



МИРЭА – Российский технологический университет

МЕТОДОЛОГИЯ МОДЕЛИРОВАНИЯ ВРМН

Курс лекций

Лектор: Ивахник Дмитрий Евгеньевич, к.э.н., доцент





Курс лекций

Содержание:

1. Методология функционального моделирования SADT
2. Бизнес-процесс как объект исследования
3. Методология моделирования BPMN
4. Методология моделирования BPMN. Элементы нотации
5. Методология моделирования ARIS
6. Методология моделирования ARIS. Построение eEPC
7. Подходы к моделированию бизнес-процессов
8. Применение подходов к моделированию бизнес-процессов



5 семестр: 8 лекций (16 часов)

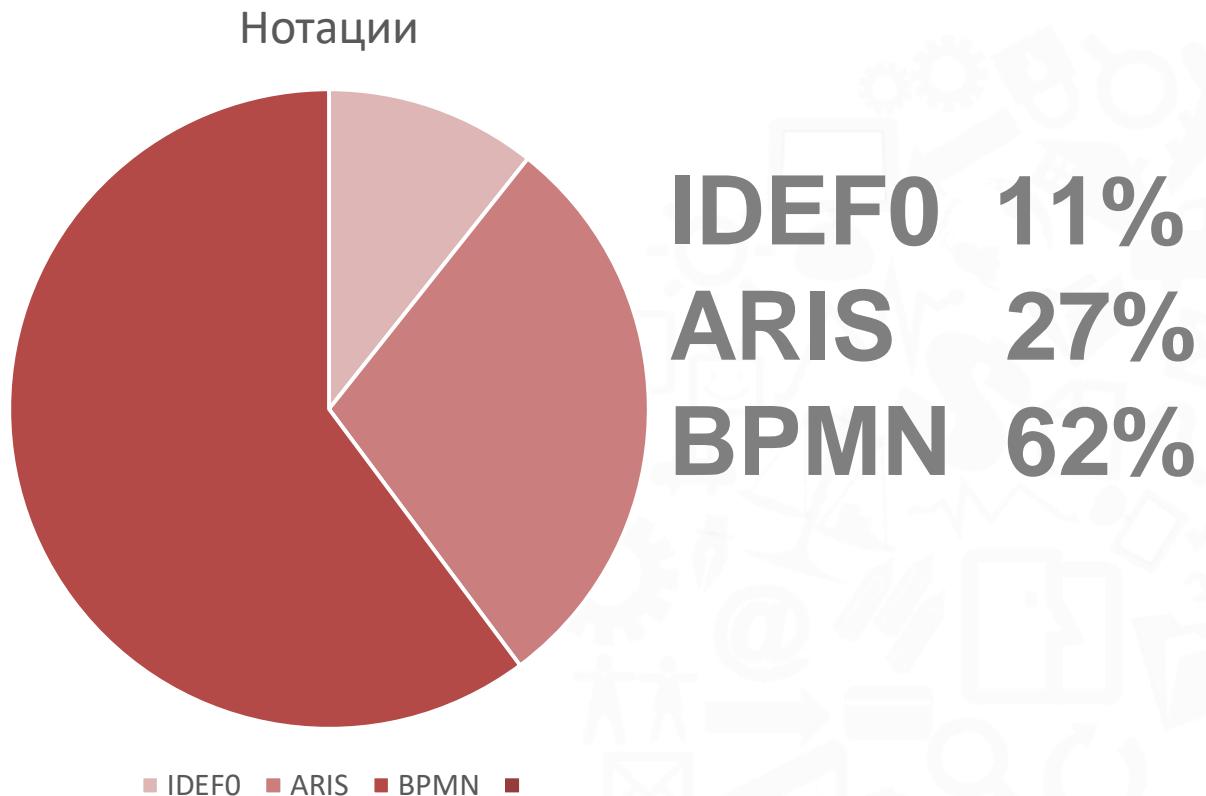


План лекции:

1. Элементы нотации BPMN 2.0: типы элемента «Задача», маркеры действий и типы событий
2. Диаграммы в нотации BPMN 2.0



Российский рынок систем моделирования бизнес-процессов





МЕТОДОЛОГИЯ МОДЕЛИРОВАНИЯ ВРМН

Процессы для которых может применяться методология ВРМ (Business Process Management):

- 1. Процессы, которые реализуются работниками различных структурных подразделений**
- 2. Сквозные и кросс-функциональные процессы**
- 3. Процессы, в целях реализации которых используются прикладные программы, работающие на различных платформах**



BPM

Business Process Management

- концепция управления, объединяющая цели организации с ожиданиями и потребностями клиентов, путем обеспечения эффективного выполнения бизнес-процессов

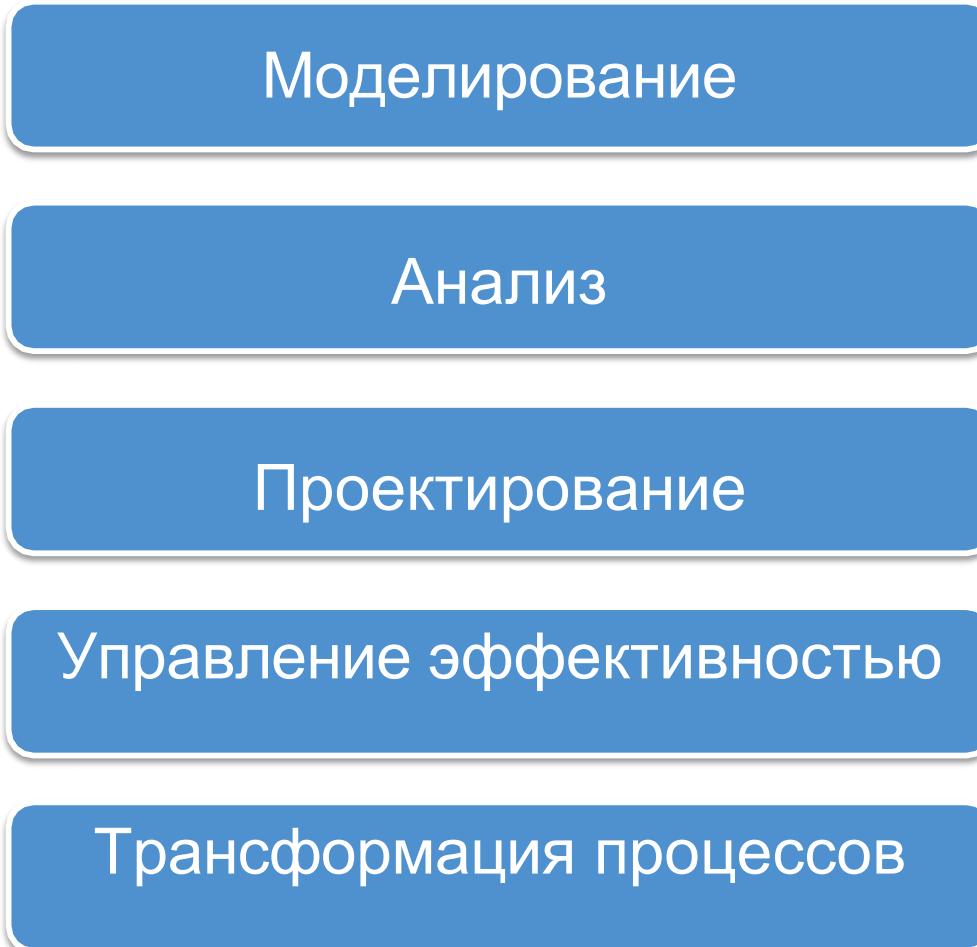
Сфера применения: моделирование бизнес-процессов с помощью специального набора элементов, понятного большинству бизнес-пользователей (бизнес-аналитики, менеджеры, управляющие процессами, исполнители процессов), а также разработчикам ИС.



Разработчик – международная организация BPMI
(Business Process Management Initiative)
(после реорганизации носит название **Object Management Group**)



Основные работы цикла управления процессами в BPMN





БРМ как подход



Совершенствование
операционной
деятельности

Реорганизация
(крупномасштабные
изменения)

Процессно-
ориентированный подход

Взгляд на бизнес через
модели процессов и в
привязке к явным бизнес-
правилам и операционно-
техническим параметрам



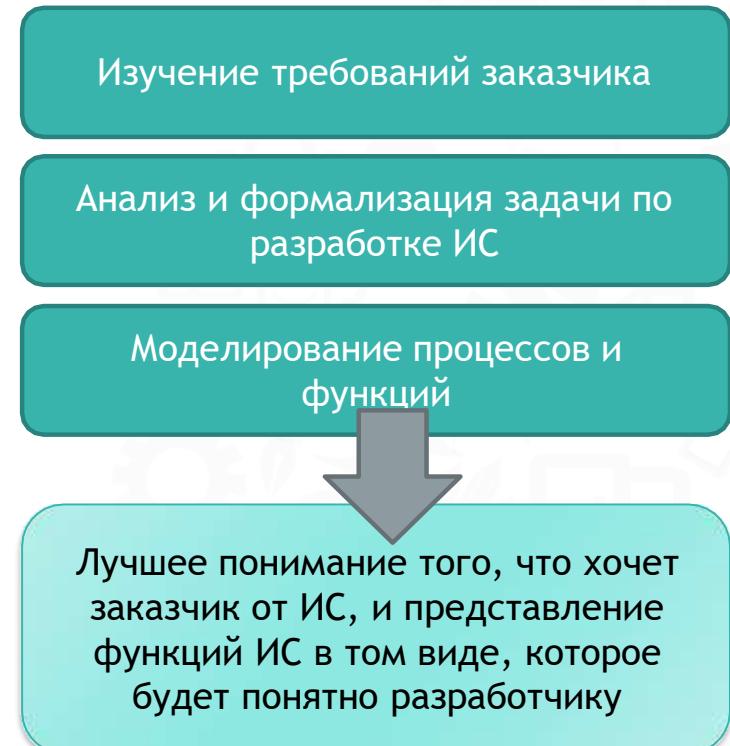
BPM как подход

Преимущества и недостатки BPMN

Плюсы	Минусы
Перевод модели бизнес-процесса в соответствующий программный код	Не предусмотрены нотации для описания организационной структуры, информационной модели, дерева целей и др.
Перенос и чтение диаграмм бизнес-процессов между различными графическими редакторами и инструментальными средствами бизнес-моделирования, которые поддерживают версию BPMN 2.0	Содержит более 100 различных символов



Место моделирования процессов в разработке ИС





Сначала анализ - затем моделирование

Опрос специалистов



Системы показателей



Графики, расписания



Документы и формы



Наблюдение за работой



Изучение требований заказчика

Анализ и формализация задачи по разработке ИС

Моделирование процессов и функций

Лучшее понимание того, что хочет заказчик от ИС, и представление функций ИС в том виде, которое будет понятно разработчику



BPMN

Исследование бизнес-процессов организации сопровождается изучением документов организации, которые регламентируют осуществление бизнес-процесса, проведением анкетирования и интервью, построением модели «as is».

Анализ бизнес-процесса позволяет создать модели «to be», позволяющие из них выбрать наилучшую альтернативу по изменению бизнес-процесса

Выбранная альтернативная модель «to be» дает возможность реализовать создание информационной системы с уже измененными бизнес-процессами

«AS WAS»

«AS IS»

«TO BE»



BPMN

Business Process Model and Notation

Цель создания BPMN:

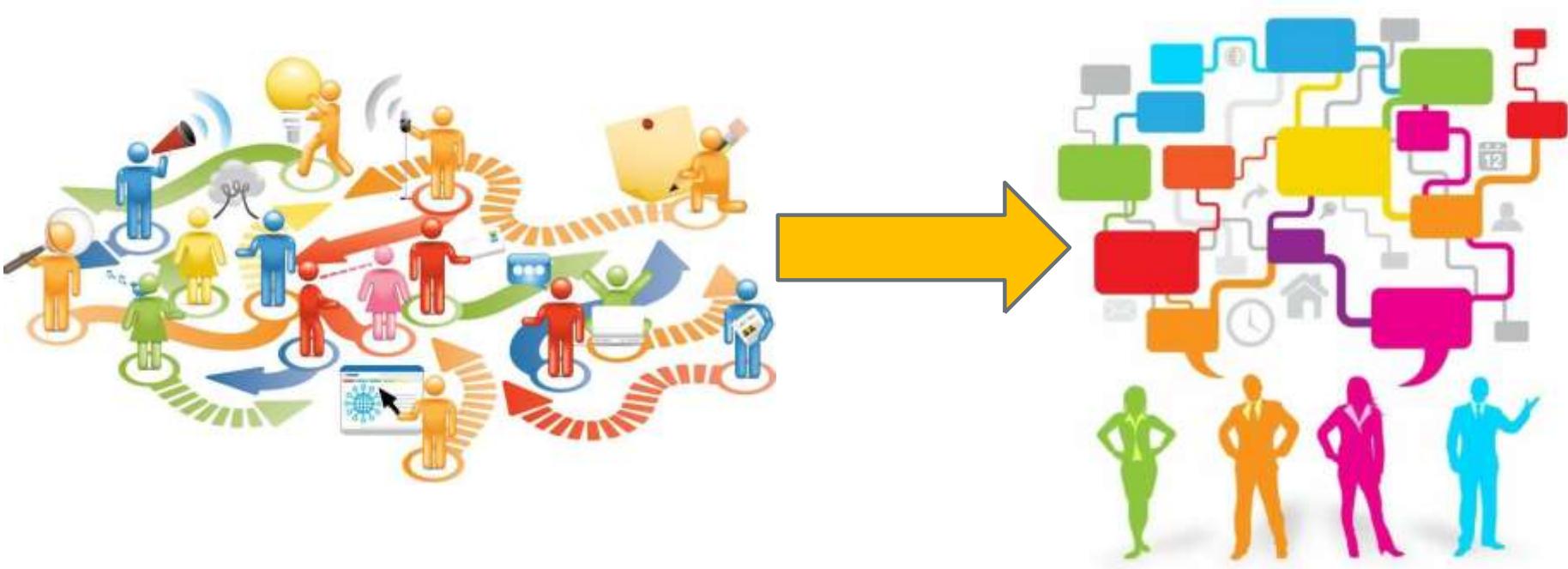
- устранение расхождения между моделями бизнес-процессов и их реализацией.
- языки XML, разработанные для исполнения бизнес-процессов, теперь могут быть визуализированы в графической нотации, понятной обычным бизнес-пользователям
- бизнес-пользователи, внедренцы, заказчики и поставщики получают простые средства доступа к информации о процессе.





BPMN 2.0

1. Бизнес-процесс описывается «как есть» и сразу подвергается автоматизации, то есть переносу в рабочую систему управления бизнес-процессами

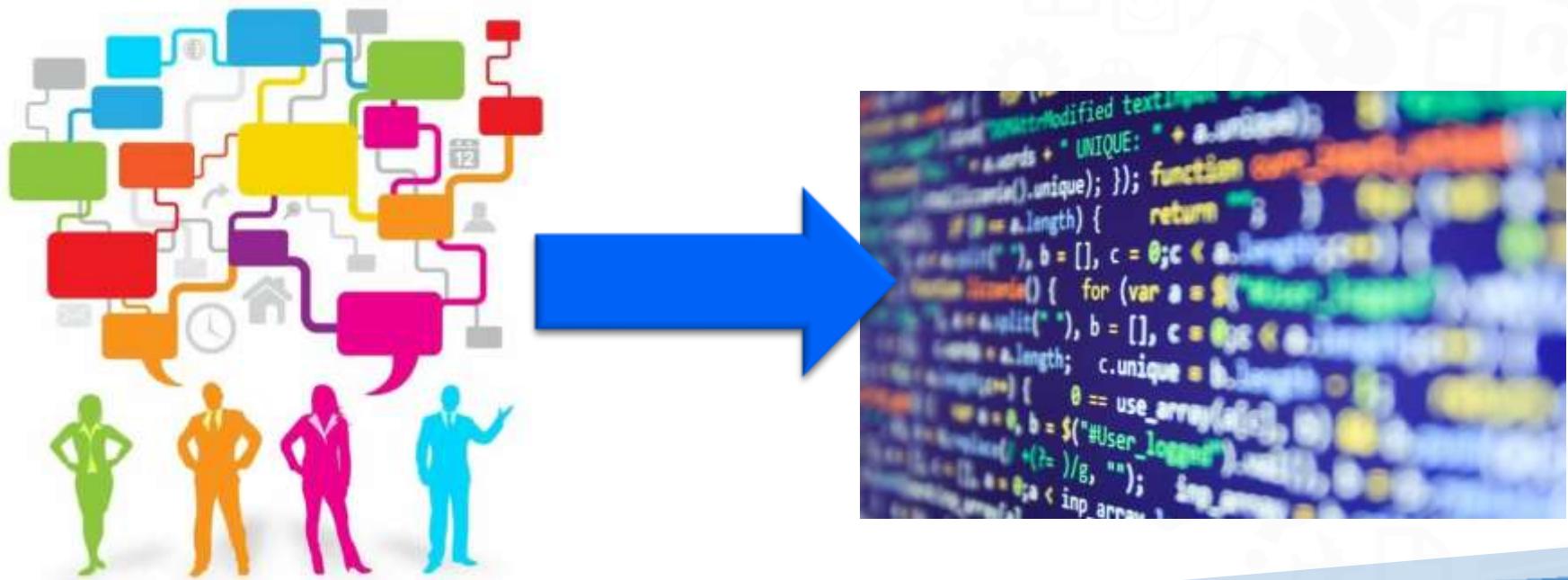




BPMN 2.0

- Для управления процессами должна использоваться система, в которой **графическая модель процесса автоматически преобразуется в программный код бизнес-приложения**.

Такая система позволяет сразу после создания BPMN-модели начать выполнение смоделированного процесса в системе.





ВРМН 2.0

3. Если есть потребность в улучшении или изменении хода реализации процесса, то (по заданию аналитика или бизнес-пользователя) это можно реализовать в короткие сроки без существенных затрат, поскольку подход позволяет перенастроить приложение в соответствии с требованиями пользователя довольно легко.



BPMN 2.0 - Метамодель и нотация бизнес-процессов

<http://bpmb.de/poster>

Действия



Задача - единица работы. Если задача помечена символом , то задача является подпроцессом и может быть детализирована.



Транзакция - набор логически связанных действий. Для транзакции может быть определен протокол выполнения.



Событийный подпроцесс помещается внутри другого процесса. Он начинает выполняться, если нинческим его начальное событие. Событийный подпроцесс может прервать родительский подпроцесс или выполняться параллельно с ним.



Вызывающее действие является точкой входа для глобально определенного подпроцесса, который повторно используется в данном процессе.

Маркеры действий
Маркер отражает поведение действия во время выполнения:

Маркер подпроцесса

Маркер цикла

Маркер параллельных множественных экземпляров (МЭ)

Маркер последовательных множественных экземпляров (МЭ)

Маркер ad hoc

Маркер компенсации

Типы задач
Тип определяет природу действия, которое будет выполнено:

Задача отправки сообщения

Задача получения сообщения

Пользовательская задача

Неавтоматизированная задача

Задача-бизнес-правило

Задача-сервис

Задача-сценарий

Поток управления

Поток по умолчанию

Условный поток

определяет порядок выполнения действий.

определяет ветвь процесса, выполняемую, когда все условия ветвления не выполнены.

связан с условием, определяющим будет ли выполнен данный поток.

Логические операторы

Оператор исключающего ИЛИ, управляемый данными

При ветвлении направляет поток лишь по одной из исходящих ветвей. При синхронизации потоков оператор ожидает завершения одной входящей ветви и активизирует исходящий поток управления.

Оператор исключающего ИЛИ, событийный

Предоставляет только события обработки или замыкающих обработочных сообщений. Поток управления направляется по той ветви, где событие произошло раньше.

Оператор И

При разделении на параллельные потоки все ветви активируются одновременно. При синхронизации параллельных ветвей оператор ждет завершения всех входящих ветвей и затем активизирует исходящий поток.

Оператор ИЛИ

При ветвлении активируется одна или более ветвей. При слиянии всех выполняющихся ветвей должны быть завершены.

Сложный оператор

Моделирует сложные условия ветвления и слияния.

Диалоги



Информационное взаимодействие задает цепочку логически связанных обменов сообщениями. Если информационное взаимодействие помечено символом , то оно может быть детализировано.

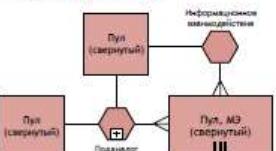


Связь соединяет информационное взаимодействие с одним участником.



Разветвляющаяся связь соединяет информационное взаимодействие с несколькими участниками.

Схема диалога



Хореографии



Участник А

Задача хореографии

Участник В

III

Задача хореографии

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Маркер множественных участников показывает, что несколько участников одного типа участвуют в хореографии.

Подпроцесс хореографии содержит детализацию хореографии, включющую несколько взаимодействий.

Участник А

Подпроцесс хореографии

Участник В

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А

Задача хореографии

Участник В

Задача хореографии

Участник С

Участник А



Нотация BPMN 2.0

Программное средство для построения моделей - **Bizagi Modeler**

<https://www.bizagi.com/>



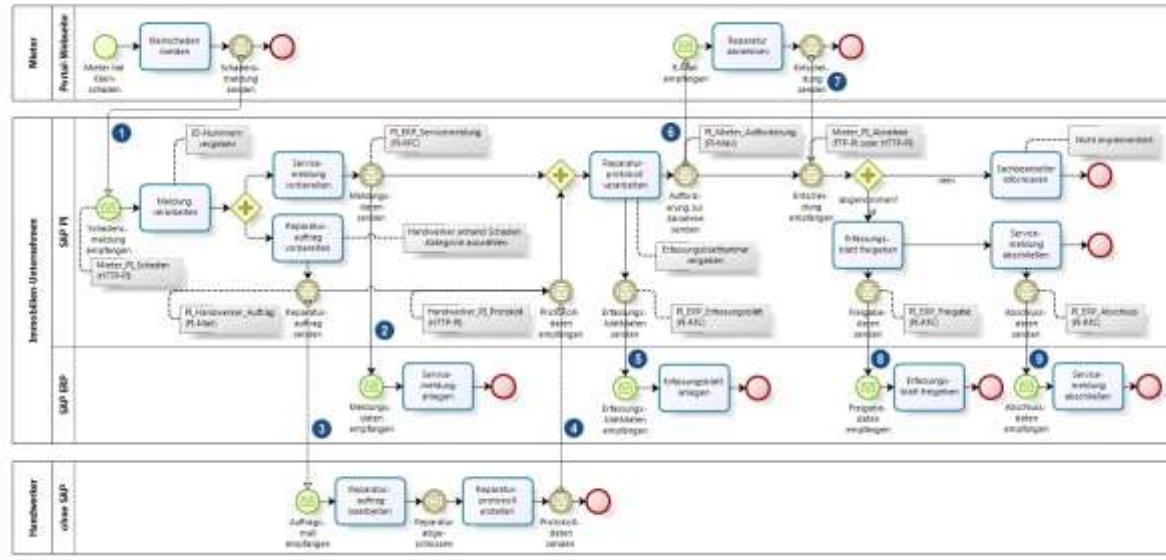
Goodbye programming,
hello modeling

Bizagi BPM software puts simplicity first. Our innovative drag and drop interface optimizes the best in clean, modern design.



Основные категории элементов ВРМН

1. Роли
 2. Объекты потока управления
 3. Соединяющие объекты
 4. Артефакты

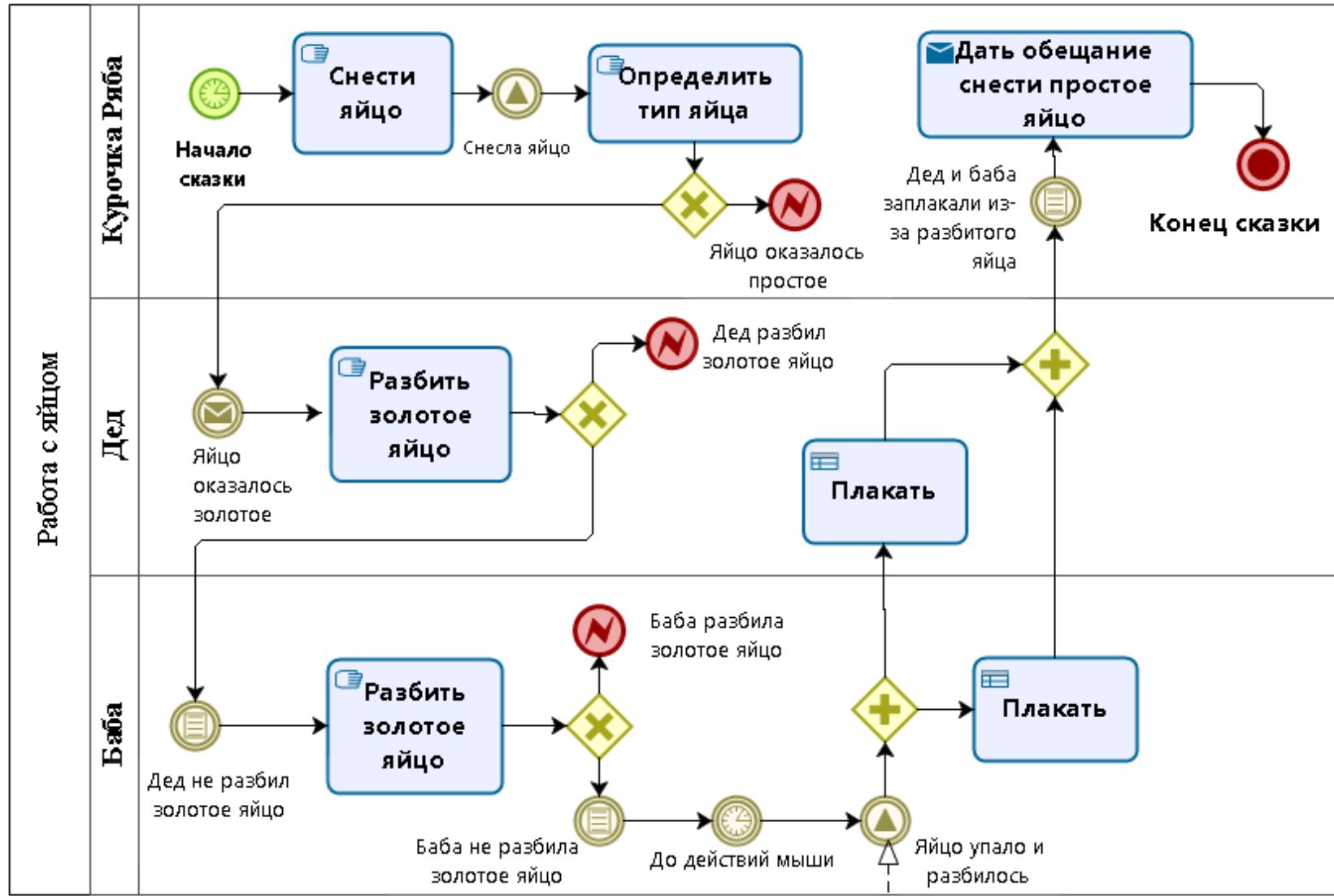




Пример моделирования процесса на основании текста русской народной сказки



*«Ох уж эти сказки,
ох уж эти
сказочники...»*





Пример моделирования процесса по сказке «Курочка Ряба» в нотации BPMN 2.0

Жили-были дед да баба. И была у них Курочка Ряба.

Снесла курочка яичко, да не простое - золотое.

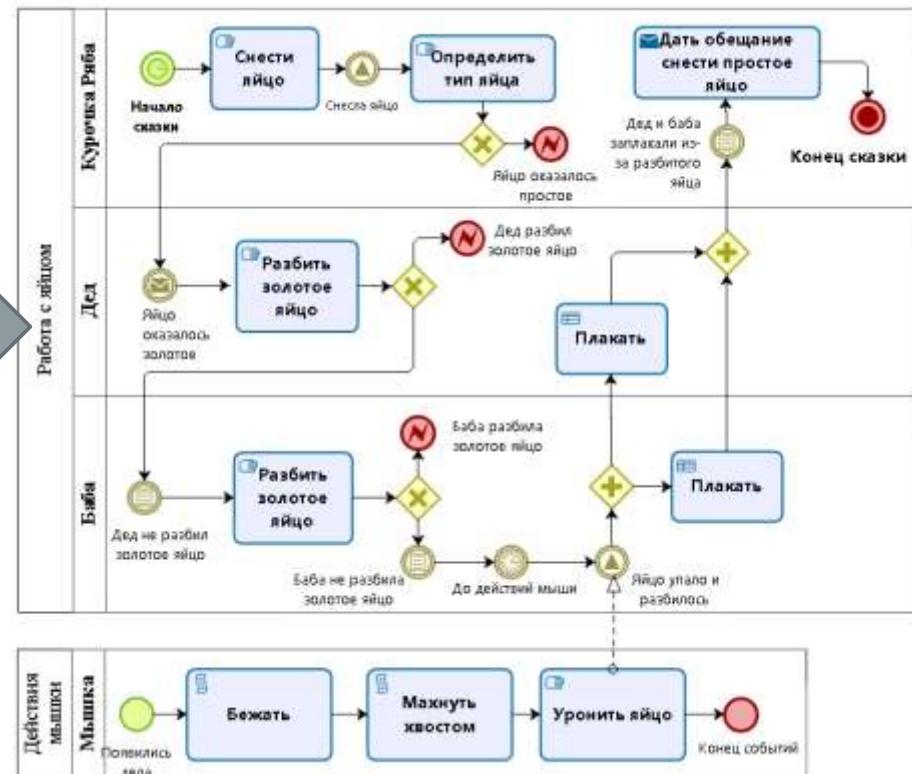
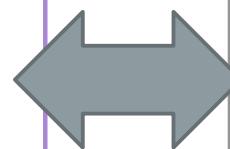
Дед бил - не разбил.

Баба била - не разбила.

А мышка бежала, хвостиком махнула, яичко упало и разбрзилось.

Плачет дед, плачет баба и говорит им Курочка Ряба:

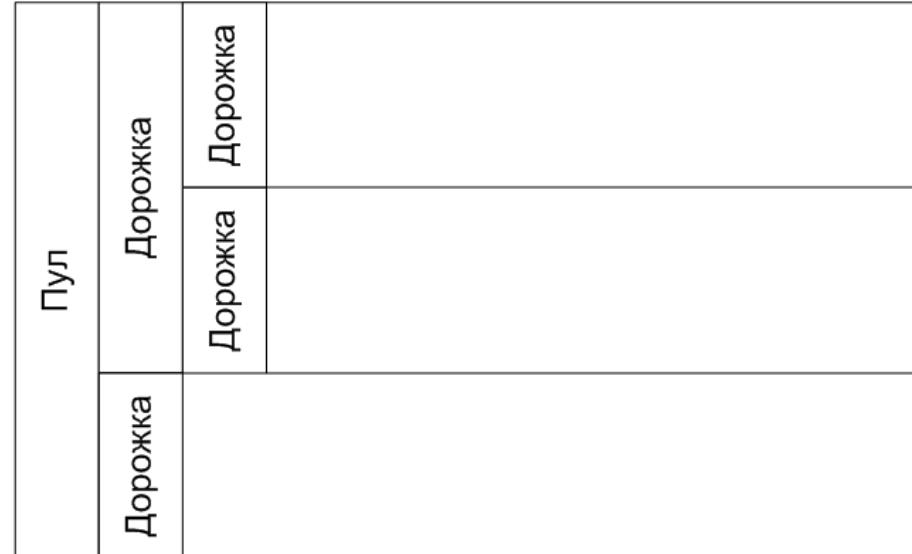
- Не плачь, дед, не плачь, баба:
снесу вам новое яичко не золотое, а простое!





РОЛИ в ВРМН: пулы и дорожки

- **Пул** предназначен для отображения потока рассматриваемого процесса. Содержимое пула - это и есть тот процесс, диаграмма которого рассматривается. На диаграмме развернутый пул может быть только один.
- **Дорожка** предназначена для отображения организационных единиц (должности, подразделения, роли, внешнего субъекта) - исполнителей задач и подпроцессов процесса ВРМН.



Свёрнутый пул



Пул		
	Дорожка 1	Дорожка 2

Процесс 1		
	Исполнитель 1	Исполнитель 2

Пулы и дорожки



Процессы и исполнители



Работа с яйцом			
Баба	Дед	Курочка Ряба	



Действия мышками	Работа с яйцом		
Мышка	Баба	Дед	Курочка Ряба

Действия мышками	
Мышка	

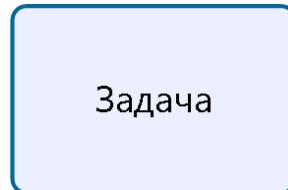


Объекты потока управления

- **события (events)**



- **действия (activities)**

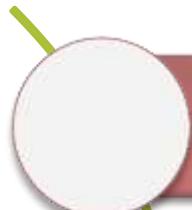
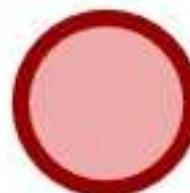


- **логические операторы (gateways)**





Объекты потока управления. События (events)



Изображаются окружностью

Означают какое-либо происшествие в мире

Отвечают на вопрос «Что произошло?»
или «Какой момент времени наступил?»

События инициируют действия или
являются их результатами.



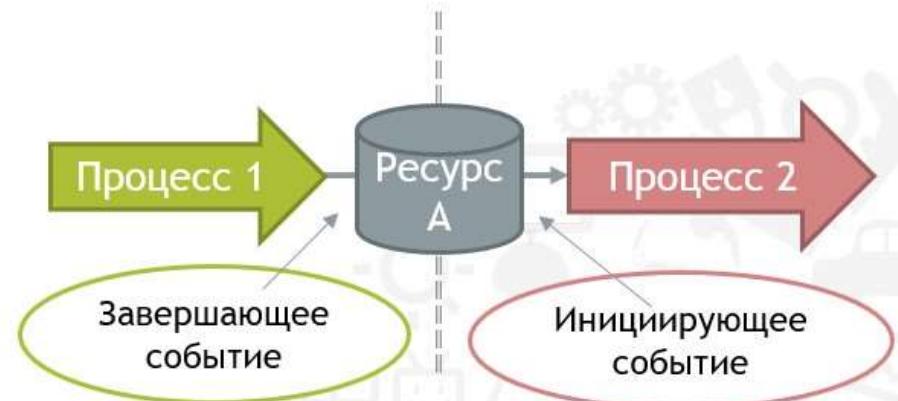
Границы процесса

Событие — наступление определенной ситуации (времени, перехода ответственности за ресурсы).

Примеры событий на границах бизнес-процессов:

- ✓ Поступил заказ
- ✓ Утвержден план
- ✓ Изготовлен продукт
- ✓ Выбран поставщик
- ✓ Получен запрос от клиента
- ✓ Каждое 7-ое число месяца
- ✓ 9:00 ежедневно
- ✓ Конец отчетного периода

И т.п.





Объекты потока управления. События (events)

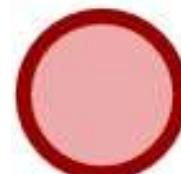
Согласно расположению в процессе события могут быть классифицированы на:



начальные (start)



промежуточные (intermediate)



завершающие (end)





Жили-были дед да баба. И была у них Курочка Ряба.

Снесла курочка яичко, да не простое - золотое.

Дед бил - не разбил.

Баба била - не разбила.

А мышка бежала, хвостиком махнула, яичко упало и разбилось.

Плачет дед, плачет баба и говорит им Курочка Ряба:

*- Не плачь, дед, не плачь, баба: **снесу** вам новое яичко не золотое, а простое!*



Работа с яйцом	Начало сказки	Снесла яйцо	Яйцо оказалось простое	Конец Сказки
	Баба	Дед	Яйцо оказалось золотое	
			Дед разбил золотое яйцо	
			Не удалось разбить золотое яйцо	
			Не удалось разбить золотое яйцо	

Действия мышки	Появились дела
Мышка	



Объекты потока управления. Действия (activities) / Задачи / Задания (tasks)

— это единица работы,
элементарное действие в
процессе.

Задача



Работа с яйцом	Дед Курочка Ряба	Снести яйцо
		Разбить золотое яйцо
		Плакать

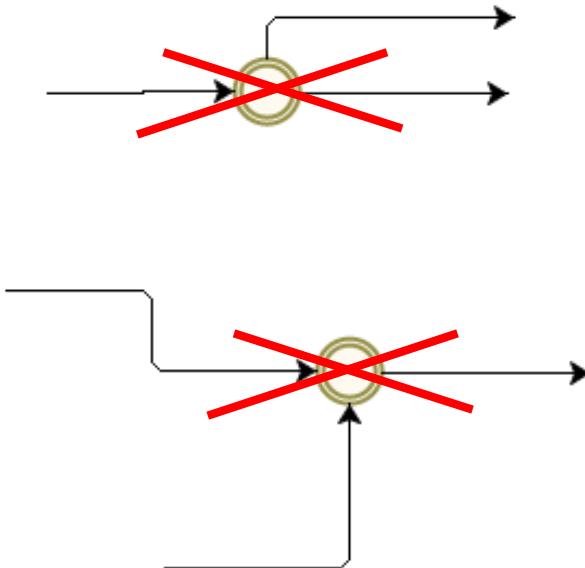
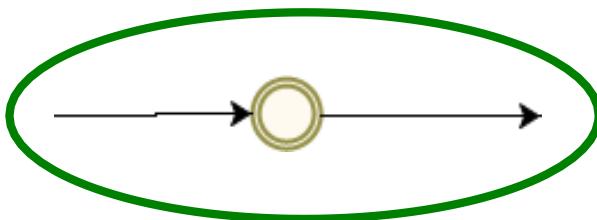
Действия мышки	Мышка	Бежать	Махнуть хвостом	Уронить яйцо
-------------------	-------	--------	-----------------	--------------





Рекомендация соединения действий и событий для моделей процессов при проектировании ИС:

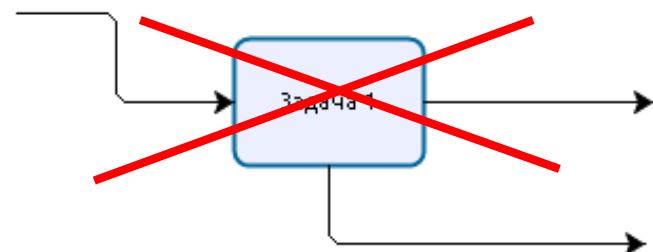
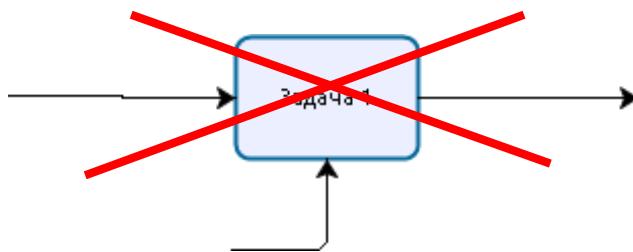
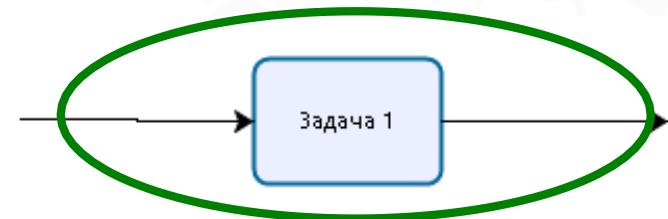
В одно **событие** может «входить»
только одна стрелка и «выходить» из
события одна стрелка





Рекомендация соединения действий и событий для моделей процессов при проектировании ИС:

В одно **действие (задачу)** может «входить» только одна стрелка и «выходить» из события одна стрелка

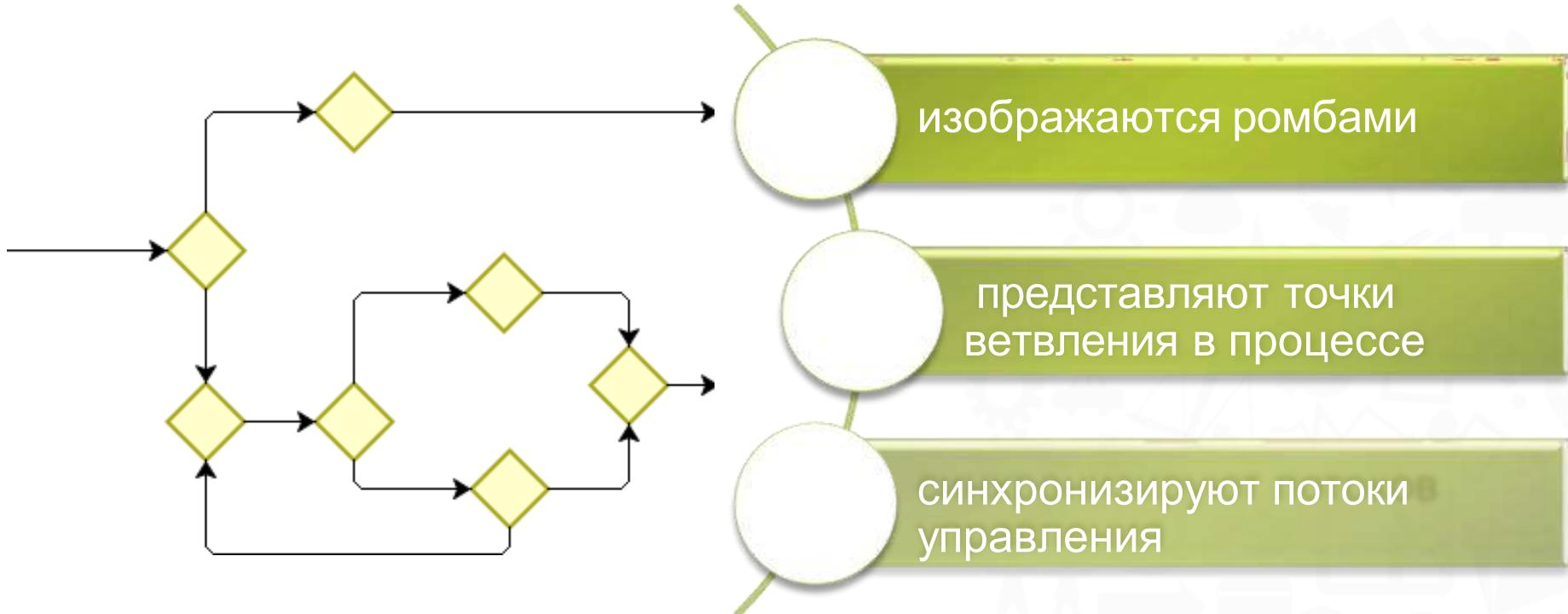




			<pre>graph LR; Start((Начало сказки)) --> Collect[Снести яйцо]; Collect -- Снесла яйцо --> Identify[Определить тип яйца];</pre>
Работа с яйцом			
Баба	Дед	Курочка Ряба	

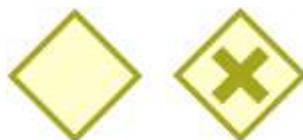


Объекты потока управления. Логические операторы





Объекты потока управления. Логические операторы



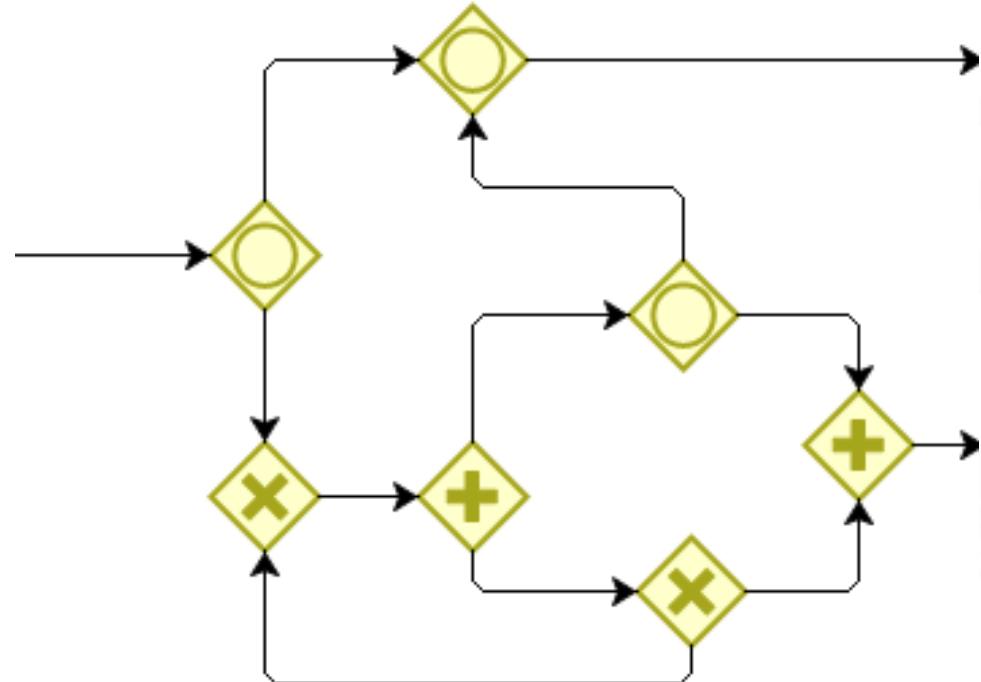
ИЛИ
исключающее



ИЛИ
неисключающее
(включающее)

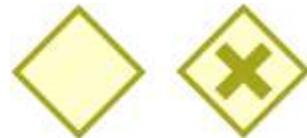


И

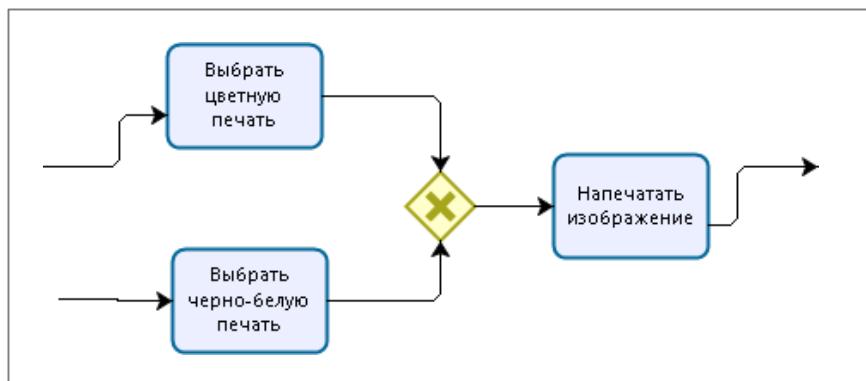




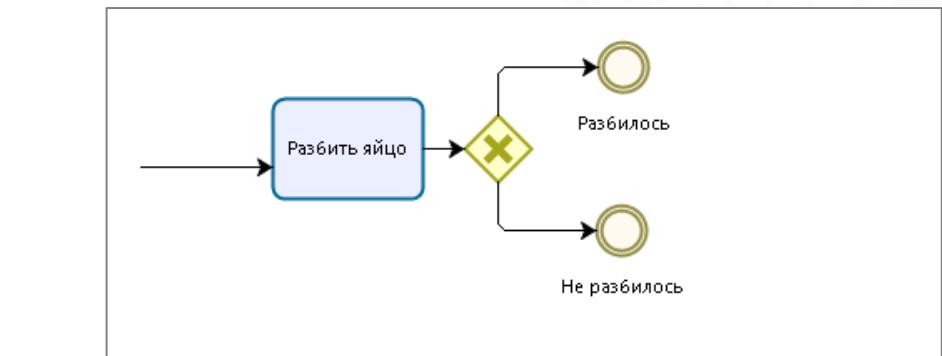
Логический оператор «ИЛИ исключающее»



Либо одно, либо другое, но не оба вместе (или для большего количества вариантов)



Слияние



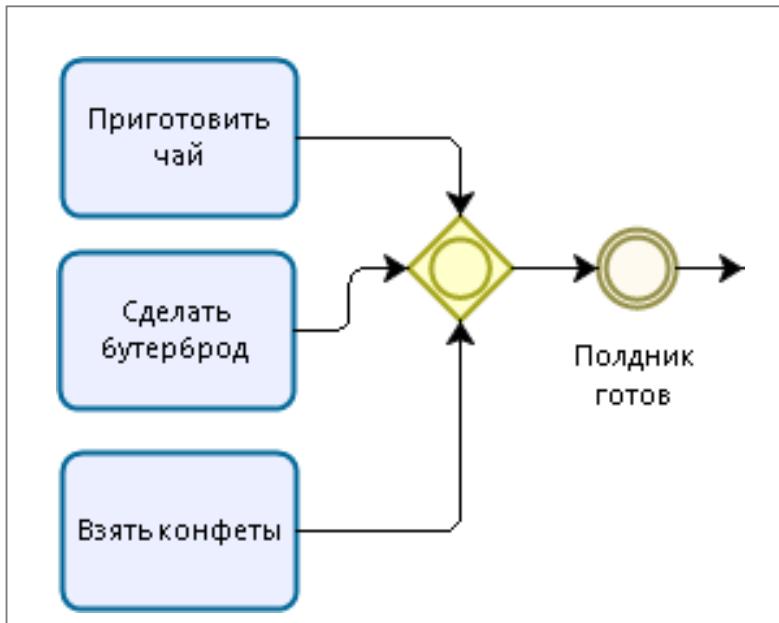
Разветвление



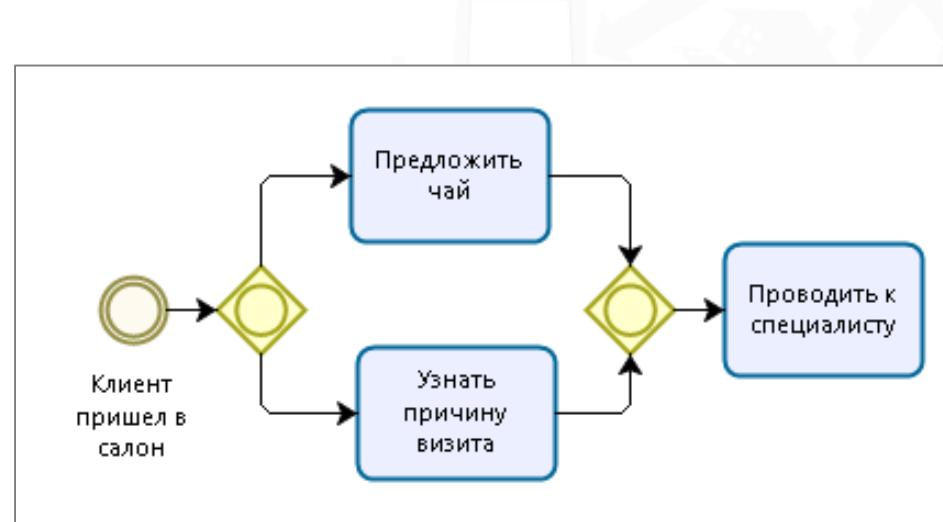
Логический оператор «ИЛИ включающее» (и/или)

Либо одно, либо другое, либо оба вместе.

Проверочная фраза «Хотя бы одна из веток должна быть выполнена»



Слияние



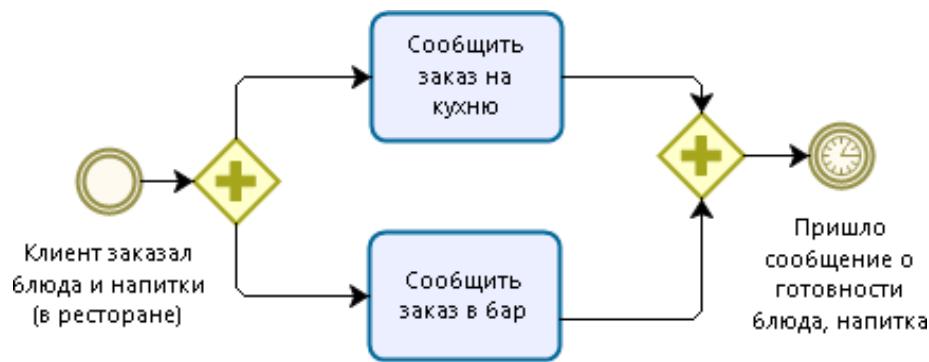
Разветвление

Слияние



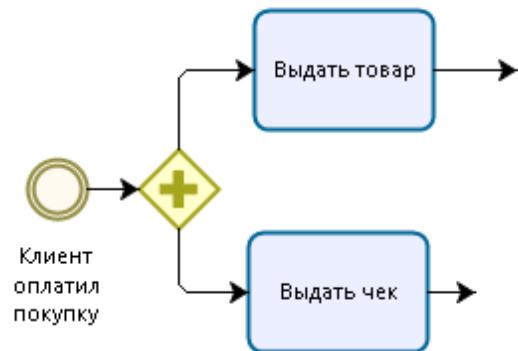
Логический оператор «И»

И то, и другое – обязательно все вместе.



Разветвление

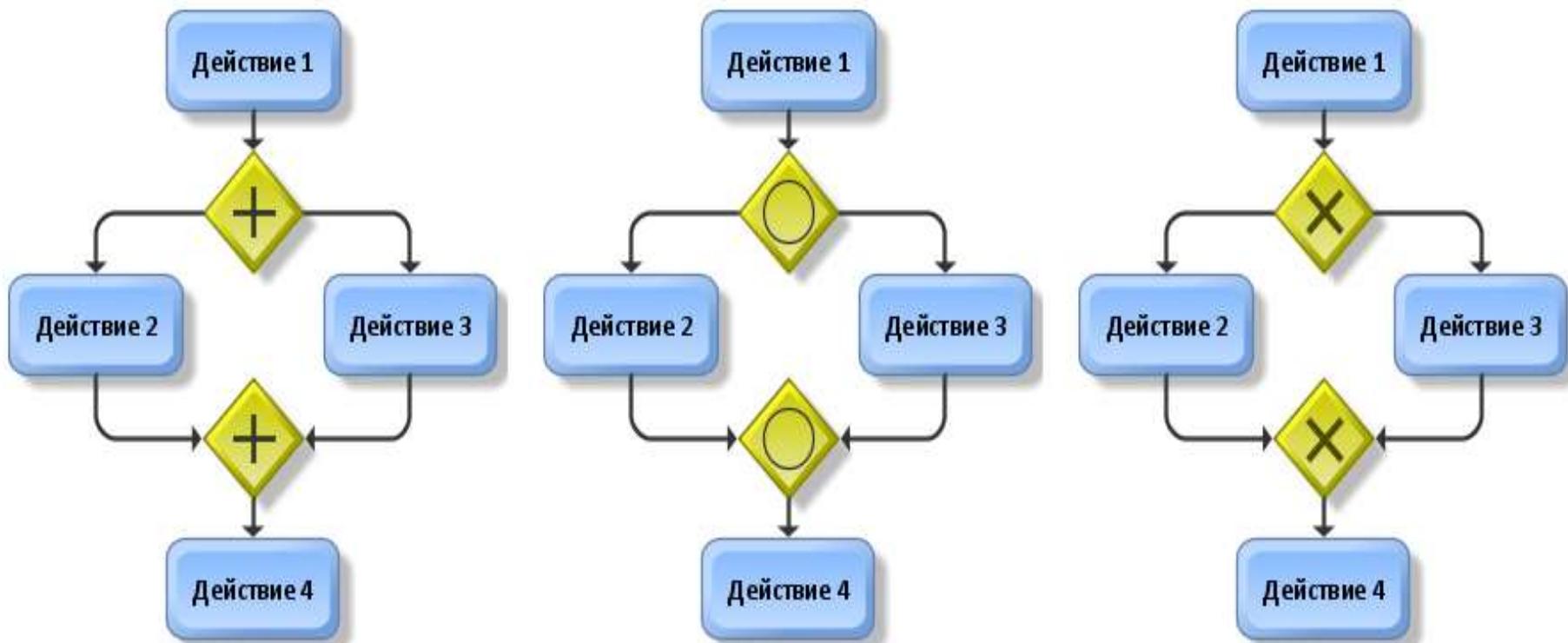
Слияние



Разветвление

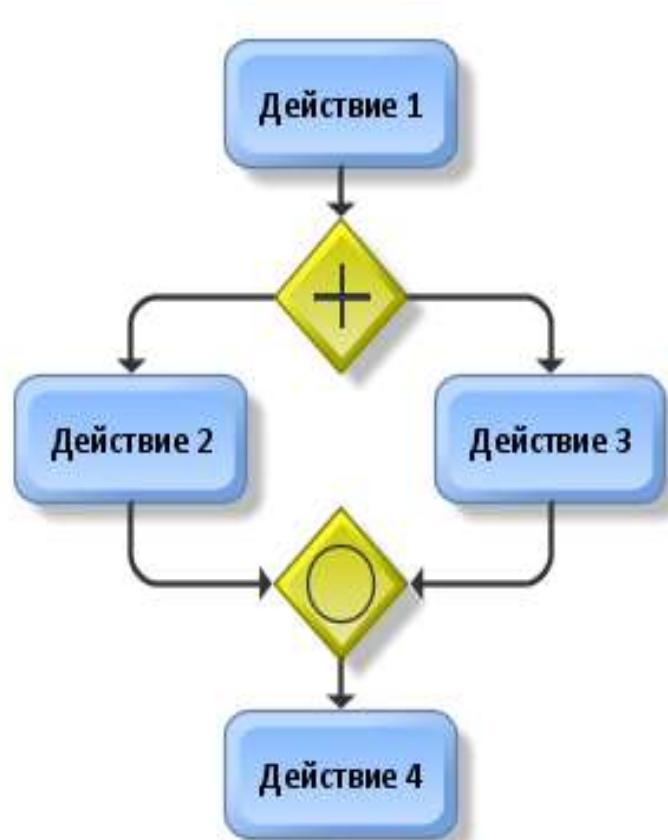


Правила применения шлюзов (допустимые ситуации)



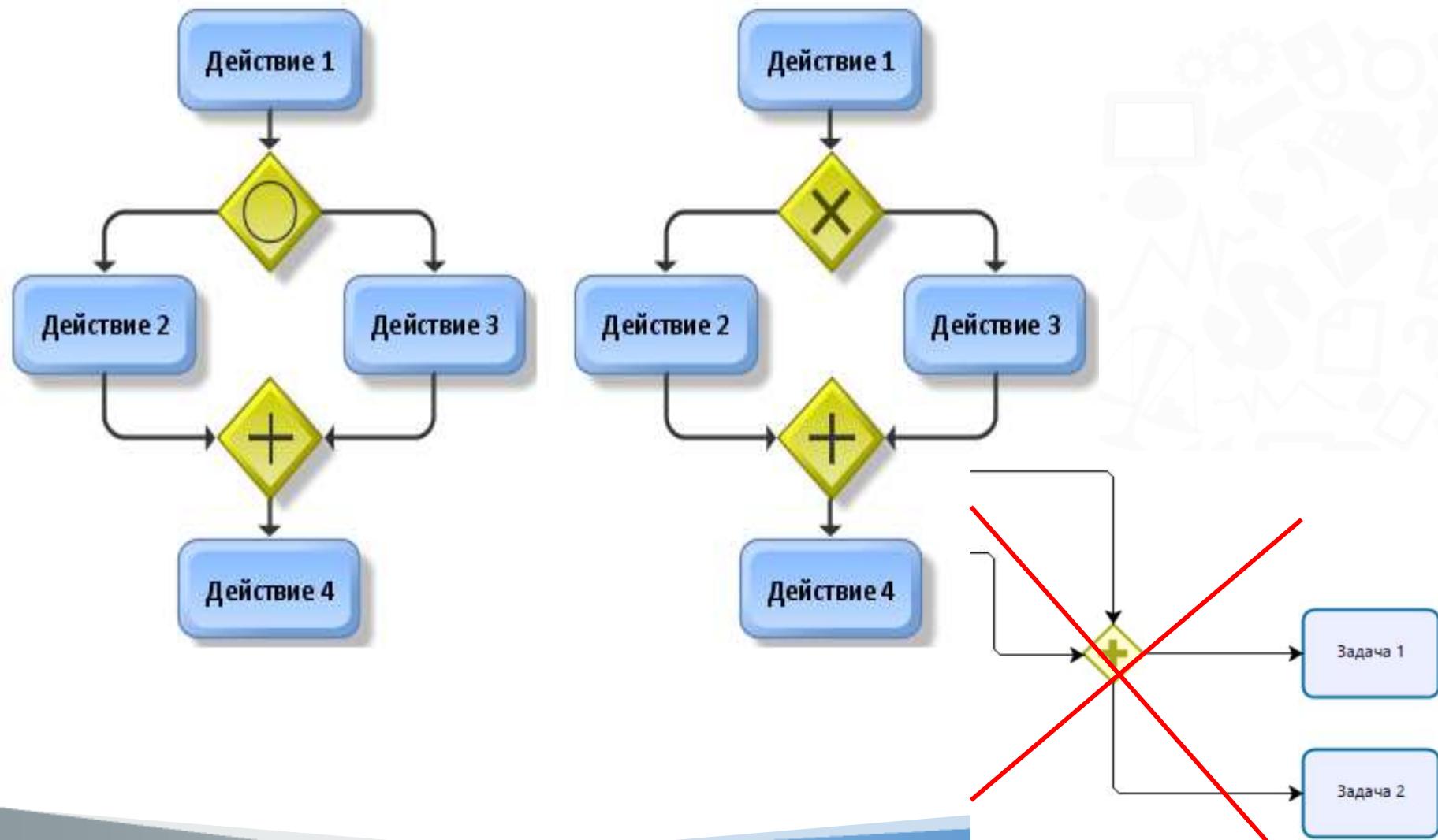


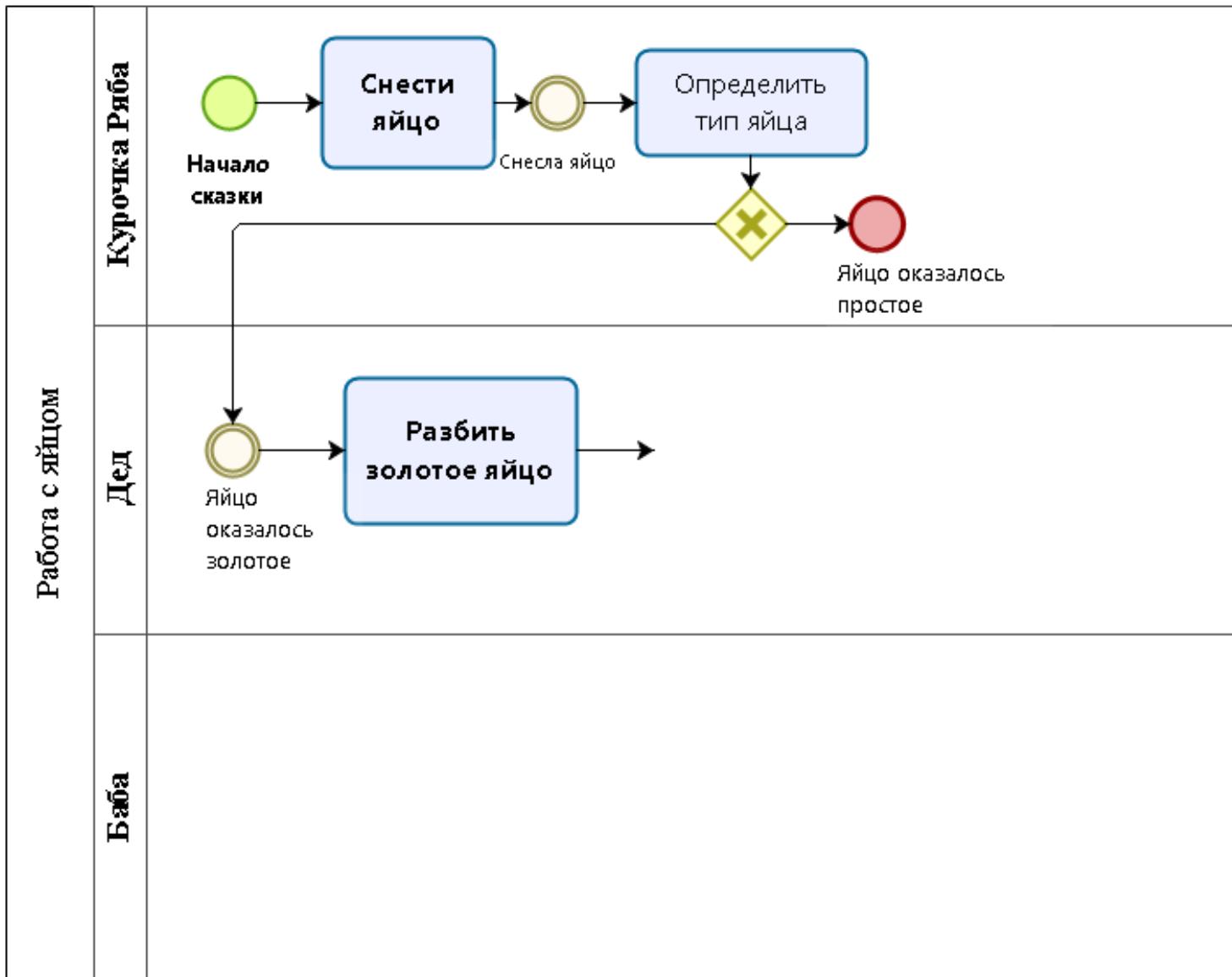
Правила применения шлюзов (допустимые ситуации)

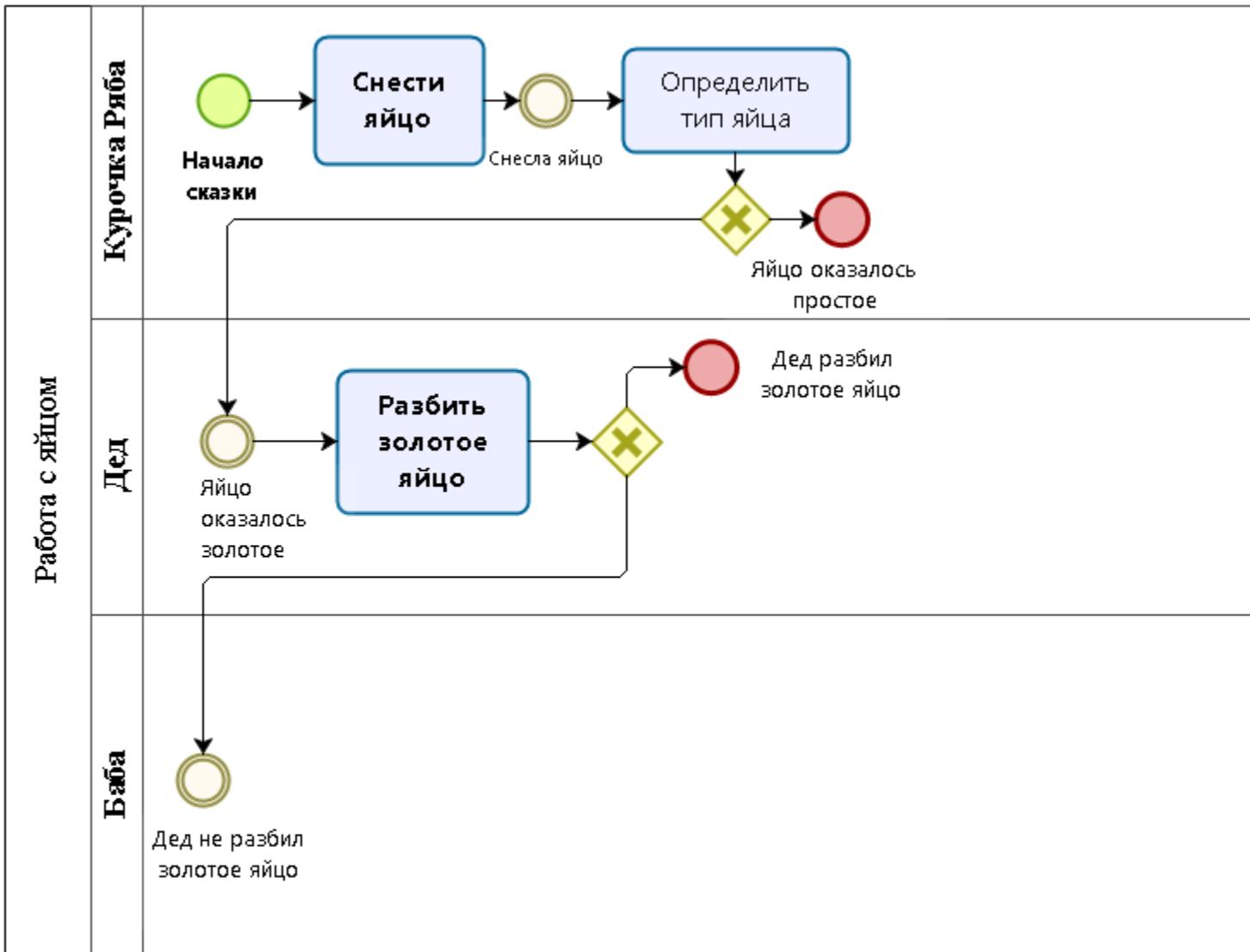


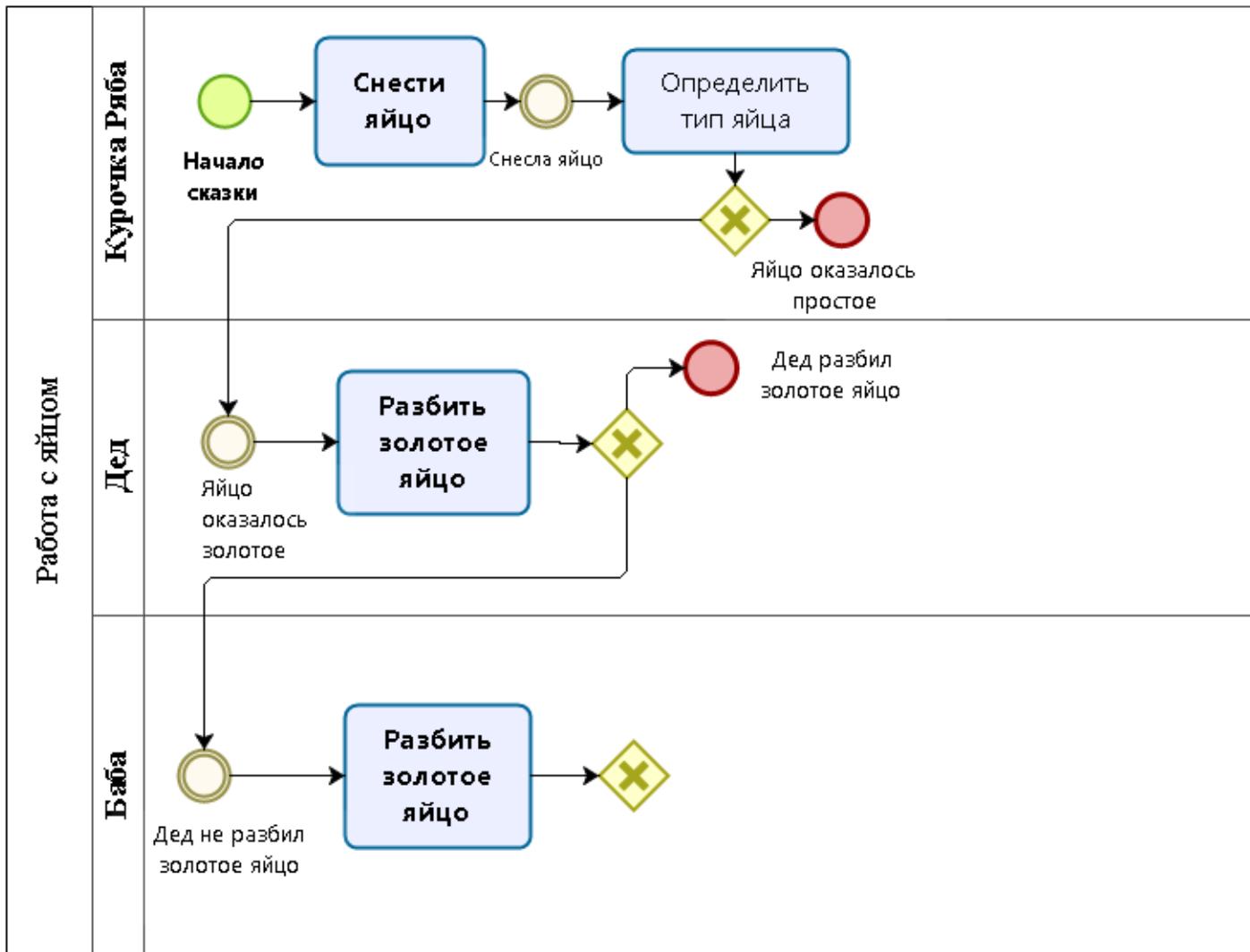


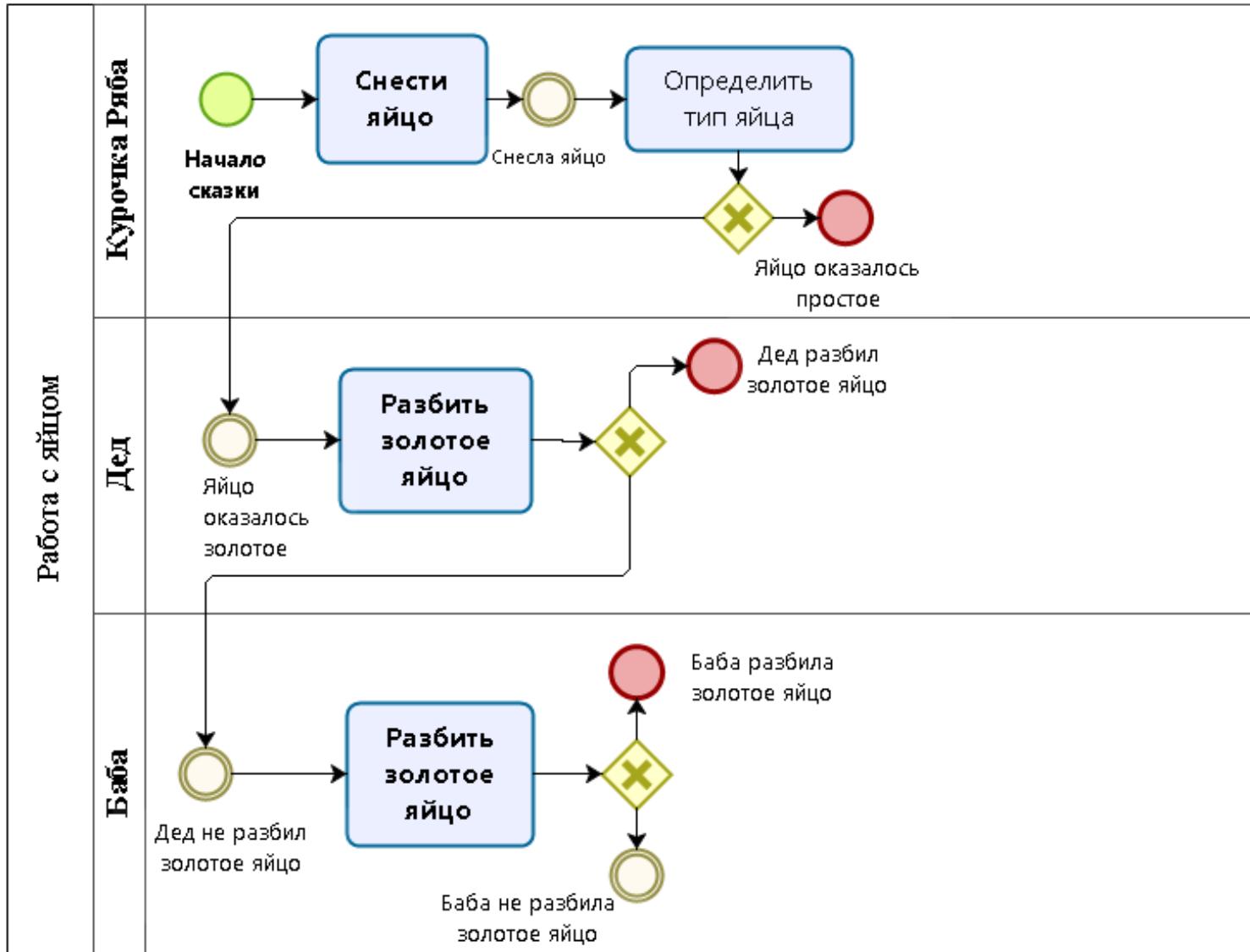
Правила применения шлюзов (недопустимые ситуации)

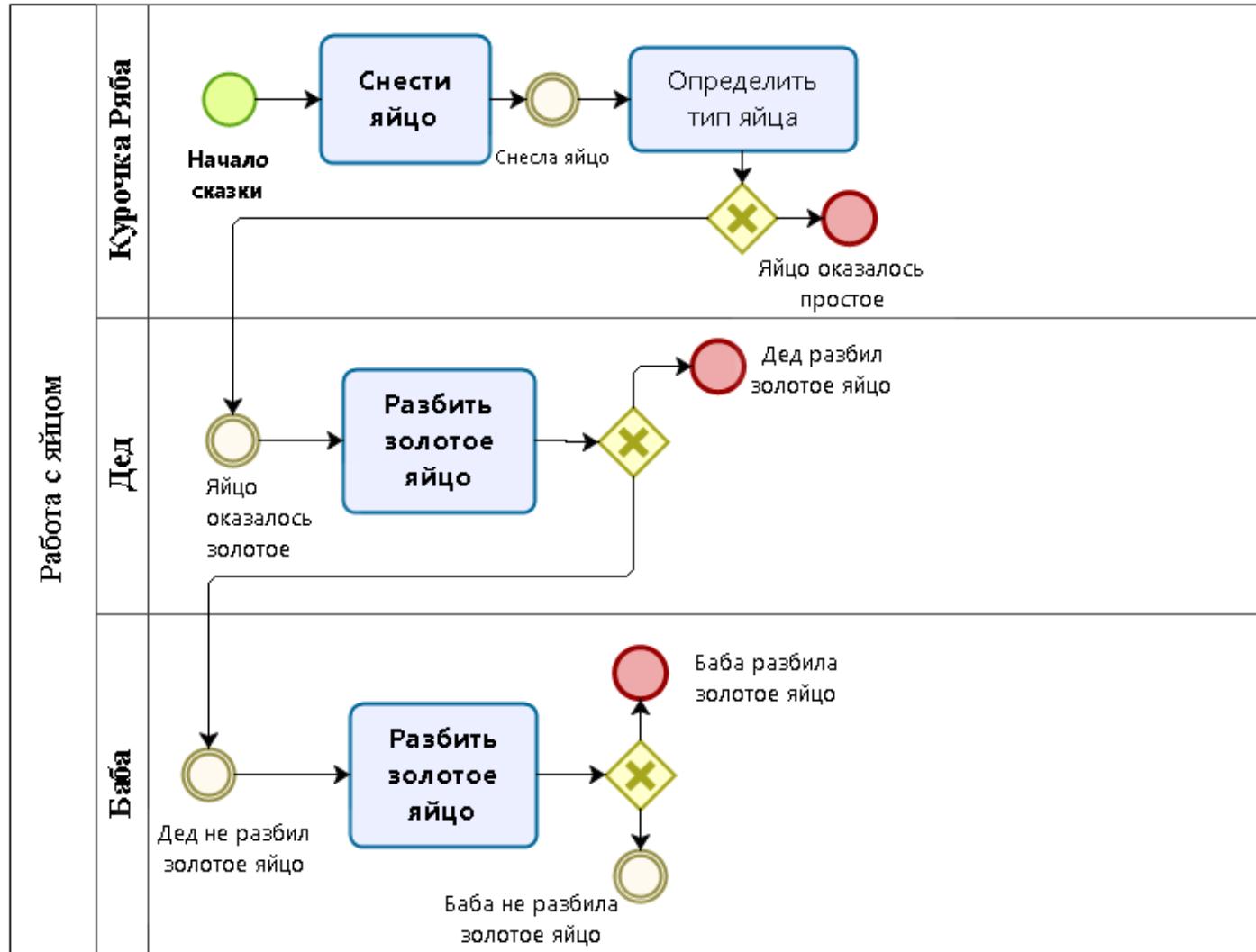


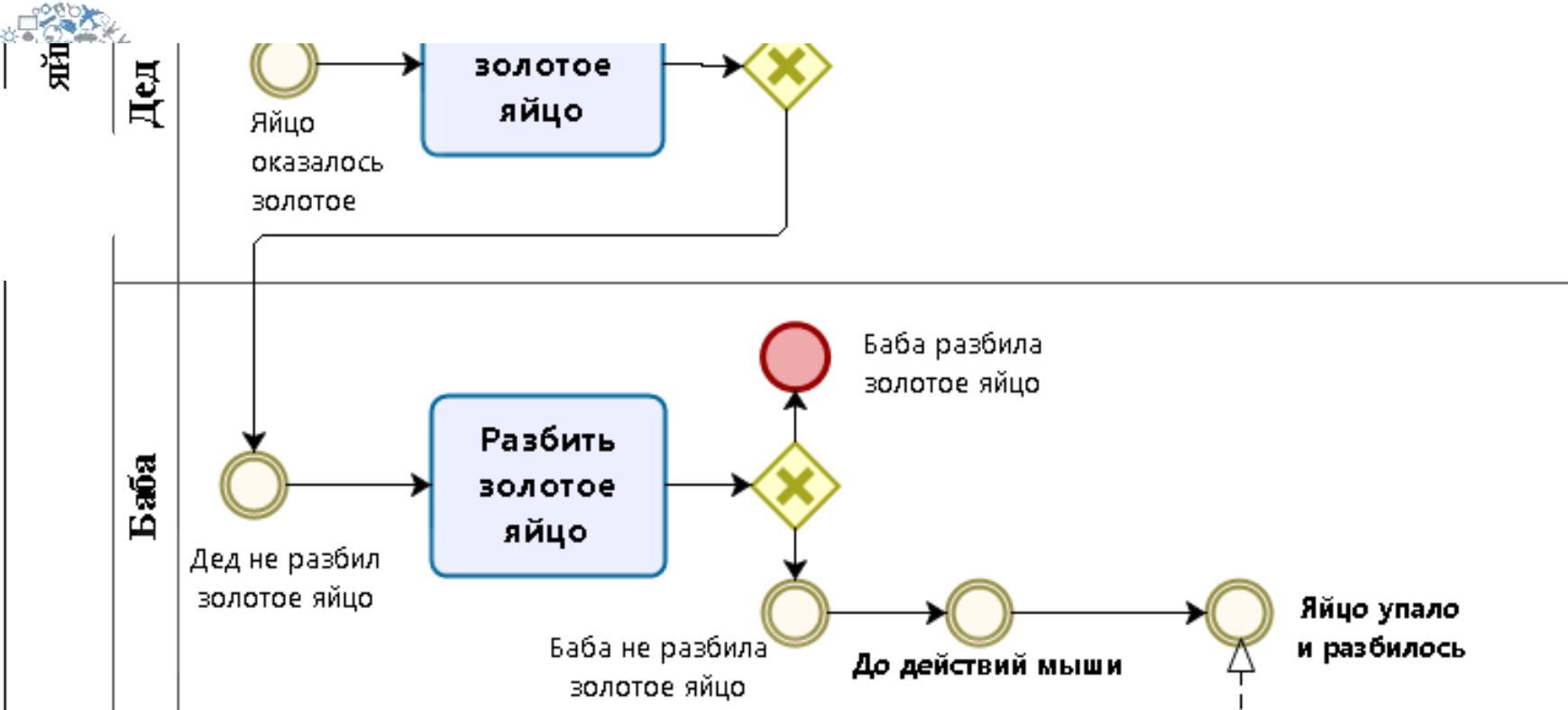






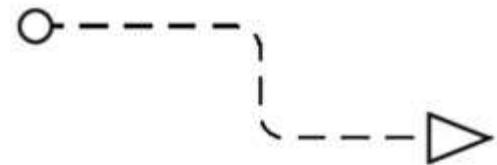




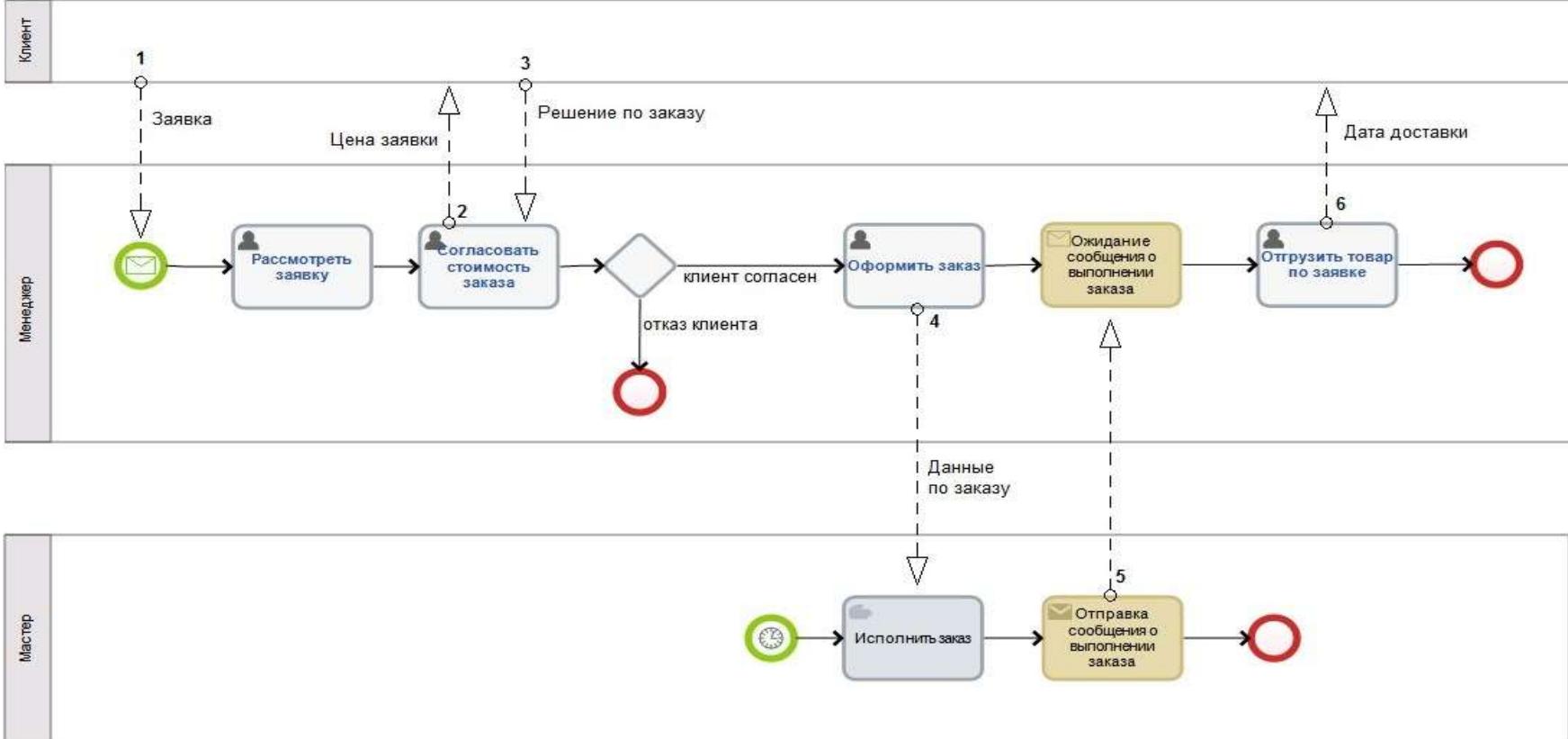




Соединяющие объекты

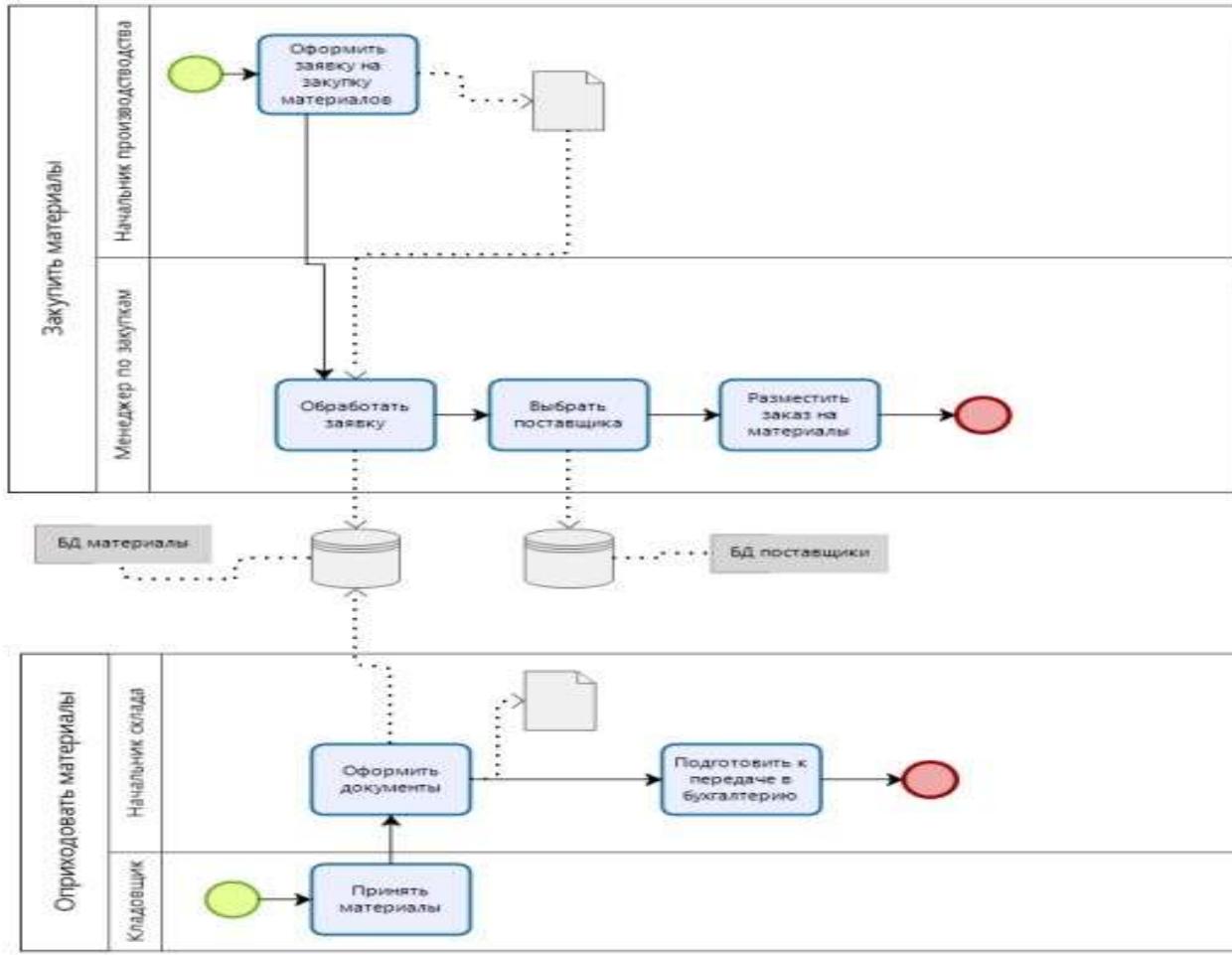


Поток сообщений ДОЛЖЕН соединять два Пула как между собой, так и Элементами потока, расположенными внутри этих Пулов. Однако он не может соединять два элемента, расположенные внутри одного и того же Пула.

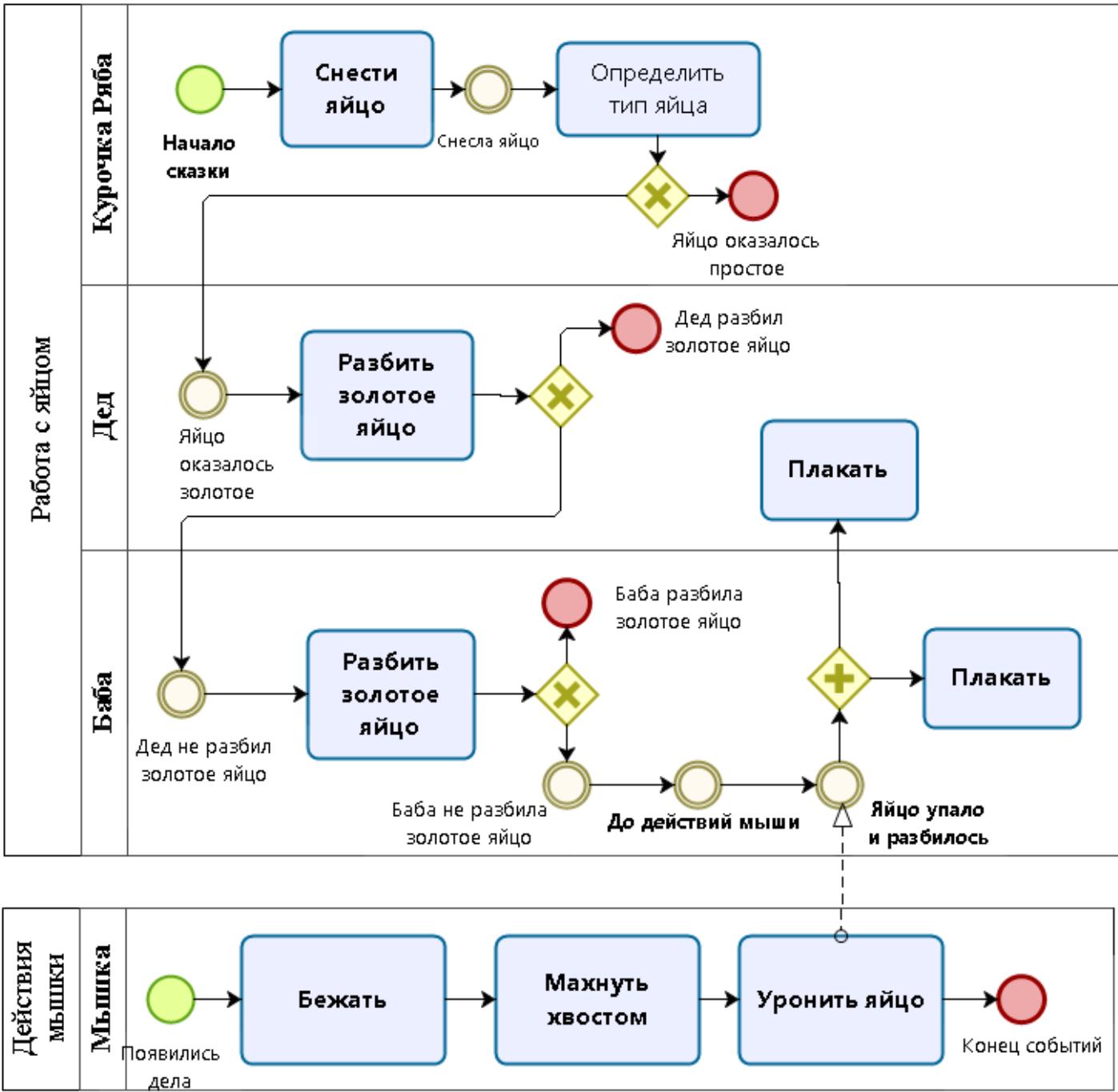


Поток сообщений может присоединяться непосредственно к границам Пула, особенно, если Пул не содержит какой-либо информации о деталях Процесса (например, является «черным ящиком»).

Поток сообщений также может пересекать границы Пула и присоединяться к Элементам потока, расположенным в данном Пуле



Связи ассоциации (пунктирные стрелки) имеют направления. Если стрелка направленной ассоциации идет от задачи к артефакту — это означает, что артефакт был сформирован или дополнен в результате этой задачи. Например, данные были записаны в базу данных или сформирован новый документ. Если стрелка направленной ассоциации идет от артефакта к задаче — это означает, что в задаче используется этот артефакт или его часть, например, выгрузка данных из базы данных, заполнение или передача документа.





Жили-были дед да баба. И была у них Курочка Ряба.

Снесла курочка яичко, да не простое - золотое.

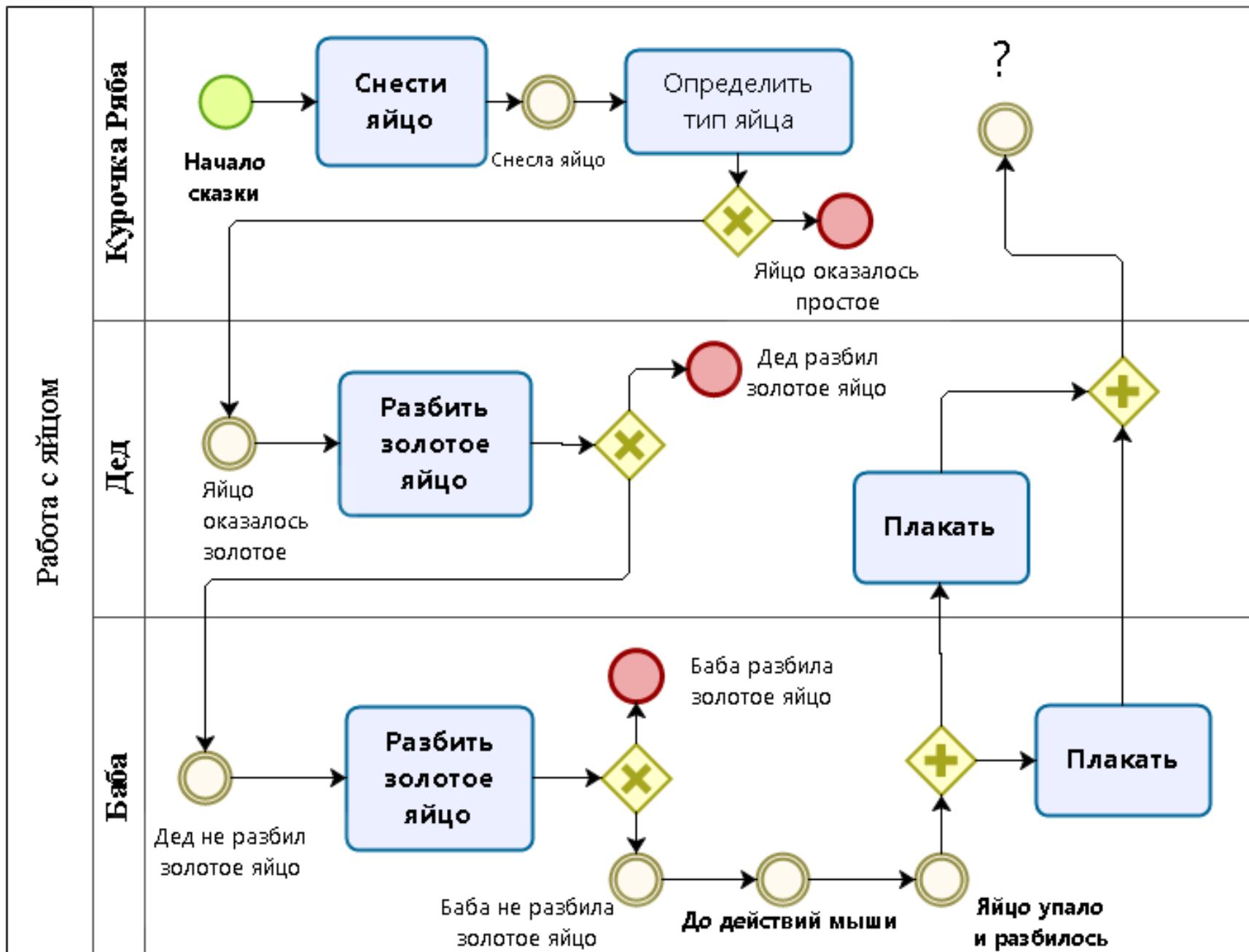
Дед бил - не разбил.

Баба била - не разбила.

А мышка бежала, хвостиком махнула, яичко упало и
разбилось.

Плачет дед, плачет баба и **говорит** им Курочка Ряба:

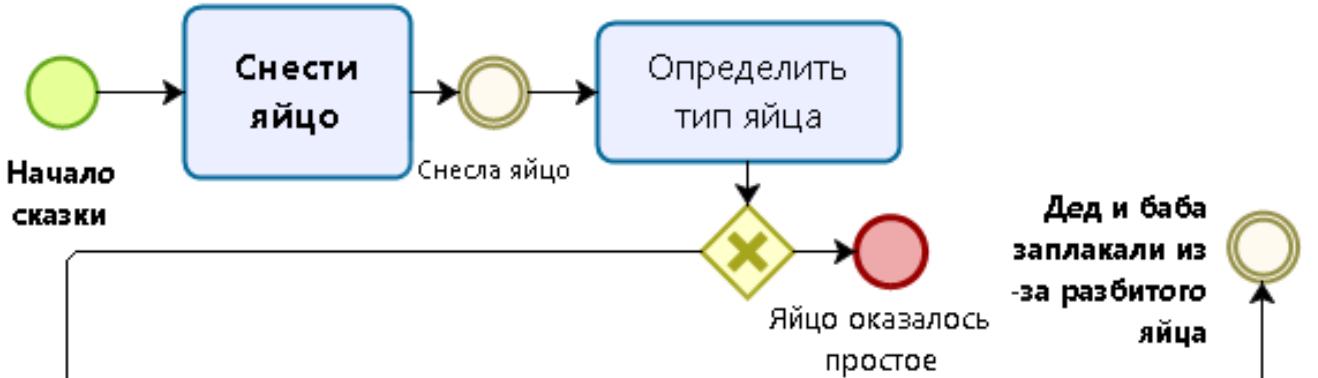
- Не плачь, дед, не плачь, баба: снесу вам новое яичко не
золотое, а простое!



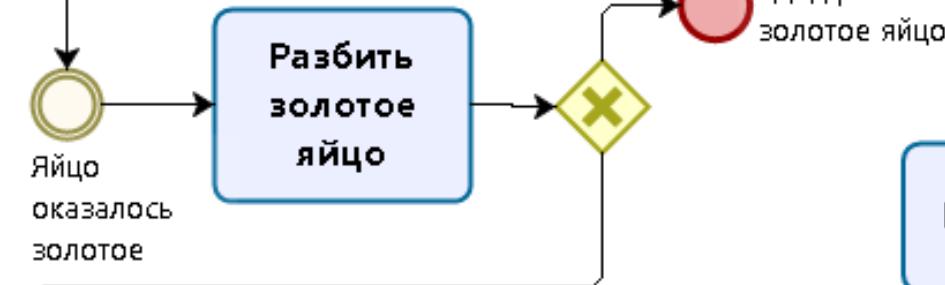


Работа с яйцом

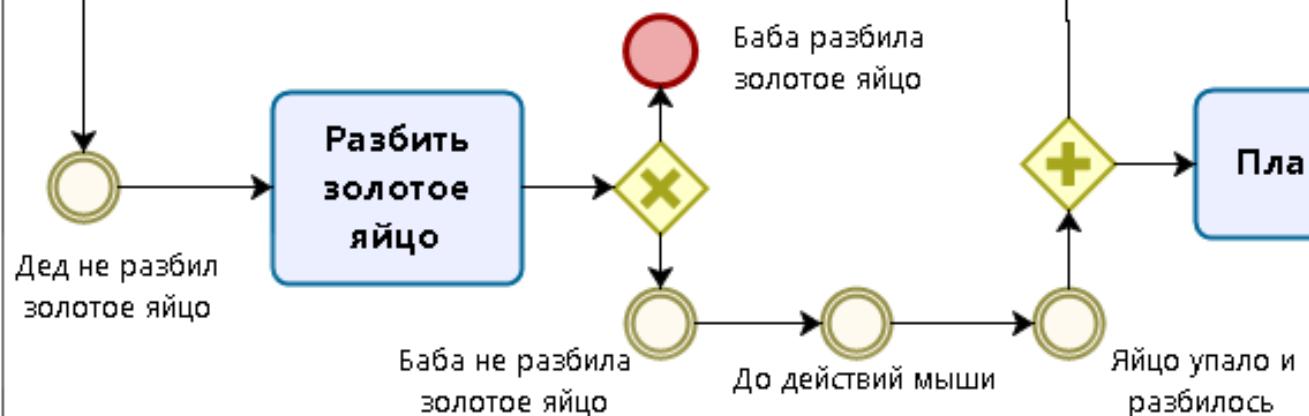
Курочка Ряба



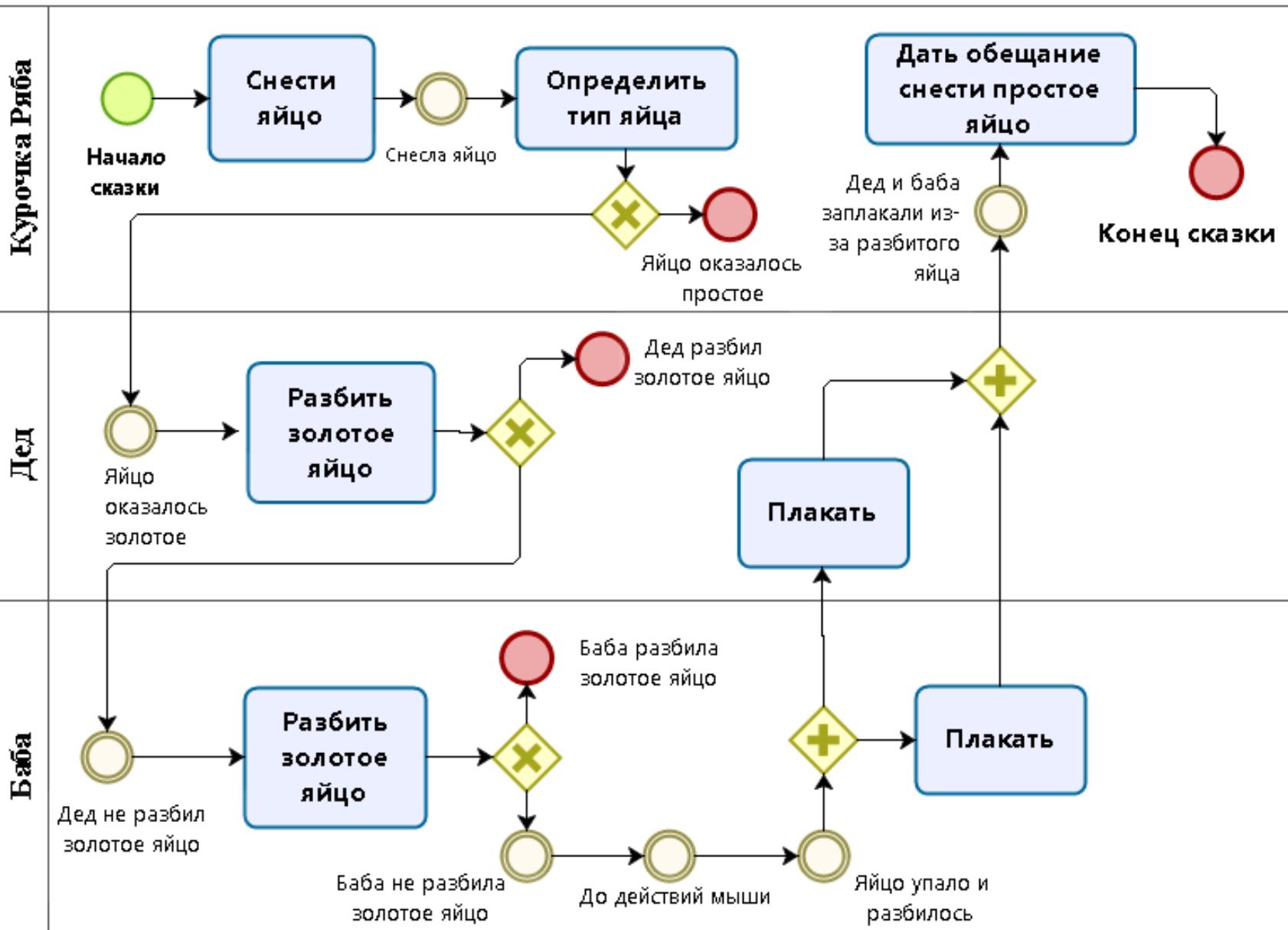
Дед

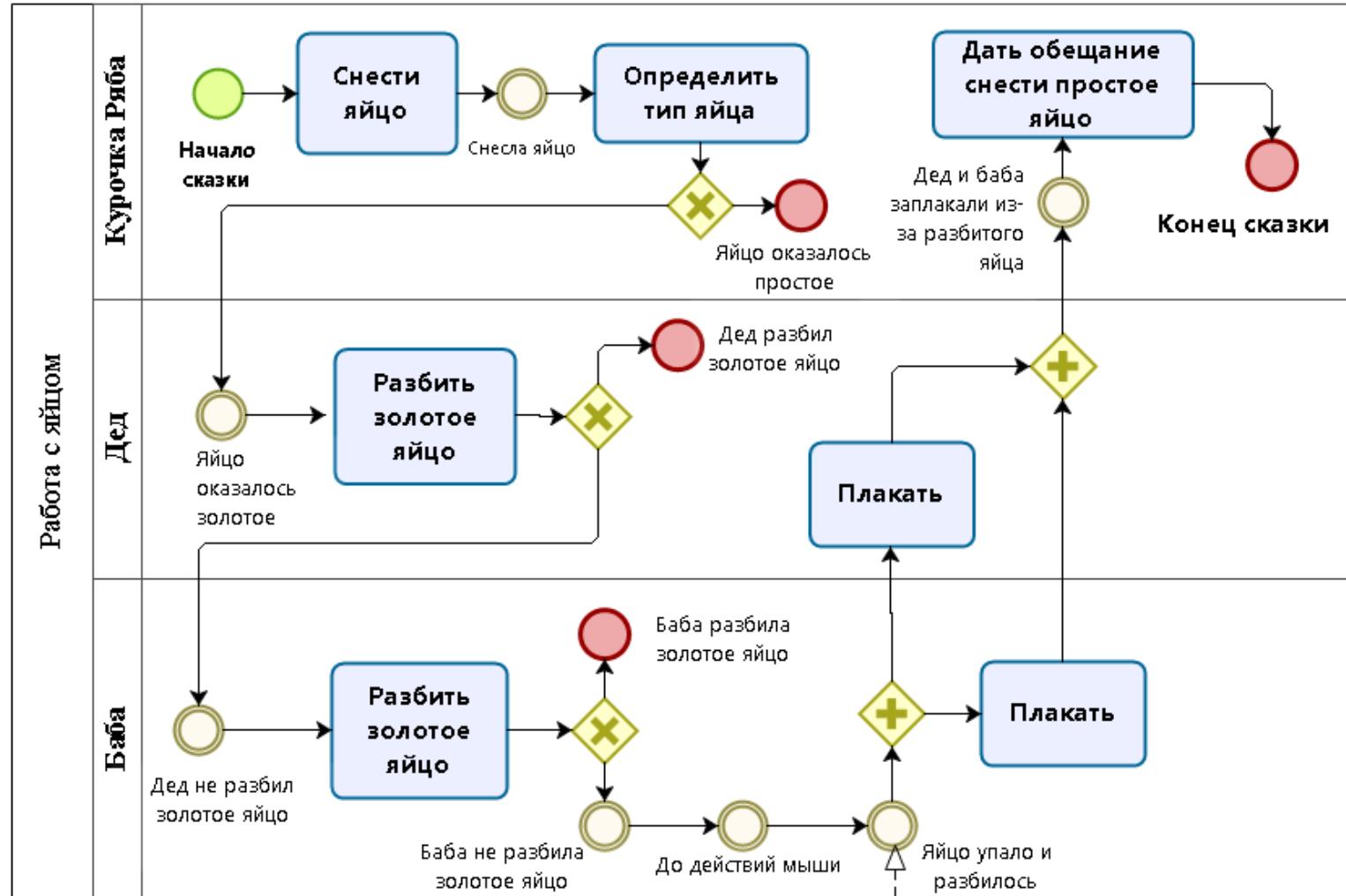


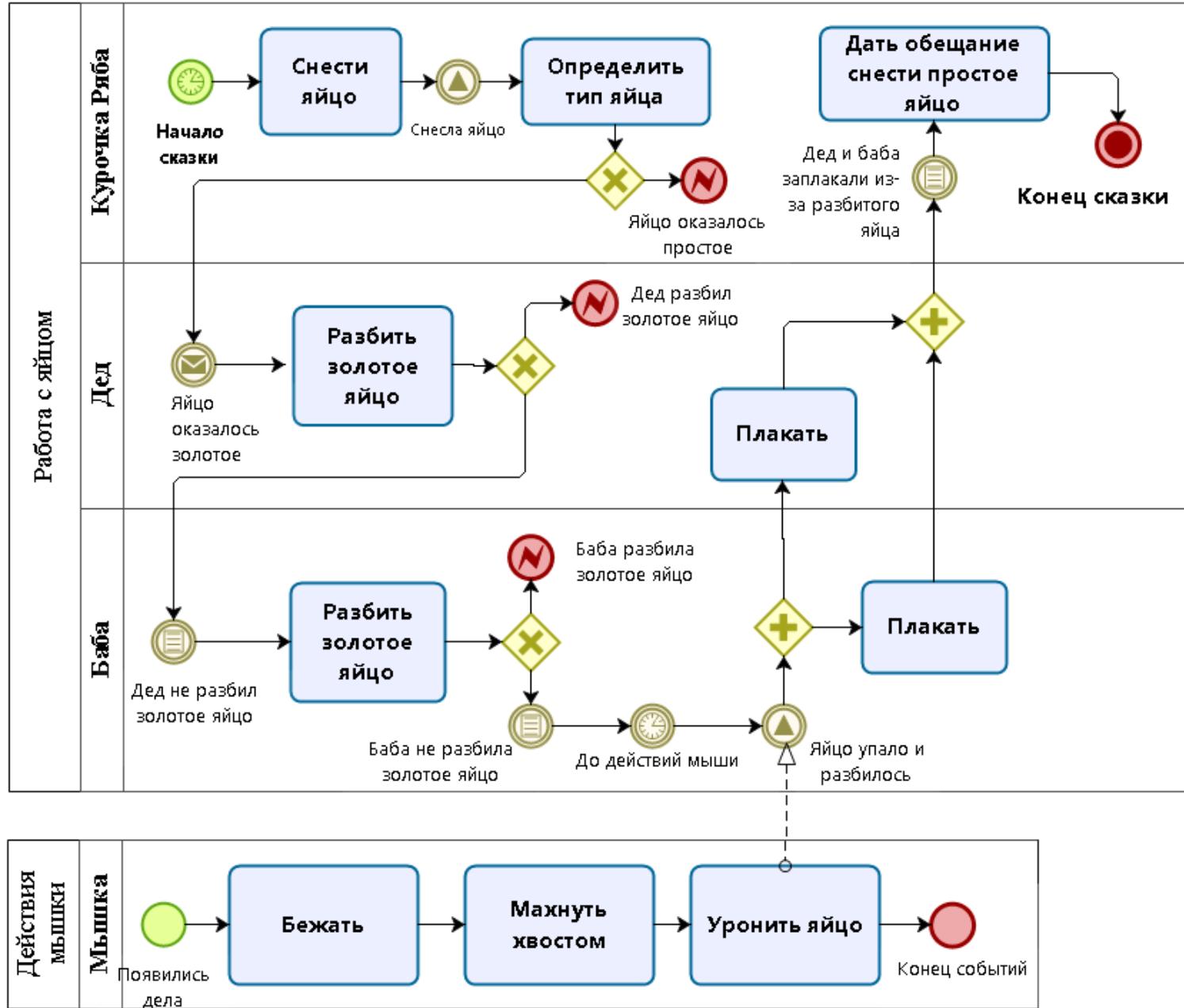
Баба

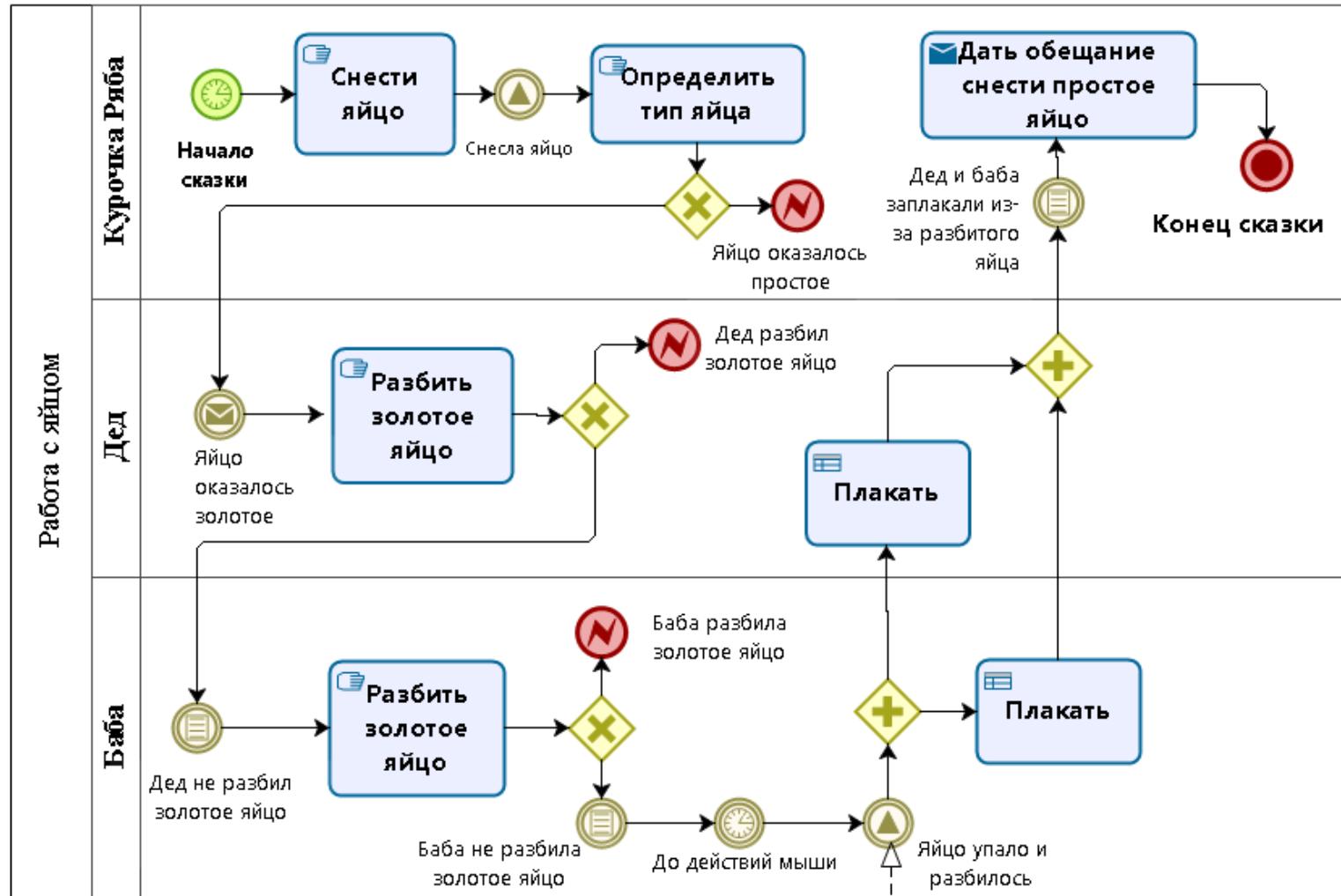


Работа с яйцом



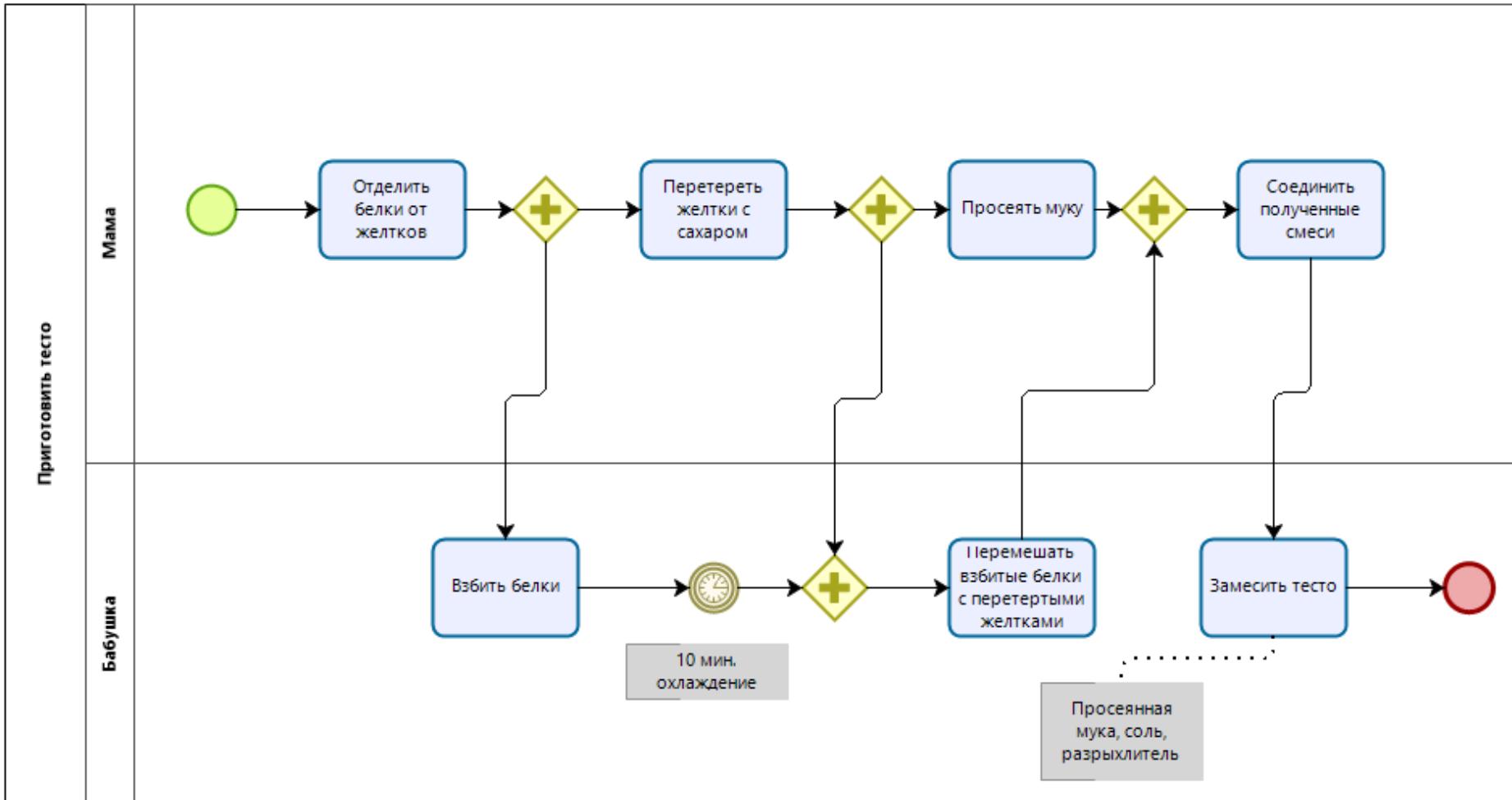








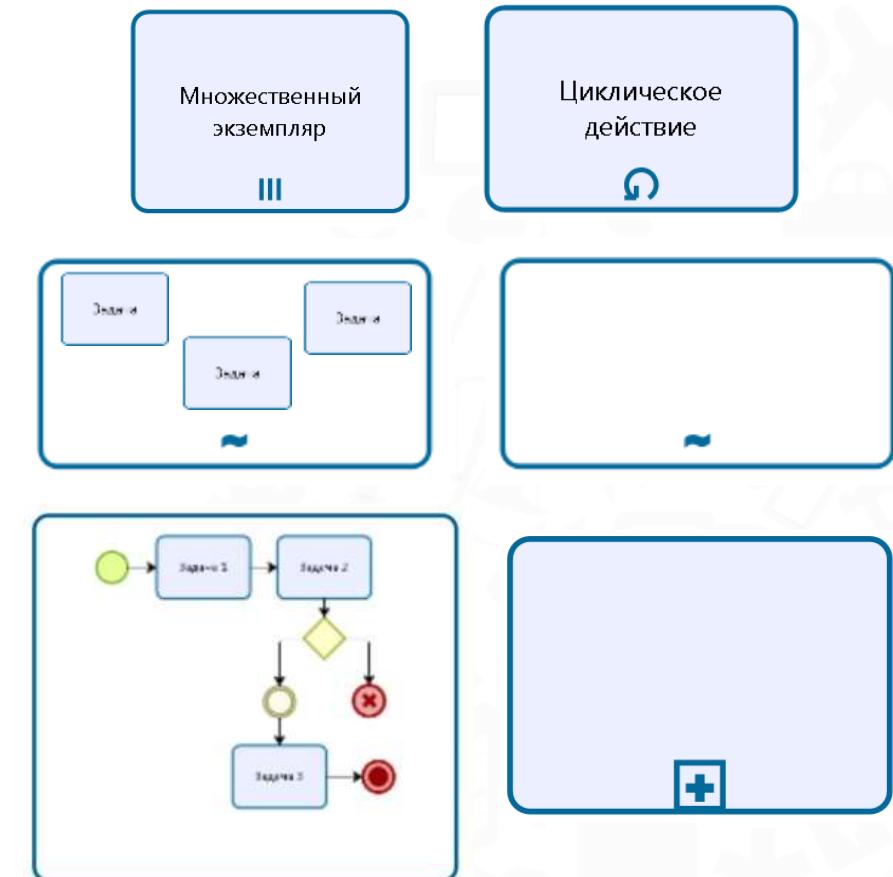
Есть ли ошибки на слайде?





Объекты потока управления. Действия

- Задание/задача (*task*)
- Множественные экземпляры (*multiple instances*)
- Циклическое действие (*loop activity*)
- Развёрнутый подпроцесс (*expanded subprocess*)
- Свёрнутый подпроцесс (*collapsed subprocess*)
- Ad-hoc-подпроцесс (*ad-hoc subprocess*)





Объекты потока управления. Действия

Элемент «Задача»

Элемент «Задача» используется для представления на диаграмме процесса действий, выполняемых в рамках описываемого бизнес-процесса (например, подготовка документа, согласование, подписание, оказание консультационной услуги, изготовление образца продукта и т.д.)



Объекты потока управления. Действия

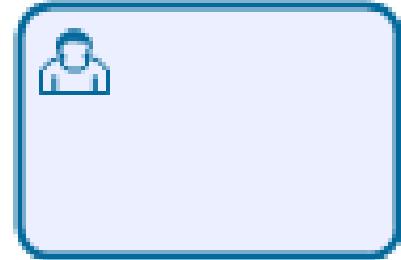
Типы элемента «Задача»

1. Пользовательская задача
2. Ручное выполнение
3. Задача-Сценарий и Задача-Сервис
4. Задача Бизнес-правило
5. Получение сообщения
6. Отправка сообщения

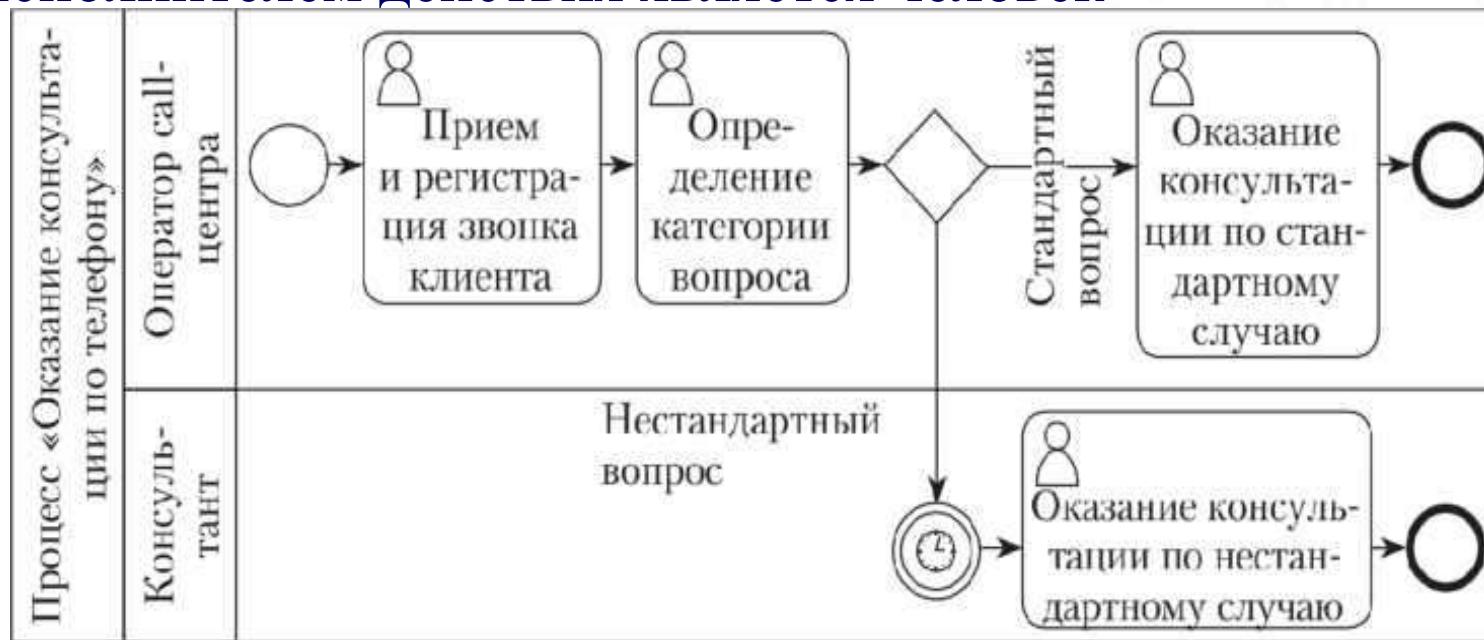


Объекты потока управления. Действия

«Пользовательская задача»



«Пользовательская задача» применяется тогда, когда исполнителем действия является человек





Объекты потока управления. Действия

Задача «Ручное выполнение»



«Ручное выполнение» применяется для отражения действий, которые выполняются людьми без использования каких-либо механизмов исполнения или информационных систем.

Например, доставка заказа или задача «почистить мандарин»)





Объекты потока управления. Действия

Задача «Сценарий»

«Сценарий» позволяет описывать действия, которые осуществляются BPMS-системой в автоматическом режиме и без участия человека.

*Задача-Сценарий — это автоматическая
процедура*



Задача «Сервис»

Задача, предназначенная для оказания услуги, которая может являться как web-сервисом, так и автоматизированным приложением





Объекты потока управления. Действия

Задача «Бизнес-правило»

Задача «Бизнес-правило» — это операция, которая запускает действие какого-то правила.



Сценарии и правила схожи. И то, и другое создается и известно заранее. Но есть и разница.

Сценарий – ряд действий, который нужно выполнить, чтобы завершить задачу.

Правило же предполагает наличие определенных условий и вариантов действий.

Сценарий выполнять обязательно. Правило дает выбор.



Объекты потока управления. Действия

Задача «Бизнес-правило»



Пример операции, связанной с бизнес-правилом — «Принять решение о закупке».

В качестве Правила будет выступать условие — *закупка разрешена в рамках определенной суммы*.

Если сумма закупки выше, то нужно согласовать с вышестоящим руководством.

В данном случае операция «Правило» может запустить операцию «Сценарий».



Получение и отправка сообщения



Получение
сообщения



Отправка
сообщения



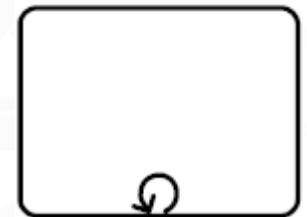
Объекты потока управления. Действия

Маркеры действий

Маркер отражает поведения действия во время выполнения, то есть будут ли они повторяться или будут выполнены единожды.

Существуют два вида циклов: *Стандартный* и *Многоэкземплярный*.

Графически цикличность отображается в виде небольшого маркера в центре нижней части фигуры.



Стандартный цикл

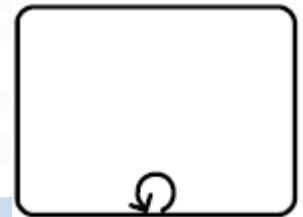


Объекты потока управления. Действия

Маркер Цикла (Стандартный цикл)

Циклическое действие выполняется, пока условие цикла верно. Условие цикла может проверяться до или после выполнения действия.

В основе стандартного повторения лежит простая мысль. Процесс будет повторяться до тех пор, пока не получен нужный результат. Например, вы будете добавлять соль в блюдо до тех пор, пока не посчитаете, что ее достаточно. При этом нужно не забывать указывать условия выхода из повторения.





Объекты потока управления. Действия

Маркер множественных экземпляров (Многоэкземплярность)

Sequential



Parallel



Три горизонтальные полоски в центре нижней части фигуры указывают на последовательную многоэкземплярность

Три вертикальные полоски в центре нижней части фигуры указывают на параллельную многоэкземплярность



Объекты потока управления. Действия

Маркеры последовательных и параллельных множественных экземпляров

Множественные экземпляры действия показывают, что одно действие выполняется многократно, по одному разу для каждого объекта. Например, для каждого объекта в заказе клиента выполняется один экземпляр действия.

Данный тип повторения отличается от Цикла, где количество повторений известно и задано заранее. Т.е. для завершения процесса и перехода к следующему, его нужно выполнить несколько раз. Например, нужно дважды проверить первичные документы, прежде чем передать в архив.

Sequential



≡

Parallel



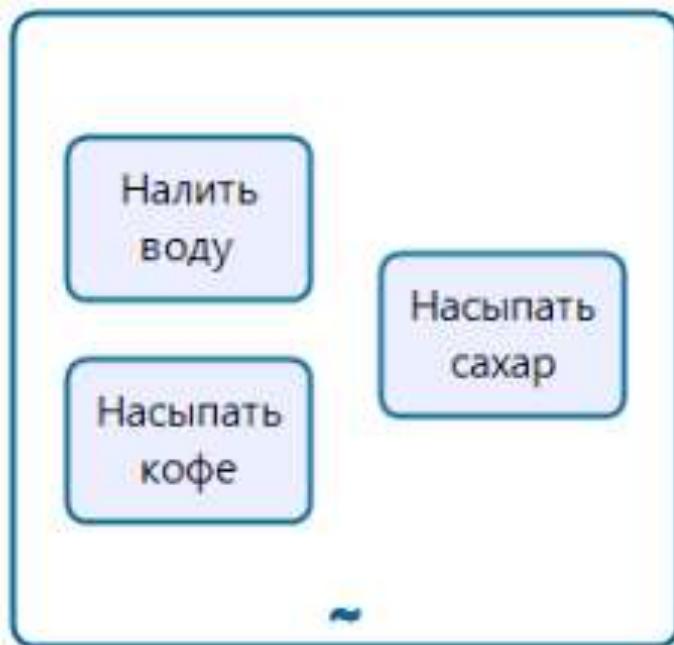
|||



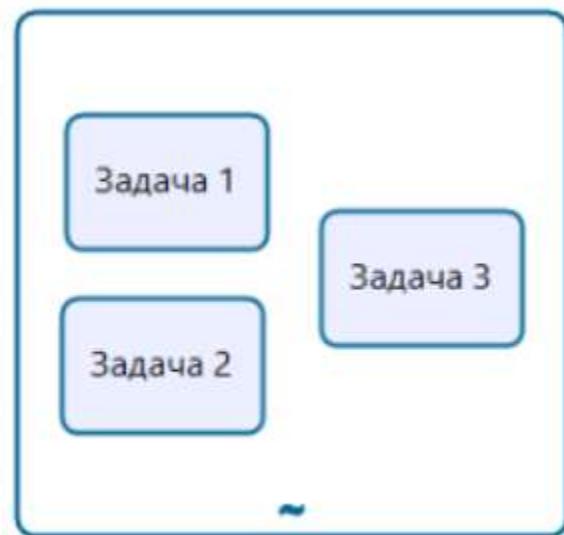
Ad-хос-подпроцесс

(*ad-hoc subprocess*) содержит задания.

Задания выполняются до тех пор, пока не выполнено условие завершения подпроцесса.



Приготовить кофе



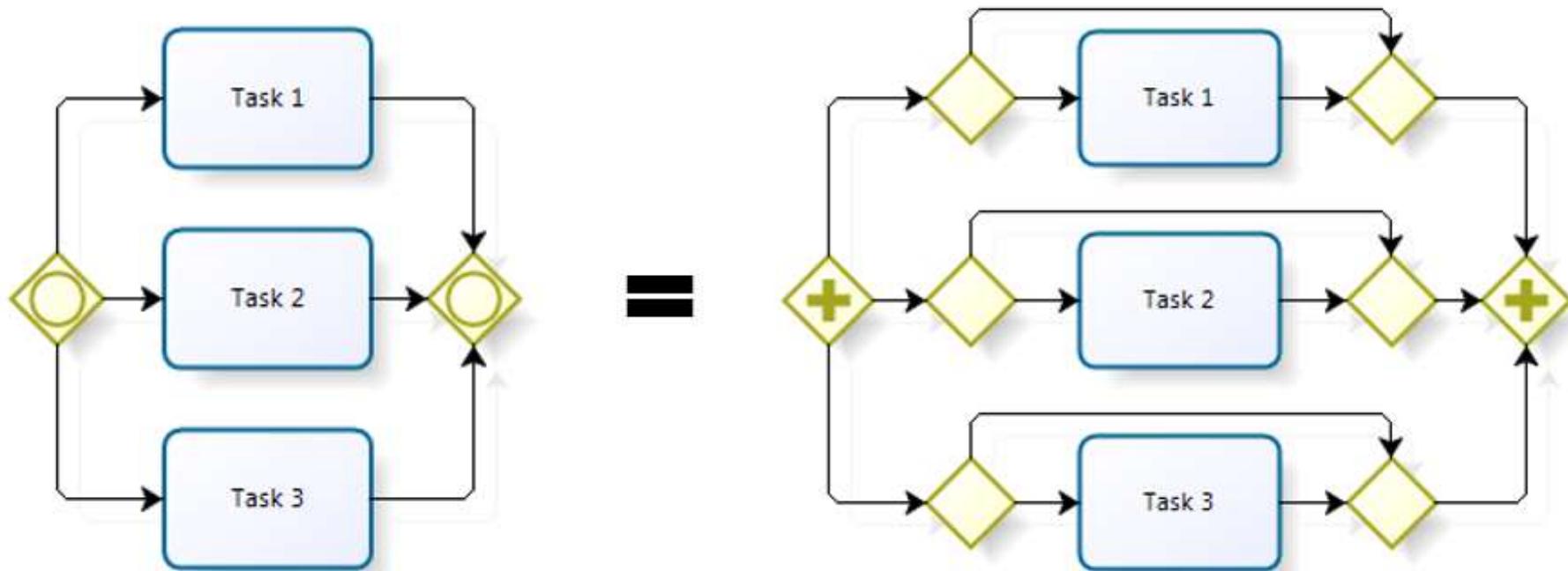
Задача 1

Задача 2

Задача 3



Условный пример на правильность применения шлюзов





Множественный экземпляр

Множественные экземпляры (*multiple instances*) действия показывают, что одно действие выполняется многократно, по одному разу для каждого объекта.

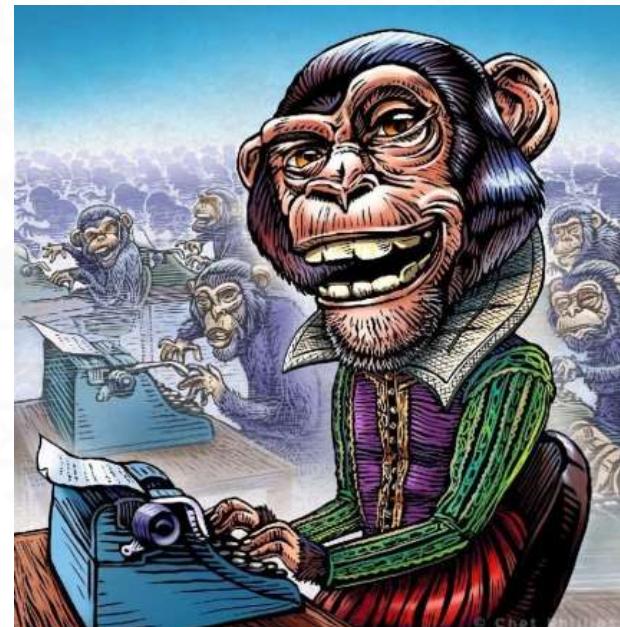
Множественный
экземпляр

III

Выдать обезьяне
печатную машинку

III

«Теорема бесконечных обезьян»

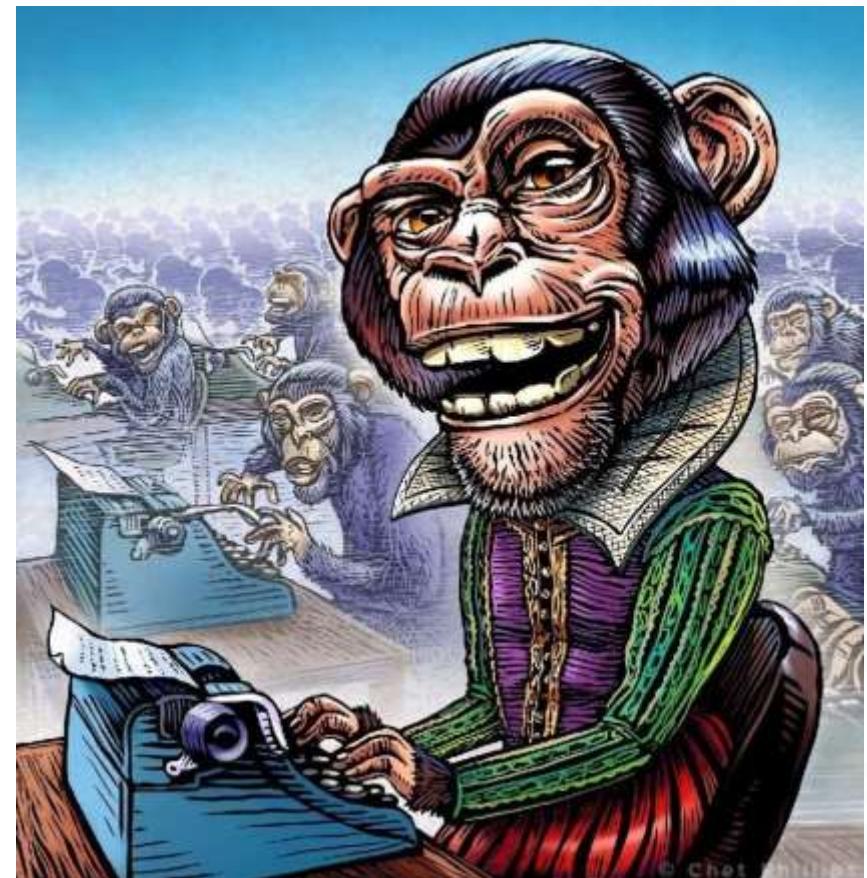




Мысленный эксперимент «Теорема бесконечных обезьян»

Если бесконечное количество обезьян случайным образом нажмут на клавиши бесконечного числа пишущих машинок, в какой-то момент они совершенно точно создадут произведения Шекспира.

Основная идея состоит в том, что бесконечное количество действующих сил и бесконечное время случайным образом создадут все и вся.





Множественный экземпляр

Множественные экземпляры (*multiple instances*)
действия показывают, что одно действие выполняется
многоократно, по одному разу для каждого объекта.

Множественный
экземпляр

III



Появилось
бесконечное
количество обезьян

Выдать обезьяне
печатную машинку

III

«Теорема бесконечных обезьян»

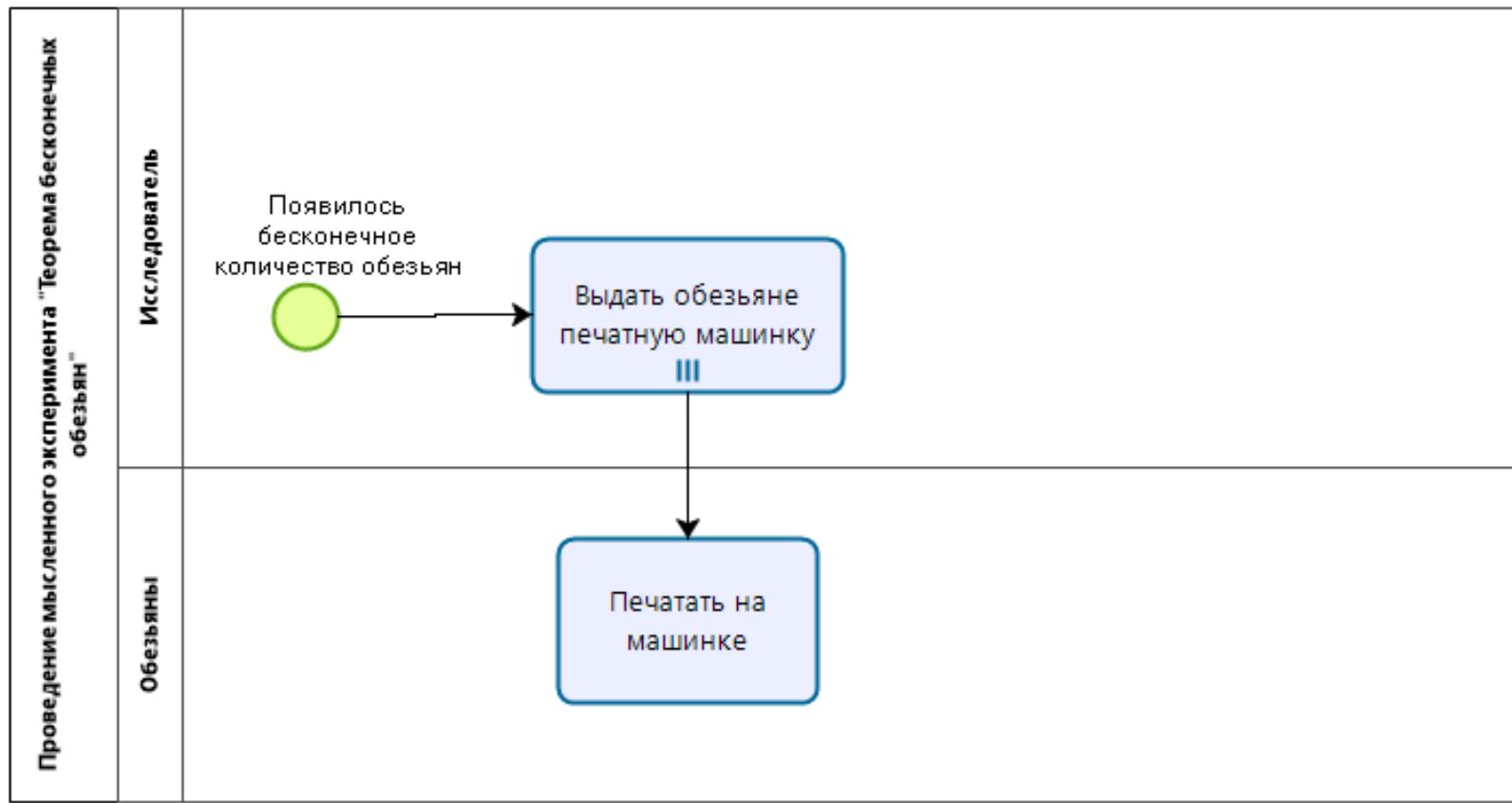


Множественный экземпляр

Множественные экземпляры (*multiple instances*)
действия показывают, что одно действие выполняется
много раз, по одному разу для каждого объекта.

Множественный
экземпляр

III

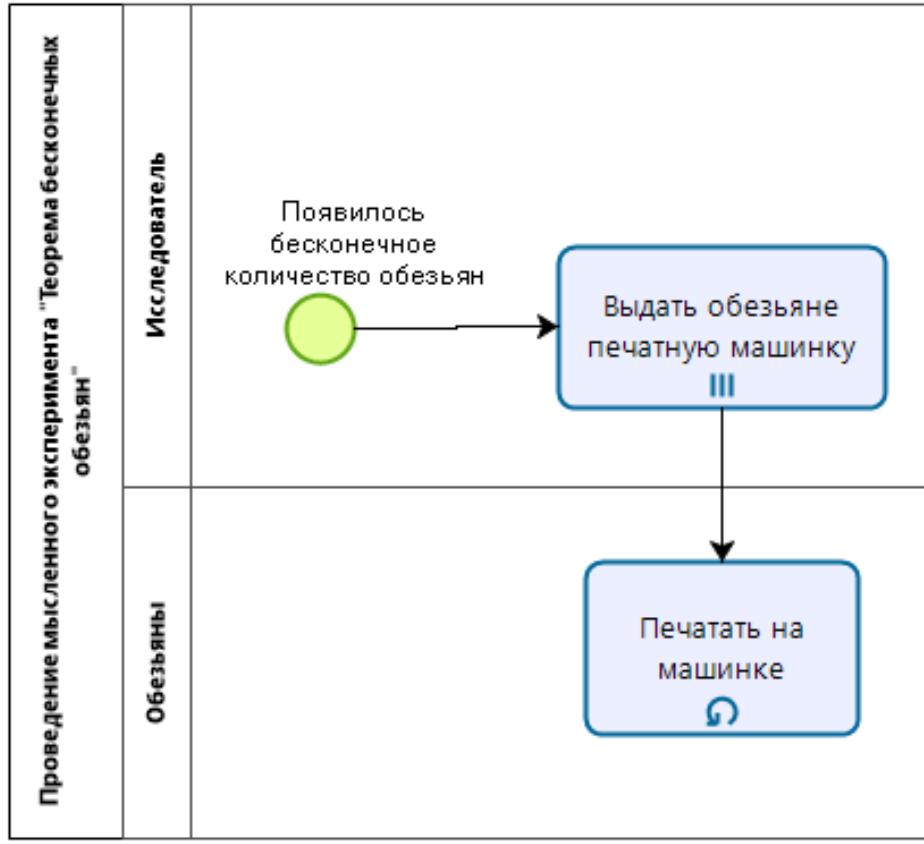




Циклическое действие / Стандартное повторение

(*loop activity*) - процесс будет повторяться до тех пор, пока не получен нужный результат.

Циклическое
действие



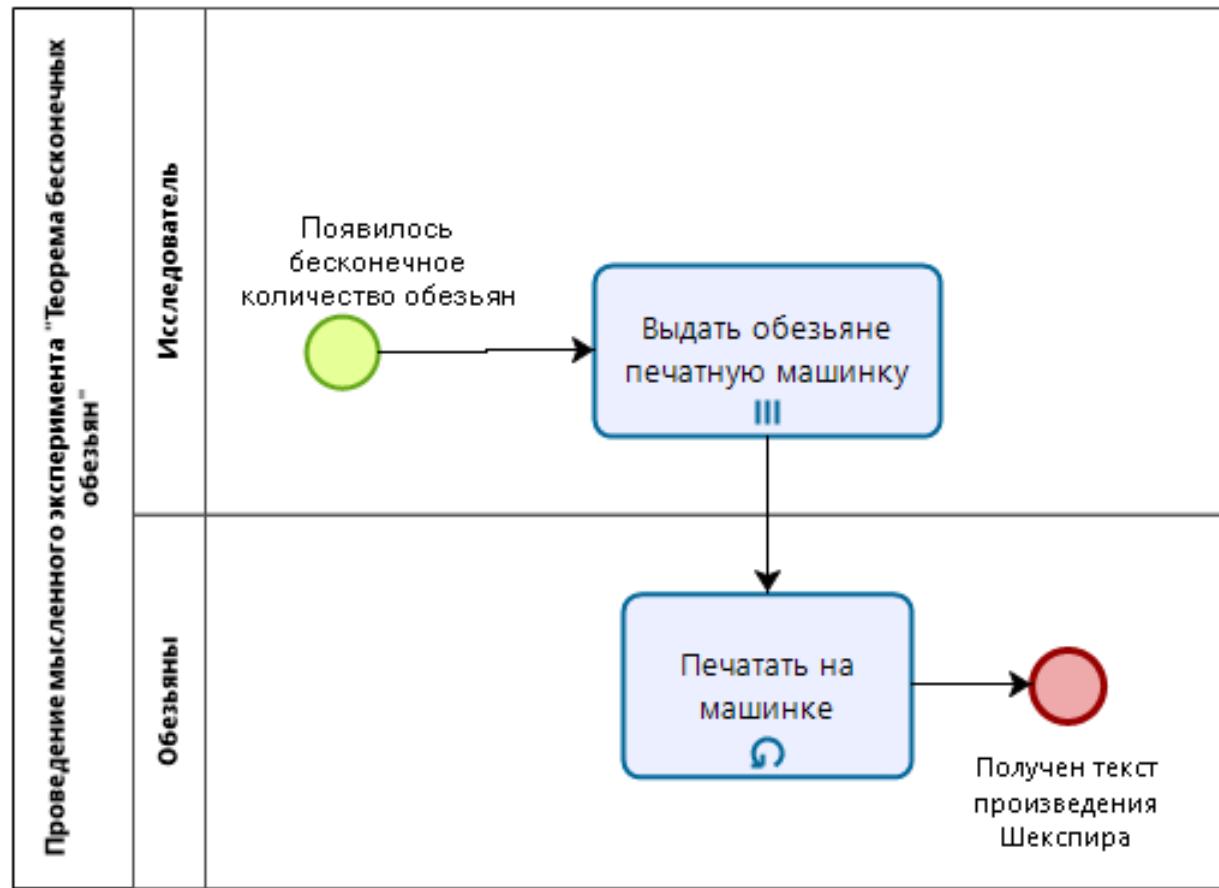


Множественный экземпляр

Множественные экземпляры (*multiple instances*)
действия показывают, что одно действие выполняется
много раз, по одному разу для каждого объекта.

Множественный
экземпляр

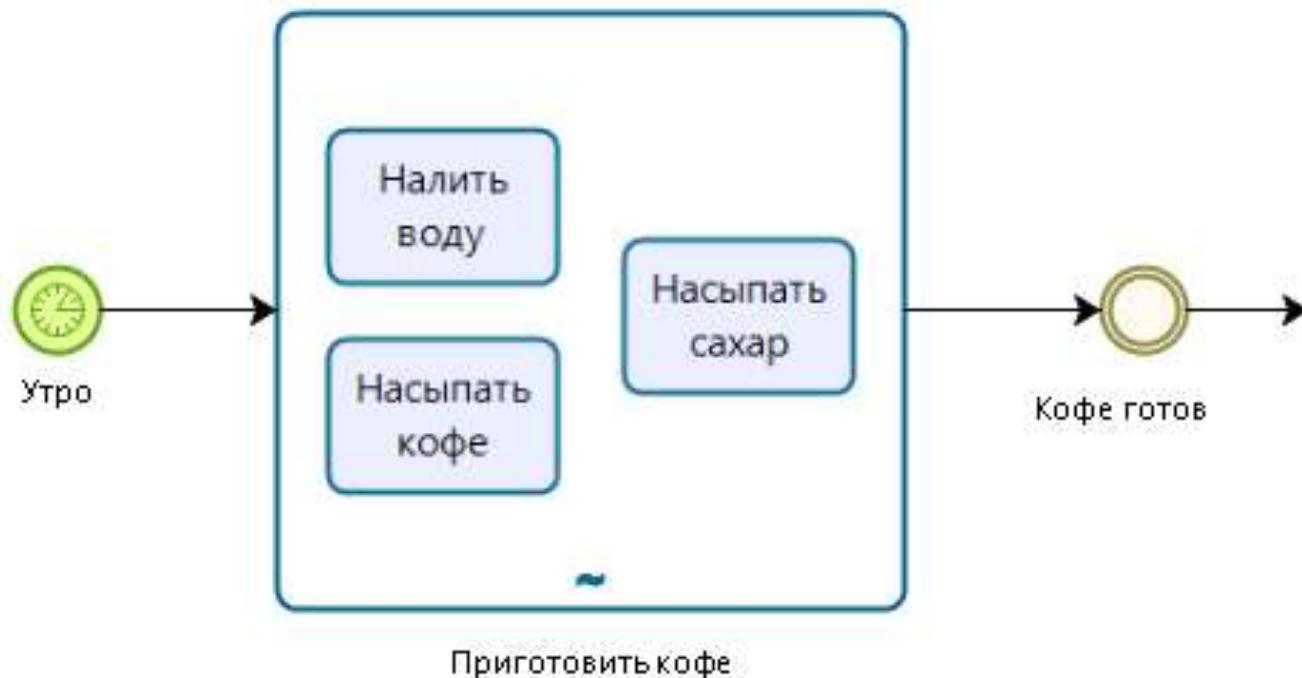
III





Ad-hoc-подпроцесс

(*ad-hoc subprocess*) содержит задания.
Задания выполняются до тех пор, пока не
выполнено условие завершения
подпроцесса.



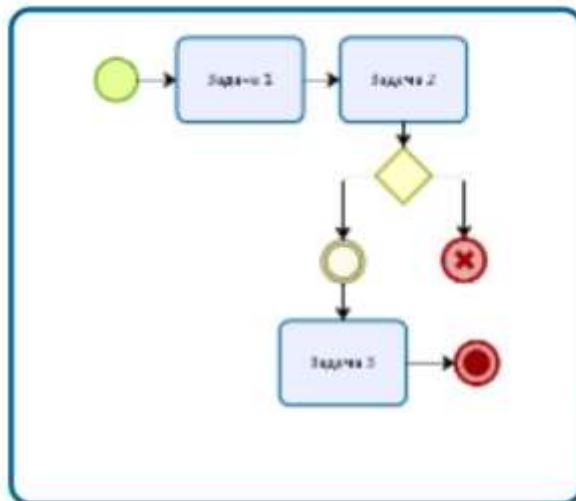


Свернутый и развернутый подпроцессы

Свёрнутый подпроцесс (*collapsed subprocess*) является составным действием, но скрывает детали реализации процесса.



Развёрнутый подпроцесс (*expanded subprocess*) является сложным действием и показывает внутри себя собственную диаграмму бизнес-процессов.





Свернутый подпроцесс

Свёрнутый подпроцесс (*collapsed subprocess*) является составным действием, но скрывает детали реализации процесса.



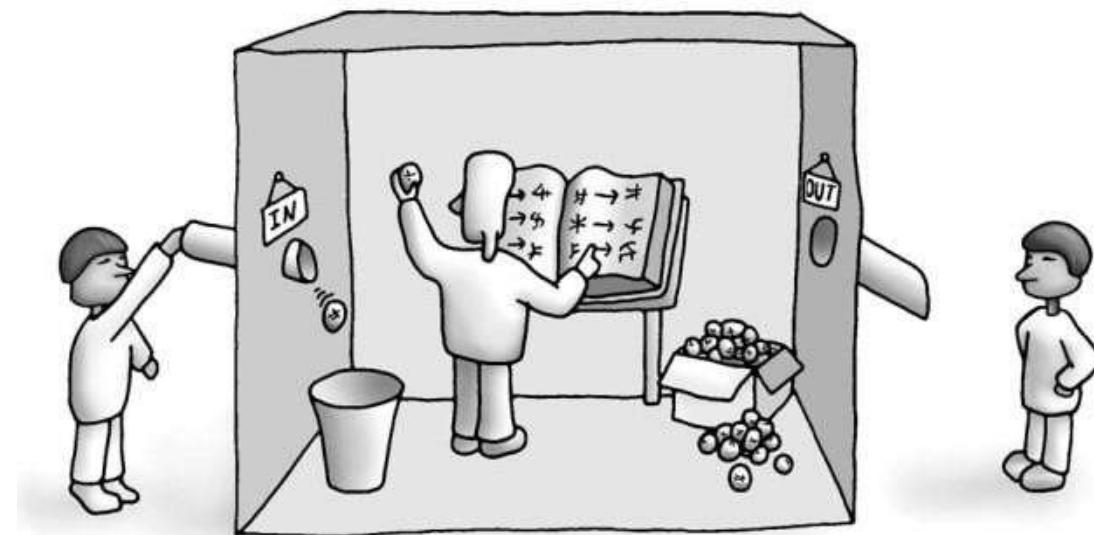


Мысленный эксперимент «Китайская комната»

Человека, говорящего на английском языке заперли в комнате, в которой есть небольшая щель для писем. У человека есть **корзины с китайскими иероглифами и учебник с инструкциями на английском языке**, который поможет перевести с китайского. Через щель в двери ему передают листки бумаги с набором китайских иероглифов. Мужчина может использовать учебник, чтобы перевести фразы и отправить ответ на китайском языке.

Хотя сам он ни слова не говорит на китайском, он может убедить находящихся снаружи, что в совершенстве владеет китайским.

院





Свернутый и развернутый подпроцессы

Свёрнутый подпроцесс (*collapsed subprocess*) является составным действием, но скрывает детали реализации процесса.

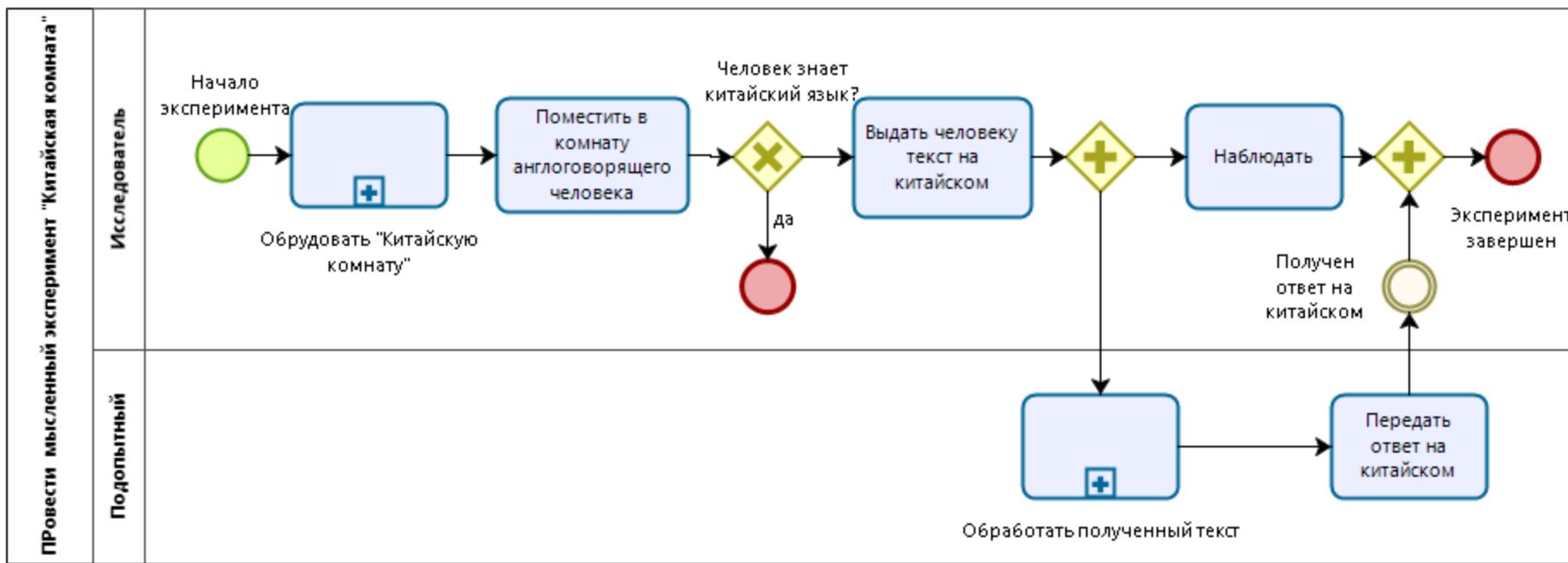


Провести мысленный эксперимент "Китайская комната"	<p>Исследователь</p> <pre>graph LR; Start(()) --> Prep[Начало эксперимента]; Prep --> Place[Поместить в комнату англоворящего человека]; Place --> Decision{Человек знает китайский язык?}; Decision -- да --> Give[Выдать человеку текст на китайском]; Give --> Observe[Nаблюдать]; Decision -- нет --> Process[Обработка полученного текста];</pre> <p>Оборудовать "Китайскую комнату"</p> <p>Человек знает китайский язык?</p> <p>да</p> <p>нет</p> <p>Наблюдать</p> <p>Обработка полученного текста</p>
Подопытный	

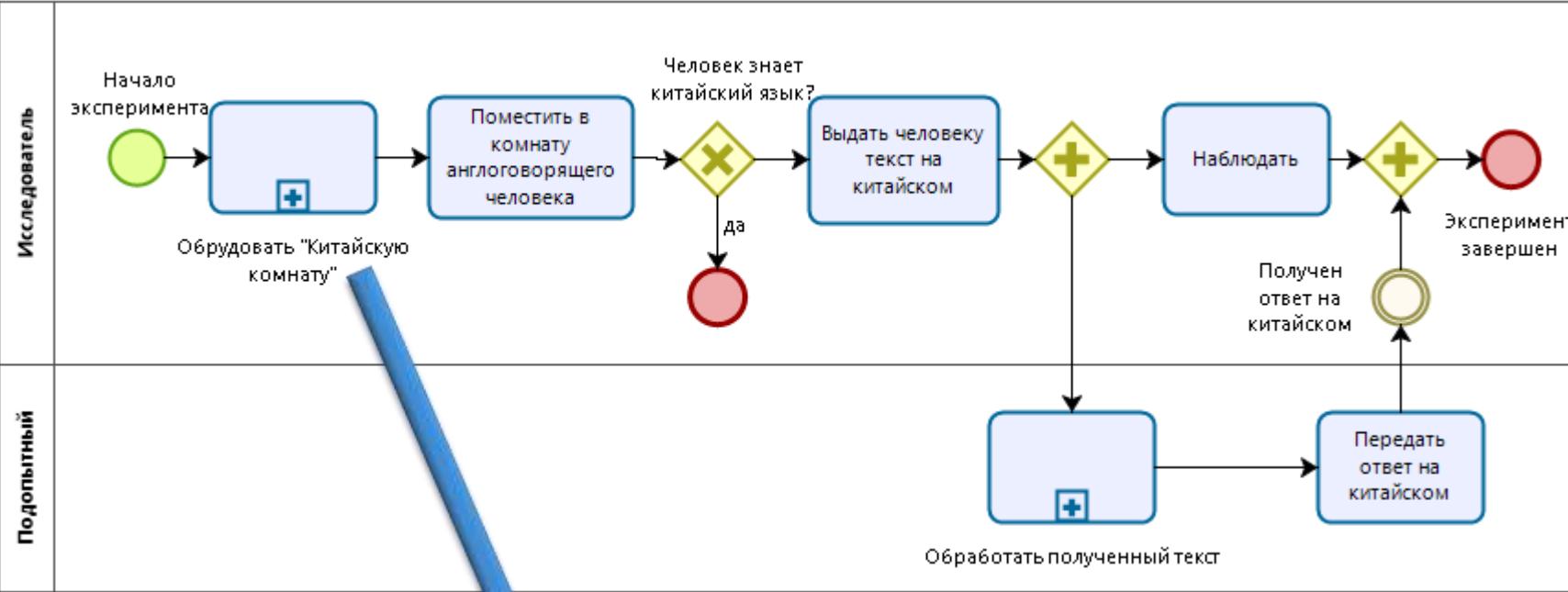


Свернутый и развернутый подпроцессы

Свёрнутый подпроцесс (*collapsed subprocess*) является составным действием, но скрывает детали реализации процесса.



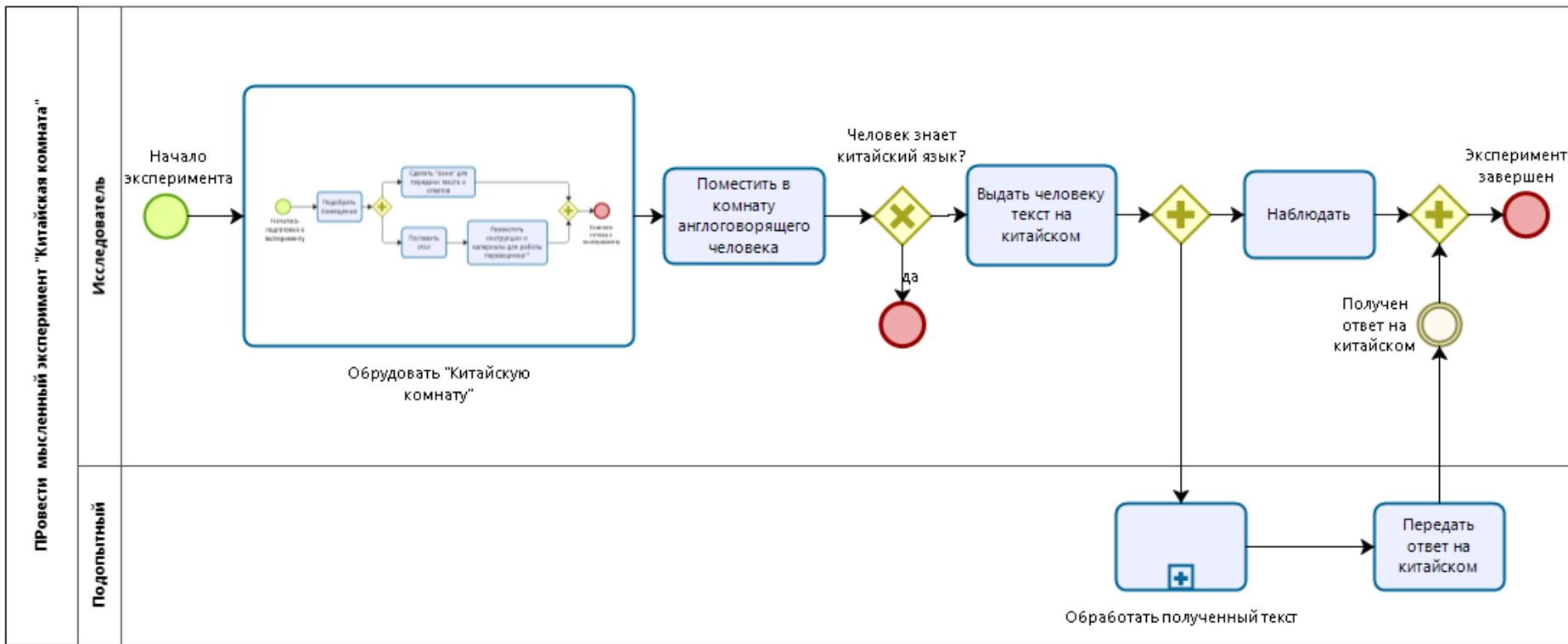
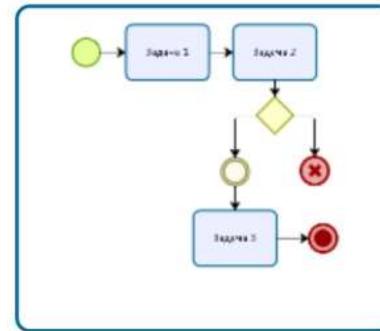
Провести мысленный эксперимент "Китайская комната"





Развернутый подпроцесс

Развёрнутый подпроцесс (*expanded subprocess*) является сложным действием и показывает внутри себя собственную диаграмму бизнес-процессов.



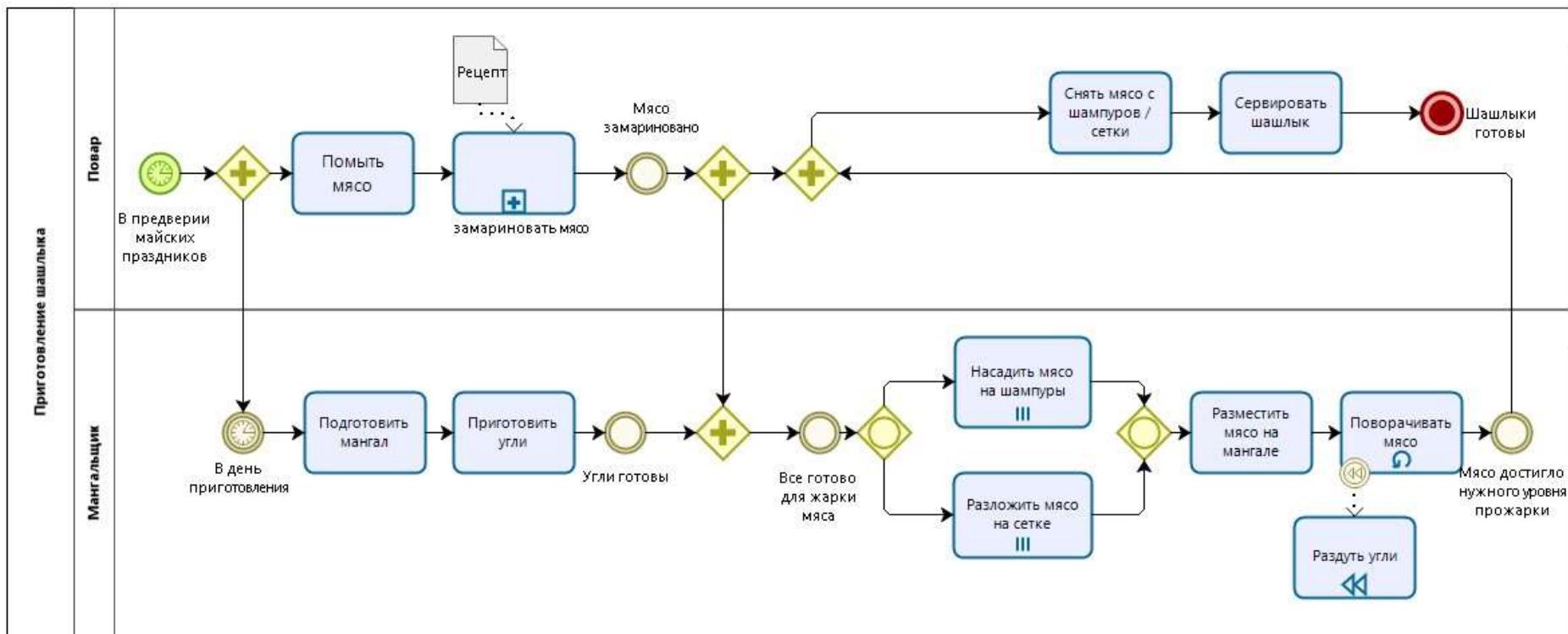


Маркер Подпроцесса

Развернутый подпроцесс содержит в себе самостоятельную цепочку объектов потока управления (событий, задач, логических операторов).

Внутри развернутого процесса будут свои стартовое и завершающее событие (или несколько завершающих событий).

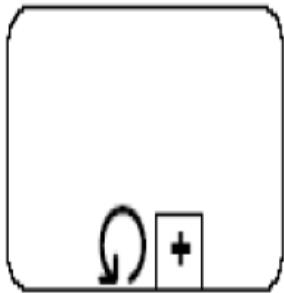
Важно, что внутри подпроцесса не может располагаться отдельный пул или дорожки: исключение будут составлять случаи, когда подпроцесс используется для ссылки на отдельный самостоятельный процесс, для которого строится своя диаграмма (с пулом, дорожками т.д.).



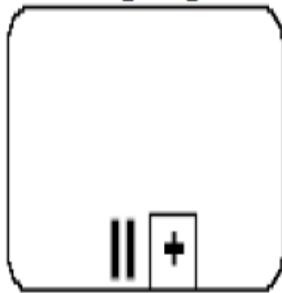


Примеры сочетания маркеров

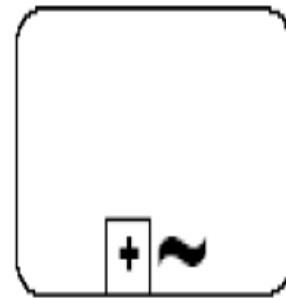
Маркер
Цикла



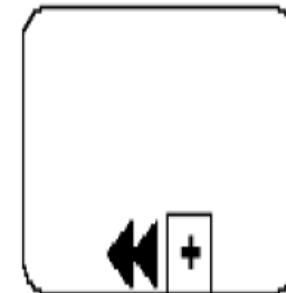
Многоэкземплярный
Маркер



Маркер
Ad Hoc



Маркер
Компенсации





Элементы «Объект» и «Хранилище данных»

Данные элементы используются для идентификации различных компонентов (обычно информации) в ходе реализации процесса в BPMN-нотации



Объект данных



Хранилище данных



Элемент «Объект»



«Объект данных» на схеме процесса показывает информацию, которая обрабатывается в ходе реализации процесса и не передается за пределы процесса, не используется после его завершения.

Данная информация не имеет управляющего воздействия и не влияет на ход выполнения процесса, она носит сопроводительный характер



Элемент «Хранилище данных»

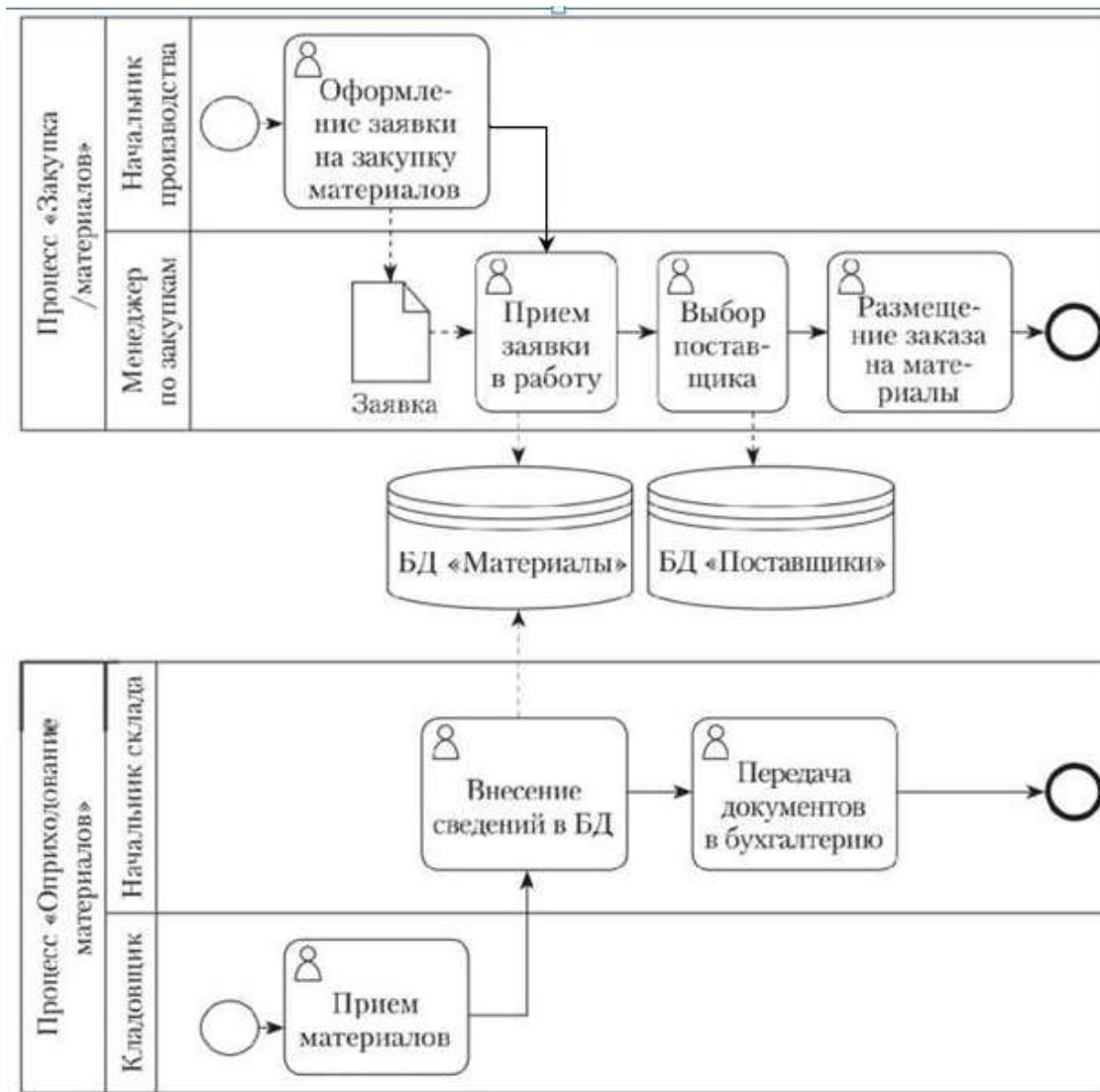


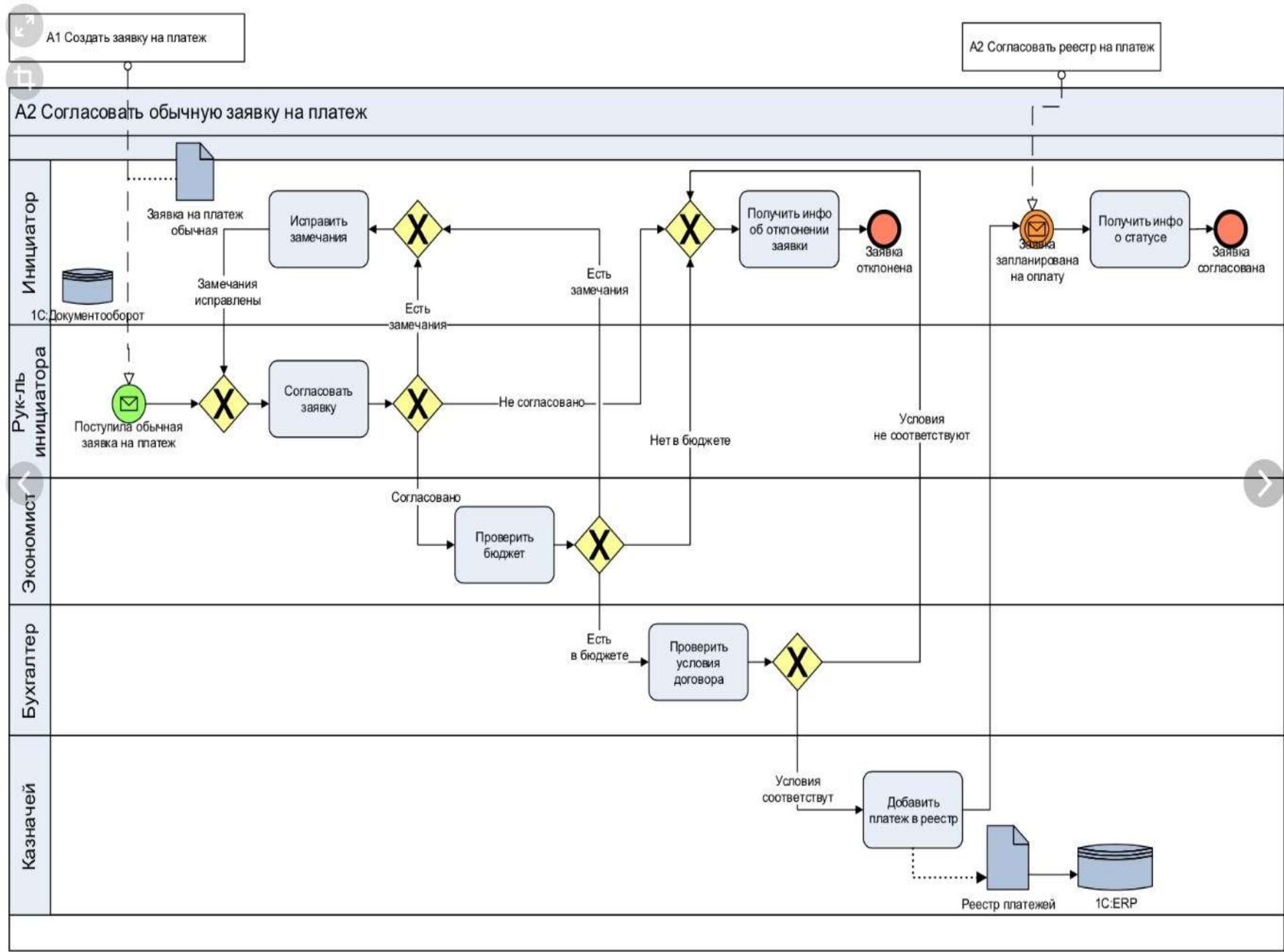
«Хранилище данных» используется для представления на диаграмме бизнес-процесса данных, которые хранятся и используются как в рамках процесса, так и за его пределами, например, другими процессами.

Хранение и актуальность данных и информации, представленных с помощью «хранилища данных» не зависит от того, закончилось выполнение процесса или нет. Они по-прежнему остаются активными и доступными для использования. Этот элемент располагается на схеме за пределами пула



Пример применения «Объекта» и «Хранилища данных»





Подготовить данные

Процесс А

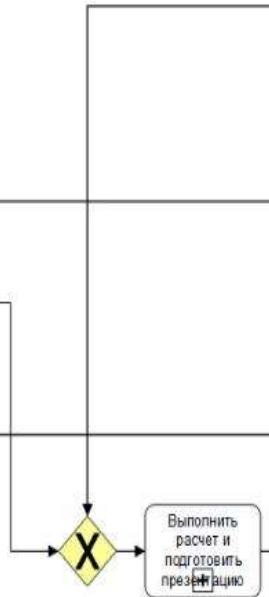
Руководитель



Исполнитель А



Исполнитель Б



А6 Формирование и выполнение задания

