

Process & Case Management в информационной системе: от автоматизации As Is к поддержке развития бизнеса

Максим Цепков

Соучредитель и главный архитектор

29 октября 2016

О чем будет доклад

- ▶ Классический подход к организации бизнеса – создание бизнес-процессов и поддержка их информационной системой, Process Management
- ▶ В **любом** процессе есть много **особых случаев**
- ▶ Включение их как ветвей процесса ведет к **чрезмерной сложности** процесса и софта, поэтому особые случаи стремились исключить
- ▶ **Тренд персонализации** влечет увеличение количества особых случаев



Решение: объединив Process и Adaptive Case Management, обеспечить **простоту и гибкость**
Бонус – поддержка **развития бизнеса**

Для кого этот доклад

- ▶ Для архитекторов – они проектируют информационные системы для бизнеса
- ▶ Для аналитиков – они определяют способы поддержки бизнеса софтом
- ▶ Для разработчиков, которым важно, чтобы их софт реально помогал бизнесу

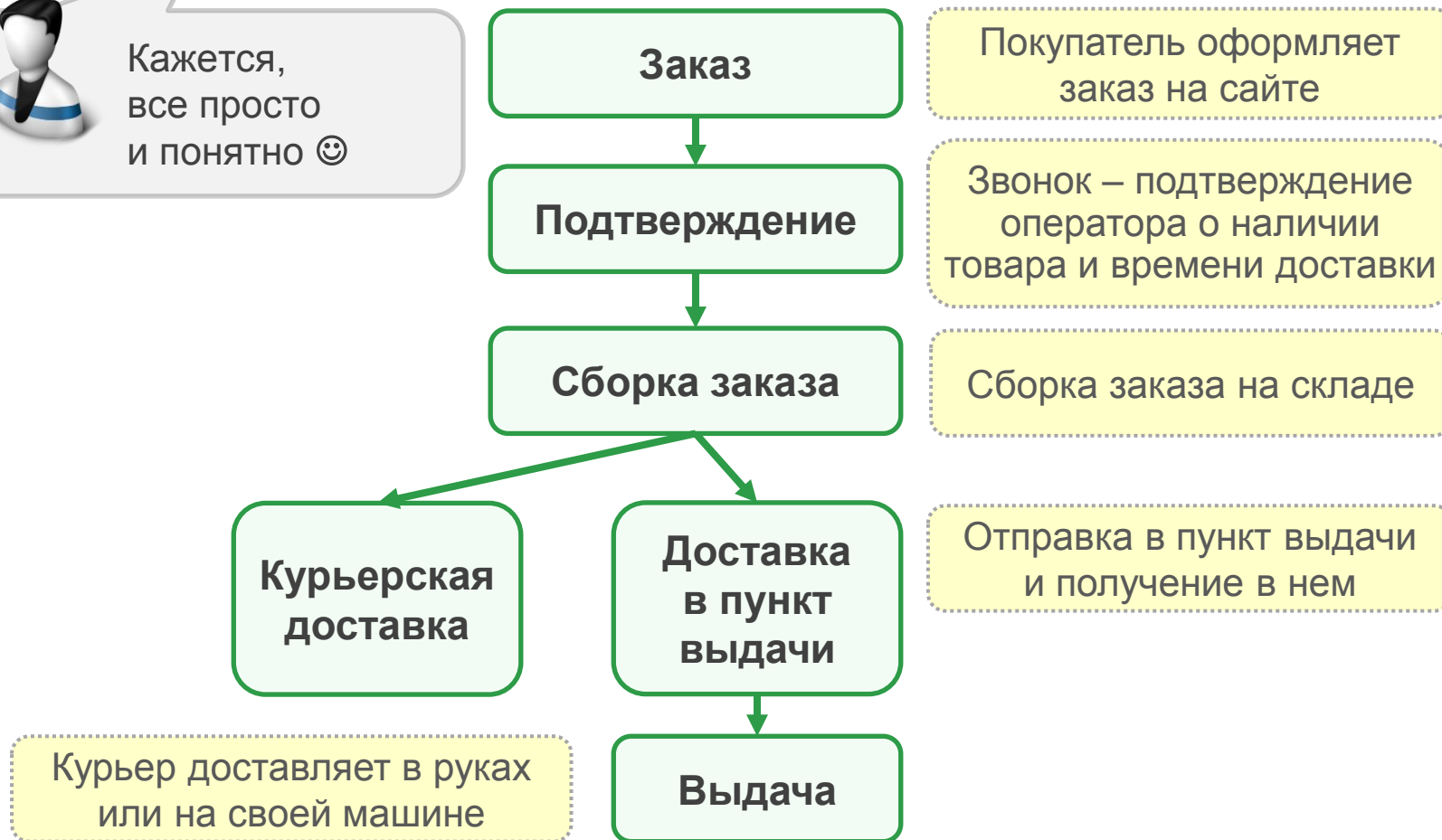
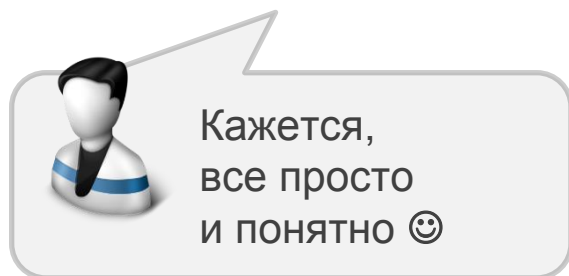
Для всех, кто считает ИТ партнером бизнеса

Пример – интернет-магазин

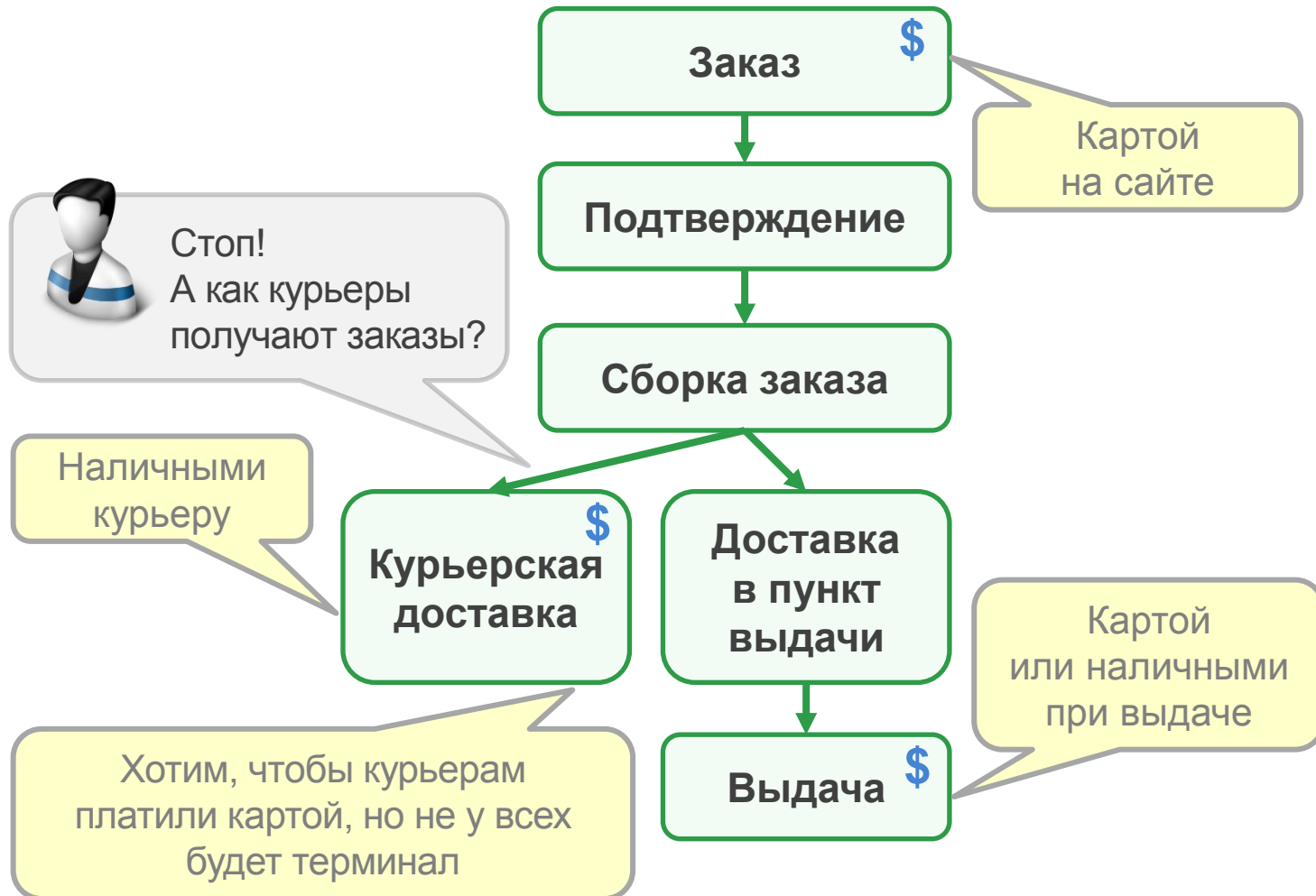
- ▶ Ситуация заказа разработки
 - Оборот вырос, сбои в доставке стали чаще – **надо исправлять ситуацию**
 - Планируется **развивать бизнес**, расширяя доступные услуги, а ограничения системы позволяют сделать не все
- ▶ Пример собран для доклада как хорошая иллюстрация его тезисов на понятном кейсе
- ▶ Рассказ будет о заказной разработке, но мы применяем те же подходы для разработки тиражируемых решений

Как усложняется представление бизнес-процессов

Процесс обработки заказа



А где оплата?





А: Как заказы назначают курьерам?

З: Их назначает **менеджер доставки**.

А: ОК, здесь вложенный процесс, а не один шаг...

А: А курьеров всегда хватает?

З: Да, мы знаем, сколько можно развезти, и операторы это учитывают при подтверждении заказа.

З: Но бывает, что курьер заболел или заказ нужно далеко везти, надо позвонить и перенести дату.

А: **Менеджер звонит** и договаривается?

З: Нет, звонят **операторы**.

З: Если дату перенесли, заказ не собирают.

А: Курьеров назначают **до сборки**?

ОК, **перерисовываем...**



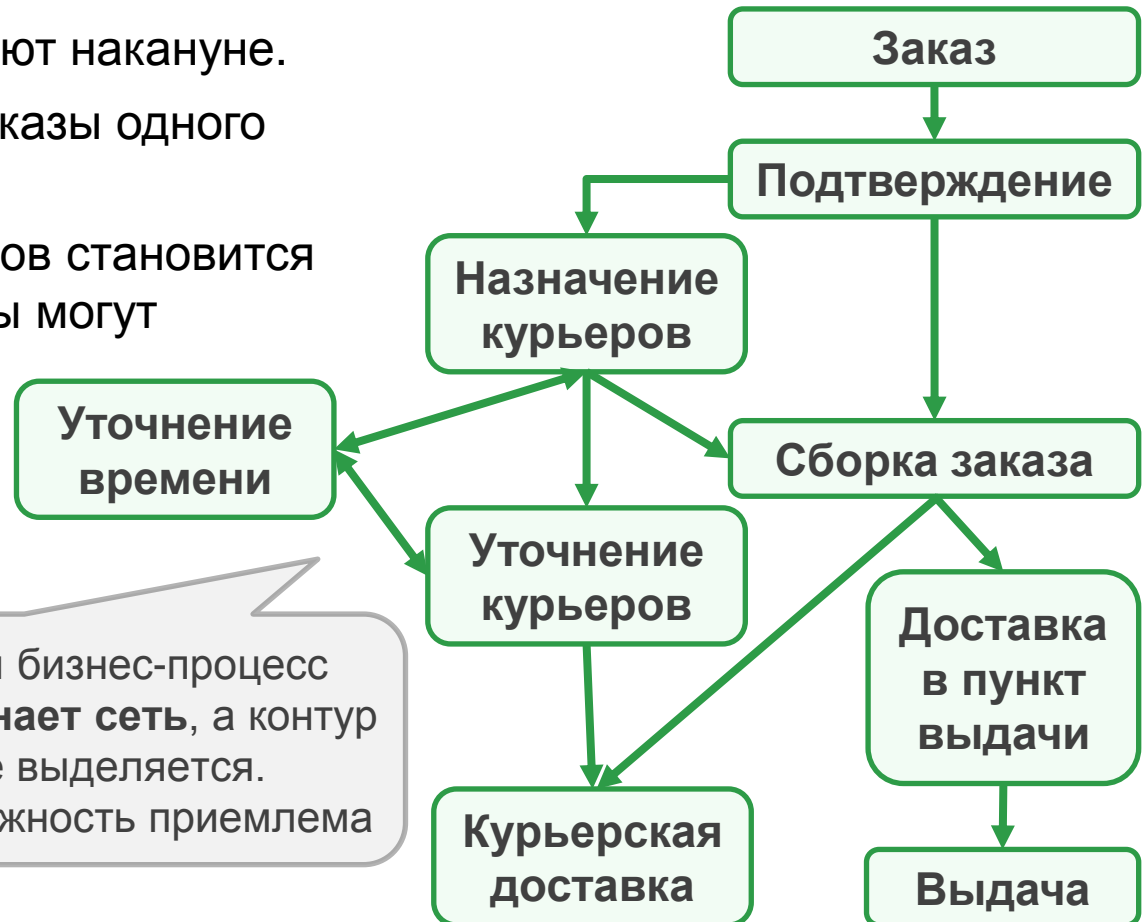


А: Заказ курьерам назначают до сборки?

З: Да, обычно это делают накануне.

З: И склад собирает заказы одного курьера вместе.

З: Но о болезни курьеров становится известно **утром**, заказы могут перераспределять...



Ранее логичный бизнес-процесс теперь **напоминает сеть**, а контур доставки уже не выделяется. Но все-таки сложность приемлема

Представление получили, переходим к задачам разработки

A: А какие бывают сбои и проблемы в доставке?

З: Ну, это не совсем сбои, это особые ситуации, их надо обрабатывать.

- ▶ Бывает, что покупателя нет на месте и мы договариваемся о новой дате
- ▶ Бывает, что мы перепутали товар и надо доставить повторно правильный
- ▶ Бывает, что везем несколько вариантов вещи и покупатель выбирает
- ▶ Бывает, что у покупателя не хватило наличных, но он может заплатить картой на сайте (сейчас это невозможно)
- ▶ Бывает, что адрес или состав заказа изменяют после того, как его собирали на складе, и надо поменять и передать другому курьеру
- ▶ Бывает, что покупатель отказывается от части заказа или нашел дефект и просит скидку, созваниваемся и согласуем – надо поменять сумму в системе

З: И надо проверить, что курьер вернул весь недоставленный товар.

Каждая ситуация – отдельная ветка процесса!

Это сейчас, а что будет развиваться?

- ▶ Сейчас операторы звонят по каждому заказу, а мы хотим перейти на СМС-уведомления в зоне, где риски невелики
- ▶ Мы хотим попробовать такую схему: сопутствующие товары предлагает не оператор, как сейчас, а курьер, представляя их покупателю на месте
- ▶ Часть товара находится на выставках в пунктах выдачи, сейчас за этим следят вручную, а надо, чтобы система об этом знала
- ▶ Заказы в другой город доставляет почта или транспортная компания, нужна специальная упаковка. Компания забирает со склада, а на почту отвозим мы. Это надо поддержать в системе, мы будем развивать такие продажи
- ▶ Бывает, что у нас просят срочную доставку и готовы платить. Сейчас это организуют вне системы, а мы хотим сделать услугу срочной доставки

Система должна позволять легко развивать процесс, пробовать новые варианты

Очевидна **проблема**

- ▶ Попытка учесть в бизнес-процессе все ветви приведет к недопустимой сложности
- ▶ Сложный софт не получится развивать в требуемом для развития бизнеса темпе

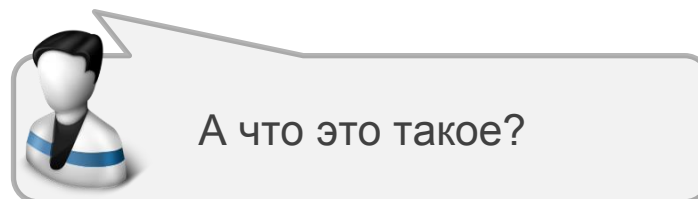


Ситуация **типична** для большого количества корпоративного софта

Что делать?

Решение

- ▶ Различаем **основной бизнес-процесс** и **исключения** из него
- ▶ Для основного процесса применяем подходы **Process Management**
- ▶ Для исключений – подходы **Adaptive Case Management**



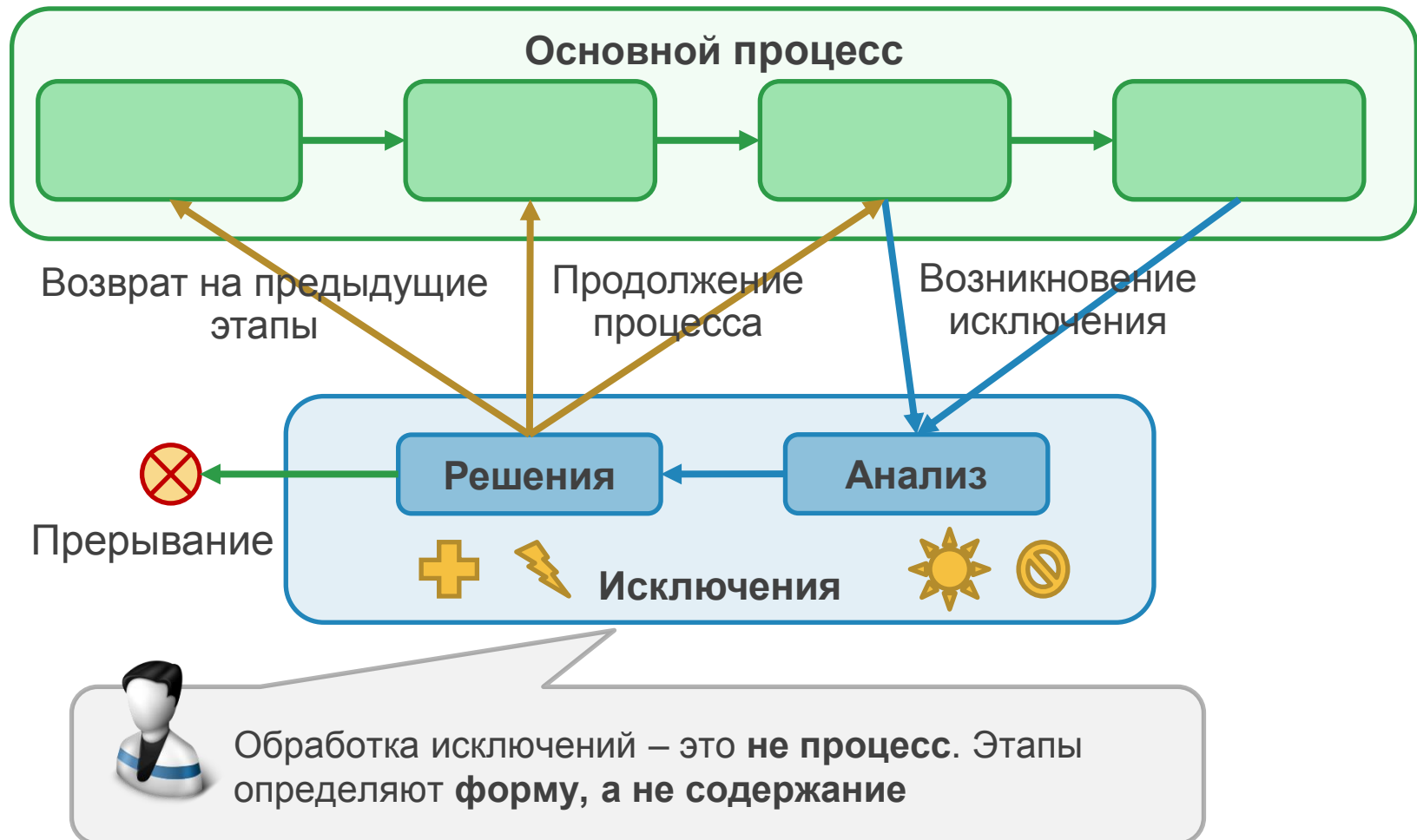
Process and Case Management

- ▶ **Process Management** вырос как способ организации однородного труда за счет разложения в конвейер простых операций
- ▶ **Adaptive Case Management** появился в таких отраслях, как медицина и юриспруденция, где обслуживание **персонализировано** и каждый случай отличается от других
- ▶ Персонализация обслуживания смещает акцент в сторону Case Management



Case Management – это
не Project Management

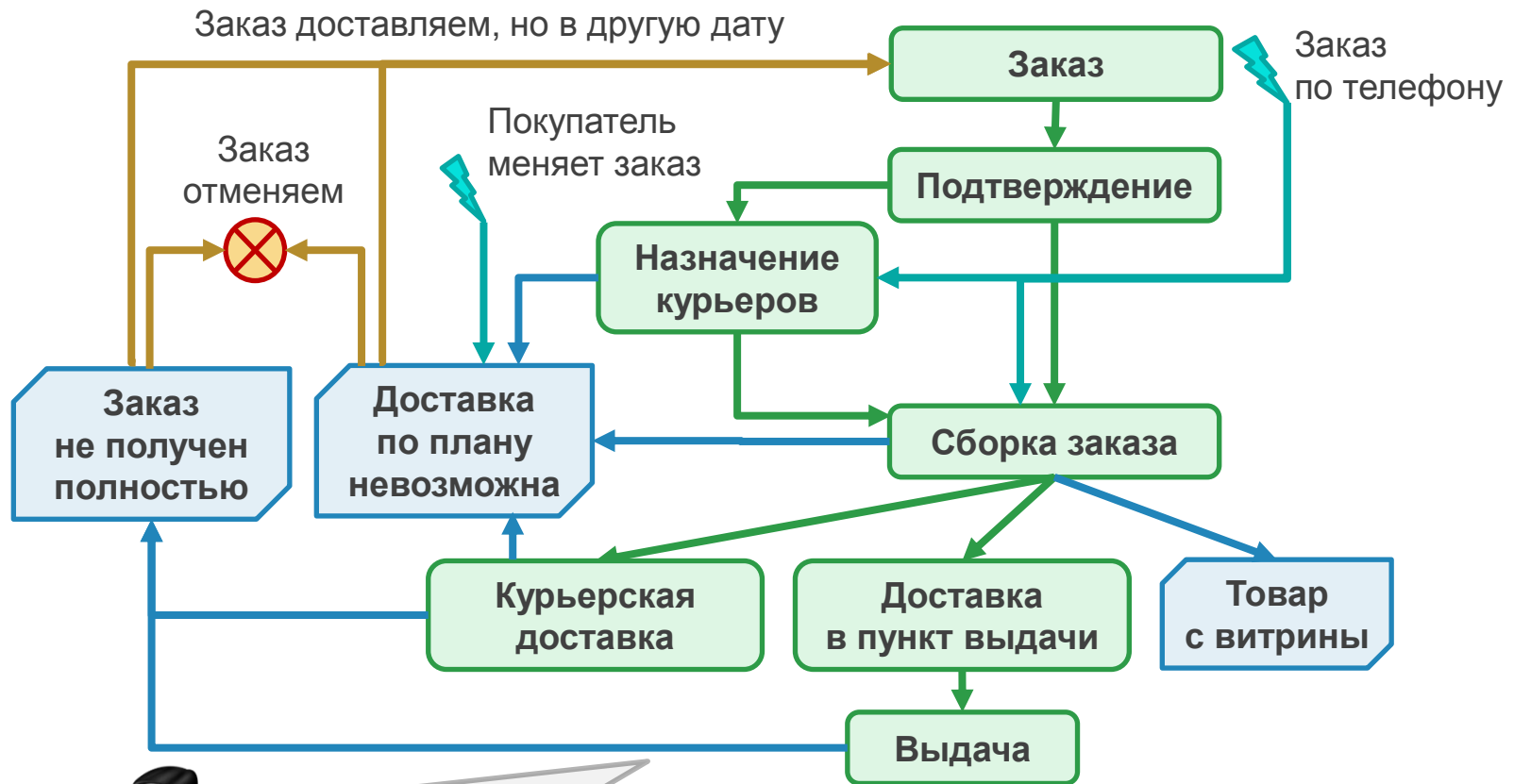
Способ совмещения подходов




Процесс и исключения

- ▶ На **любом** шаге основного процесса **ВОЗМОЖНЫ ИСКЛЮЧЕНИЯ**
- ▶ Обработка исключений ведет
 - к «**проталкиванию**» процесса
 - или **возврату** на предыдущие шаги
- ▶ При проектировании нужно определить
 - типы исключений
 - ожидаемую частоту
 - варианты результатов

Процесс и исключения в интернет-магазине




 Схема процесса приняла **обозримую** форму с небольшим количеством **типов** исключений

Проектируем и реализуем работу с исключениями

Исключения есть в любом процессе!

- ▶ Их необходимо **зафиксировать** на этапе бизнес-анализа
 - Формат Use Case, на первый взгляд, подходит: помечаем альтернативные сценарии как исключения
 - Однако обработку исключений выполняют **не те** люди, что ведут основной процесс
 - Поэтому лучше **разделять** описания, указывая в основном процессе лишь наличие исключений
- ▶ В архитектуре системы необходимо **выбрать уровень** поддержки обработки исключений и реализовать его

Уровни работы с исключениями

- ▶ **Минимальный** – позволяет «протолкнуть» исключения через систему
- ▶ **Средний** – позволяет анализировать исключения и дорабатывать бизнес-процесс
- ▶ **Глубокий** – позволяет прозрачно и качественно обрабатывать исключения, используя Adaptive Case Management



Выбор уровня зависит от построения бизнеса

Минимальный уровень обработки

➤ Обучаем BPM-движок

- **остановить** экземпляр процесса
- **создать** экземпляр процесса в **неначальном** состоянии
- **вернуть** экземпляр процесса на несколько шагов или, остановив один, сделать копию в раннем состоянии

На эти функции накладываем ограничения по доступу, но они есть для всех процессов во всех состояниях

➤ Делаем операции для отражения результатов обработки исключения

- например, форма возврата товара курьером с приемкой кладовщиком

➤ Обработка исключений идет вне системы



Без этого система **фактически не работает**, но об этом часто забывают, и разработчики выполняют действия вручную

Уровень анализа

- ▶ Дополняем систему **журналом исключений**, который фиксирует
 - Остановку или возврат по бизнес-процессу и особые операции с указанием **типа исключения**, его времени и обстоятельств
 - Результат обработки исключения и время обработки
- ▶ Реализуем аналитические отчеты по журналу для технологов бизнеса

Пример реализации: заказ получен не полностью

➔ Функции системы

- Отметить неполученные позиции или весь заказ, **для анализа** указать в журнале причину, по которой их не получили
 - При необходимости повторной доставки сделать **новый** заказ тому же адресату, наследующий оплату, **связать новый заказ с предыдущим**, указав причину в журнале
 - Вернуть оплату для оплаченных на сайте заказов
 - Принять недоставленный товар на складе на хранение до следующей доставки или как свободный для других заказов
- ➔ Причины указываются из стандартного списка с возможностью дополнения и описания ситуации в комментарии



Синим выделены расширения уровня анализа

Уровень Case Management

- ▶ Исключения – отдельный объект системы
- ▶ Бизнес-логику реализует пара: BPM-движок и task-трекер
 - При исключении для экземпляра процесса в BPM-движке создается тикет в трекере, процесс приостанавливается
 - По тикету идет взаимодействие по обработке исключения
 - Результат обработки отображается в экземпляре процесса BPM-движка
- ▶ Реализуем интерфейсы работы с исключениями
 - Современный вариант интерфейса – чат с уведомлениями
- ▶ Анализ исключений ведем по тикетам

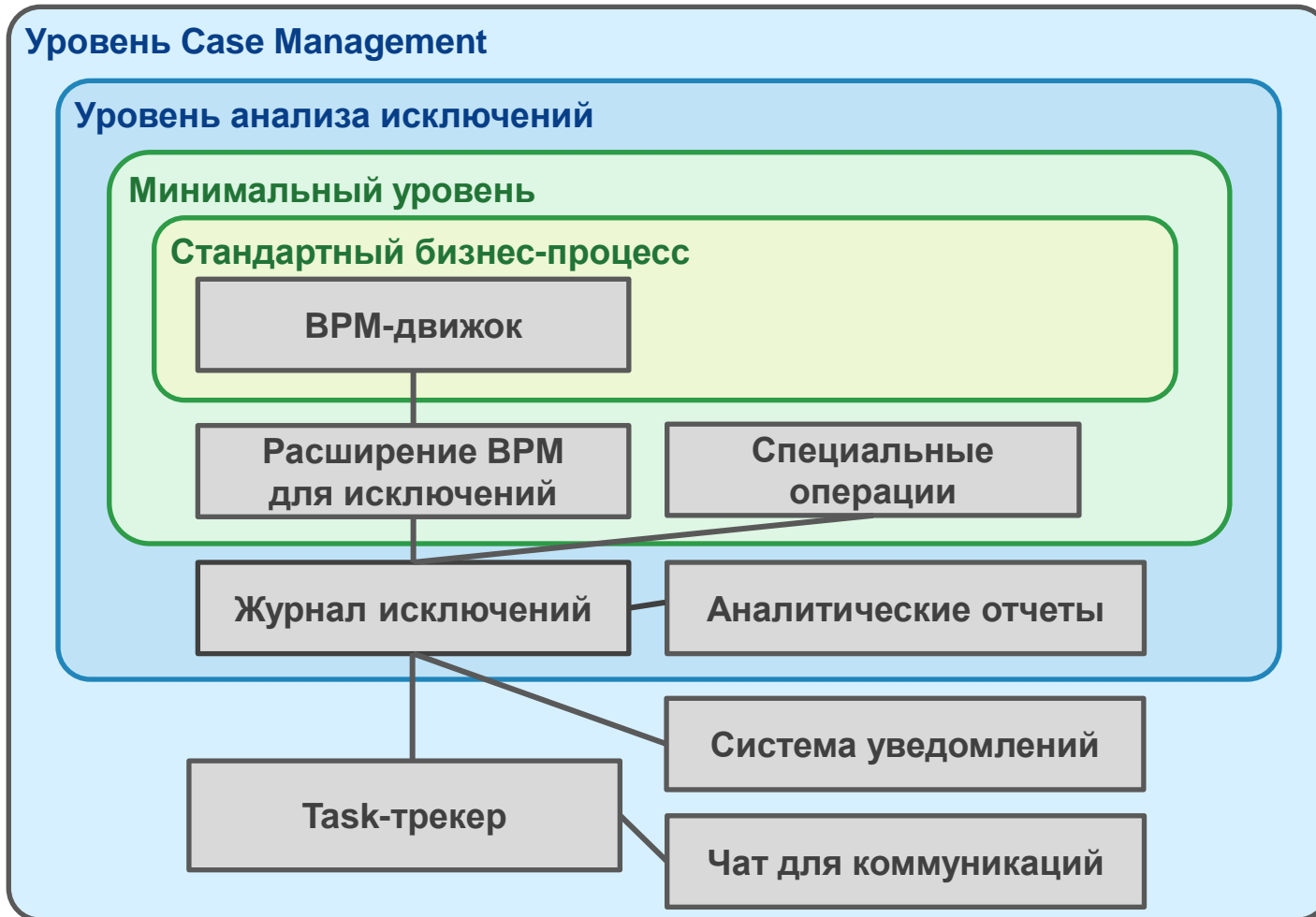
Case «Заказ получен не полностью»

➤ Возможный сценарий работы

- Оператор по звонку курьера заводит отдельный тикет к заказу, где указаны ситуация и набор ожидаемых действий (связаться с клиентом, принять товар на склад, повторно доставить, вернуть оплату)
- Сотрудники следят за появляющимися тикетами и обрабатывают их, набор ожидаемых действий изменяется
- Обработка носит асинхронный характер
- В некоторый момент инцидент считается решенным и продолжается основной процесс обработки заказа

➤ Система поддерживает сценарий в виде чата вокруг тикета, включая уведомления

Архитектурная схема



Выбор уровня работы с исключениями

Позиция бизнеса для каждого уровня

Адаптивный и развивающийся бизнес

Анализ исключений для достройки процесса

Минимальный уровень – рабочий бизнес-процесс

Стандартный бизнес-процесс



Исключения – зло, это всегда ошибка сотрудников, пусть сами и мучаются



В процессе бывают неожиданности, с ними надо уметь справиться



Качество работы с исключениями важно для обслуживания клиентов, а их анализ позволит совершенствоваться



Надо пробовать новое и развиваться, предоставлять новые услуги и персонализировать существующие

Контроль в масштабном бизнесе

- ▶ Эффективный масштабный бизнес основан на хорошей организации процессов
 - При этом любые нештатные ситуации жестко решаются не в пользу клиента – персонал не умеет иначе
 - Сейчас это ведет к конкурентному проигрышу
- ▶ Фиксируя исключения, можно организовать работу с ними **отдельно** от основного процесса
 - Привлечь персонал другого уровня
 - Держать под контролем обработку исключений
 - Анализировать их и совершенствовать бизнес

Развитие бизнеса через механизм Adaptive Case Management



Пример – сопутствующие товары

- ▶ У ряда товаров есть сопутствующие (например, чехлы и защита для смартфонов)
 - Обычно операторы предлагают их заказать
 - Альтернатива – курьер привозит и предлагает на месте
- ▶ Вариант с курьером можно проверить
 - В заказ включать скрытые позиции, выдавая курьеру
 - Если не получилось – оформлять возврат
 - Эксперимент потребует малых доработок
 - Можно провести ограниченно для конкретных товаров
- ▶ При успехе – продумывать сценарий поэтапного массового внедрения

Пример – срочная доставка

- ▶ Придумываем, как организовать срочную доставку отдельных заказов без существенных напряжений
- ▶ По запросу такой услуги заводим тикет, который обрабатывается сотрудниками придуманным способом
 - Ограничиваем число доставок доступной мощностью
- ▶ Оцениваем процесс и стоимость услуги: убрать ее, оставить дорогой опцией или попробовать сделать массовой

Подводя итоги

- Современные ИТ – партнер бизнеса
- Современный бизнес требует **персонализации** сервиса и непрерывного **развития** услуг
- **Качество бизнеса** определяется и **процессами**, и обработкой **исключений** в них
- **Process & Case Management совместно** дают баланс простоты и гибкости, обеспечивая качество бизнеса и его развитие



Спасибо! Вопросы?

Максим Цепков mtsepkov.org