бизнес-логики
- Использование объектных языков побуждает ограничиться объектными моделями

Хочется

- Использовать модель при проектировании интерфейсов
- Расширять способы моделирования

CUSTIS®

Решение – технологические рельсы



CUSTIS®

Что такое технологические рельсы

Технологические рельсы – это способ перевода модели на едином языке в код

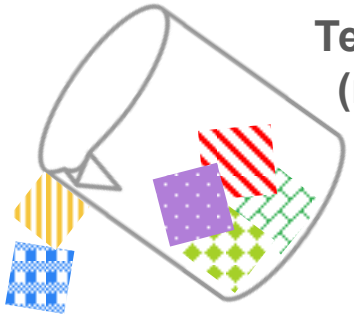
- Платформы, фреймворки и библиотеки
- Способы их организации в единое приложение
- Способы отражения модели приложения в код
- Типовые задачи и design patterns для их реализации

Формулируются на всех уровнях – от архитектуры до частных задач посылки сообщения

«Применяем Domain model и Rich Objects» – частный случай технологических рельсов

CUSTIS®

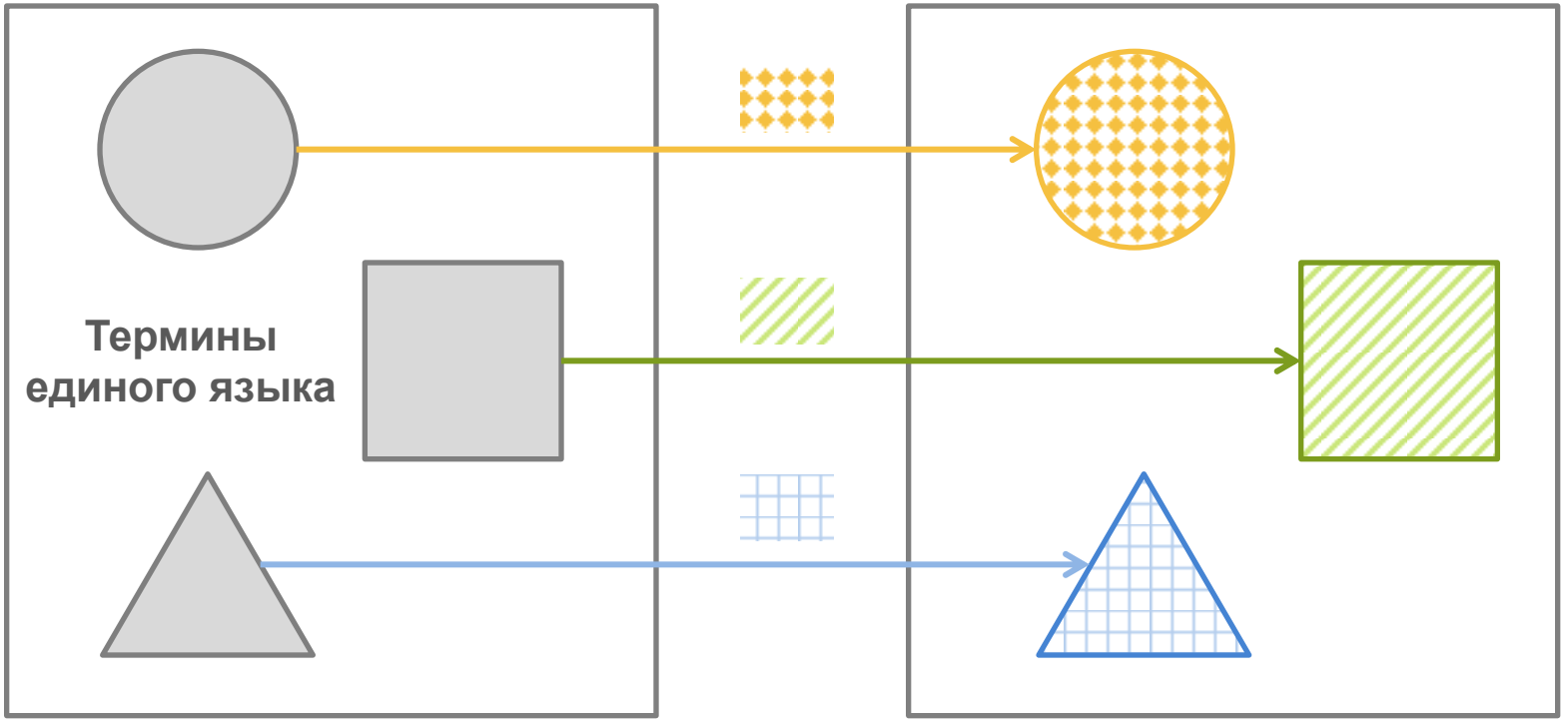
По рельсам ☺



Технологические рельсы
(шаблоны реализации)

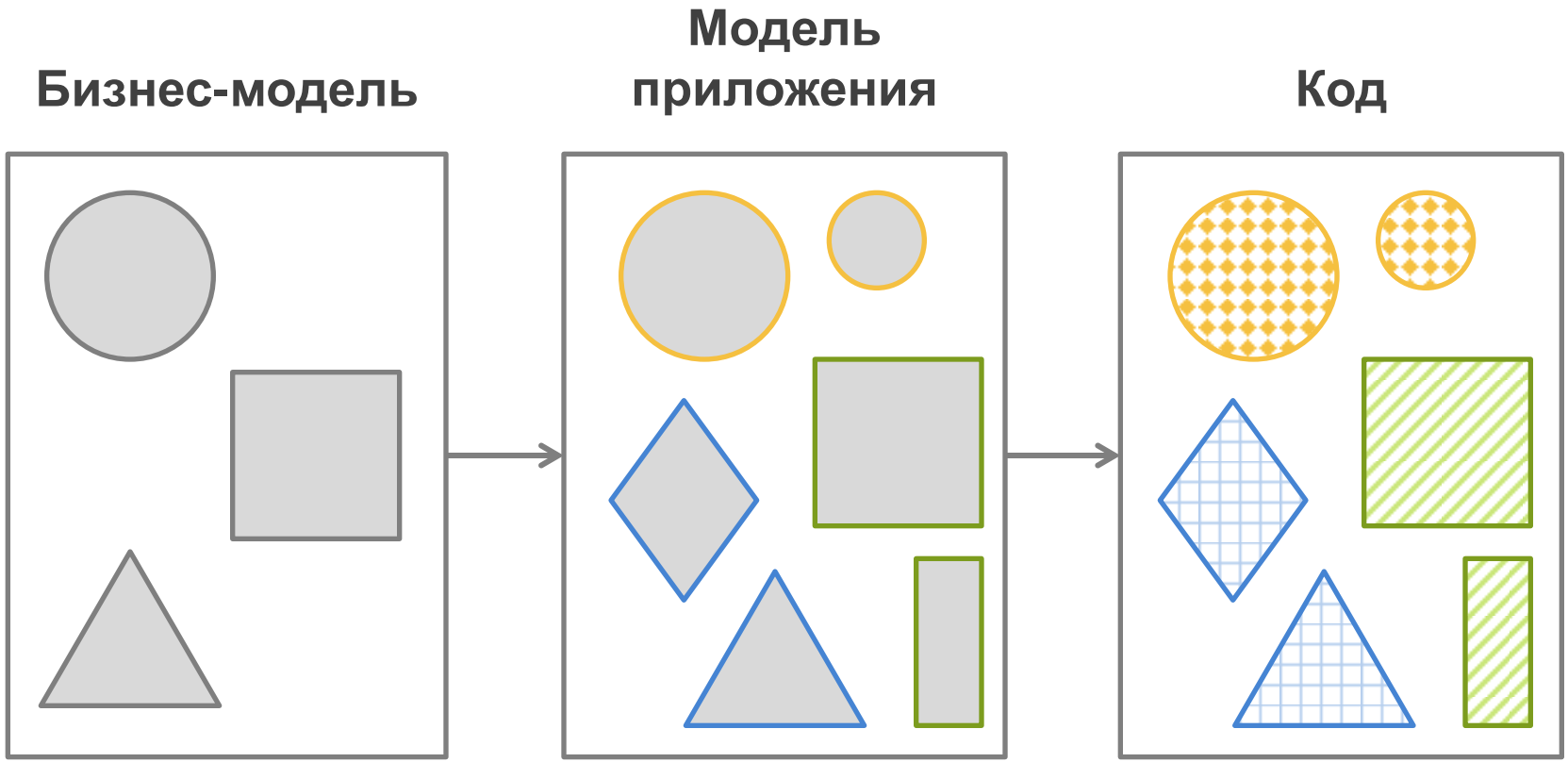
Модель
на едином языке

Код



CUSTIS®

А как без рельсов



CUSTIS®

Виды технологических рельсов

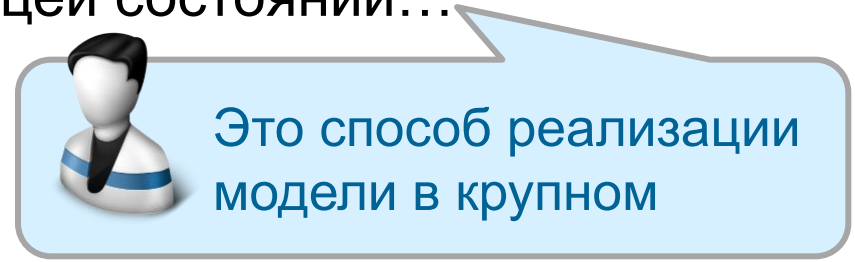
- Архитектурные
- Инфраструктурные
- Рельсы предметной области

CUSTIS®

Архитектурные рельсы

Структура приложения в крупном, например:

- Java с Hibernate и Spring
- Создаем anemic-объекты с разметкой Hibernate для каждого бизнес-класса, используем их так же, как DTO для интерфейса
- Бизнес-логику каждого класса реализуем в отдельном статическом классе
- Документооборот описываем через состояния объекта, для реализации используем StateEntity с декларативной таблицей состояний…
- …



CUSTIS®

Инфраструктурные рельсы

Способы реализации стандартных задач: печатных форм, отправки сообщений, интеграции, например:

- Все печатные формы реализуются с помощью библиотеки X
- Доступ к библиотеке – через сервис-локатор
- Для бизнес-объекта с печатной формой создаем класс PrintForm<TClass>, в котором каждой форме соответствует отдельный метод…
- На интерфейсе команды печати появляются автоматически, исходя из созданных классов…

CUSTIS®

Рельсы предметной области

Способ реализации конструкций, характерных для предметной области проекта, например документа с товарными строками:

- Строки документов хранятся в отдельных таблицах, однако в объектной модели доступ к ним – исключительно через шапки документов
- Для реализации типовой работы класс документа наследуем от WareDoc<TDocHeader, TDocItem>, что обеспечивает стандартный функционал…
- Если документ не хранит цены и стоимость, а берет их из справочников, реализуем стратегию PriceCalc
- …

CUSTIS®

Как технологические рельсы решают описанные проблемы?

CUSTIS®

Рельсы для печатной формы

Задача: Для накладной надо печатать форму ТОРГ-12

Задача – типовая, решение – единообразное

Требования к решению:

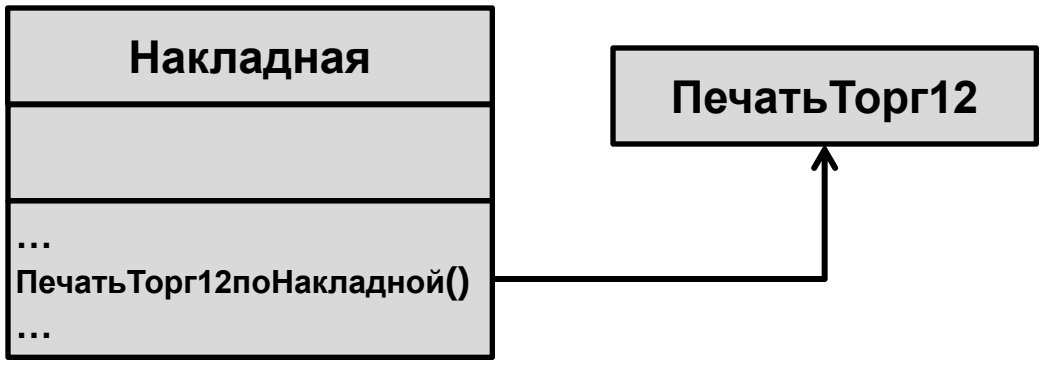
- Отделить печать от бизнес-логики
- Обеспечить независимое тестирование
- Желательно обобщенное решение

Декомпозиция кода и независимое тестирование – признаки хорошего стиля

CUSTIS®

Печать форм: решение 1

Обобщенный сервис ПечатьТорг12
и метод НапечататьТорг12 у класса Накладная

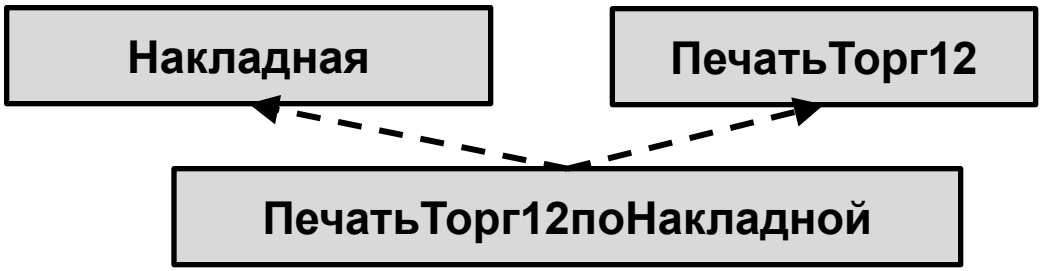


- ☺ Простое и очевидное решение
- ☹ Увеличивается класс Накладная, в нем смешивается разнородная бизнес-логика
- ☹ Класс Накладная зависит от сервиса печати, это сильно затрудняет тестирование

CUSTIS®

Печать форм: решение 2

Обобщенный сервис ПечатьТорг12 и сервис ПечатьТорг12поНакладной

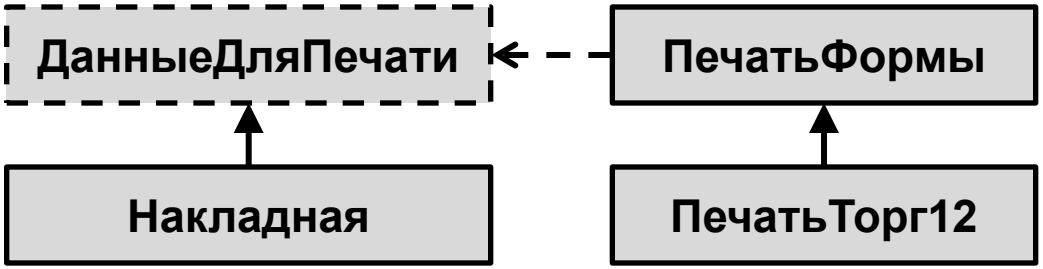


- ☺ Логика печати вынесена из бизнес-объектов
- ☹ Для тестирования ПечатьТорг12поНакладной необходима Накладная со всей инфраструктурой
- ☹ Таких классов печати становится довольно много

CUSTIS®

Печать форм: решение 3

Обобщенный сервис ПечатьФормы и сервис ПечатьТорг12поНакладной



☺ Логика печати вынесена из бизнес-объектов

☺ Печать и бизнес-логика тестируются независимо

Решение применяем ко всем задачам печати, тогда в модели достаточно перечислить формы

CUSTIS®

Расширение модельного поля

- Диаграммы состояний для документооборота
- Использование модели для интерфейсов
- Задачи ведения учета

CUSTIS®

Диаграммы состояний

- У класса-документа есть атрибут **состояние**, описывающий текущее место в документообороте
- В модели для документов создаем диаграмму состояний, на ней же показываем роли
- В классе для каждого перехода реализуем метод, помечая его метаданными
- В начале и в конце метода перехода вызываем PreTransit и PostTransit, которые обеспечивают логирование и контролируют состояние документа

CUSTIS®

Модель для интерфейсов

- Реализуем конструктор обобщенных интерфейсов на основе метаданных бизнес-объектов – таблиц с фильтрами для просмотра, диалогов для изменения
- Обеспечиваем точки расширения для реализации дополнительной логики

☺ В постановках на 90% интерфейсов – списки полей

☺ Реализация требуется только для особых случаев

CUSTIS®

Задачи ведения учета

- Присутствуют в большинстве enterprise-приложений
- Плохо описываются в терминах объектов

Решение

- Специальные диаграммы учета
- Базовое учетное ядро – обобщенный учет
- Шаблон реализации диаграмм учета на учетном ядре

☺ Учет описывается в адекватных терминах

☺ Диаграммы обеспечивают эффективное понимание

Подробнее о диаграммах состояний и реализации учета –
в докладе на ADD-2011 «Необъектные модели предметной области»

CUSTIS®

Технологические рельсы и процесс разработки

CUSTIS®

Аналитик и архитектор

- В небольших проектах аналитик и архитектор – один человек
- В крупных проектах аналитик создает модель, а архитектор проектирует техническую реализацию
- В классическом процессе каждый из них работает со своей моделью
- В DDD – совместная работа над одной моделью

☹ Конфликты – ответственность плохо разграничена

CUSTIS®

Разработка «с рельсами»

- Аналитик – создает модель и согласует ее
- Архитектор – отвечает за технологические рельсы, которые обеспечивают эффективную реализацию
- Конструкции единого языка – способ синхронизации модели с технологическими рельсами

☺ Разграничение ответственности

☺ Эффективность проектных решений обеспечивается технологическими рельсами

☺ Параллельная работа над проектом

CUSTIS®

Подводя итоги

CUSTIS®

Что достигнуто?

Технологические рельсы:

- сочетают описание модели в понятных заказчику терминах со сложными техническими решениями
- помогают соблюдать ограничения, накладываемые фреймворками, платформами и библиотеками на организацию программного кода, поддерживают их совместную работу и однородность решений в рамках проекта

Все это обеспечивает эффективную разработку и является залогом долгого и успешного развития ИТ-системы, уже находящейся в эксплуатации

Спасибо!
Вопросы?

Максим Цепков
M.Tsepkov@custis.ru