ПРОЕКТИРОВАНИЕ, РАЗРАБОТКА И ЭКСПЛУАТАЦИЯ ИНФОРМАЦИОННЫХ СИСТЕМ

Лекция 4

7 марта 2019 г.



BPMN

BPMN (Business Process Model and Notation) (нотация и модель бизнес-процессов) – это система условных обозначений (нотация) для описания бизнеспроцессов.

Стандартизирована консорциумом **ОМС**, тем же, что и язык **UML**.

	Период	Методологии
1-я волна	1920-1980 rr.	блок-схемы, ориентированные графы, методологии SADT, IDEF, DFD * SADT, Structured Analysis and Design Technique – Методология структурного анализа и проектирования; позднее переименованная в IDEFO.
2-я волна	1990-e rr.	*EPC, Event-driven Process Chain — цепочка процессов, управляемая событиями, разработана в 1992 г. Метод EPC стал частью концепции ARIS (Architecture of Integrated Information Systems — архитектура интегрированных информационных систем).
3-я волна	2000-e ee.	BPMN – Business process model and notation



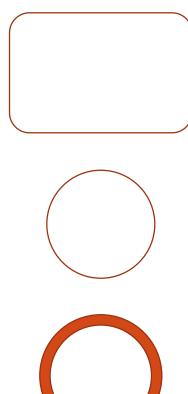
БИЗНЕС-ПРОЦЕСС

Бизнес-процесс – это:

• коллекция взаимосвязанных **активностей**

•выполняемых в ответ на независимое событие

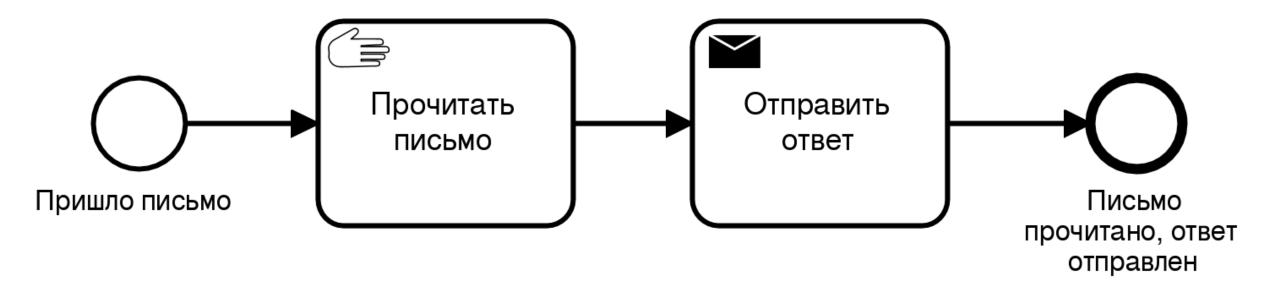
•с целью достижения определенной цели







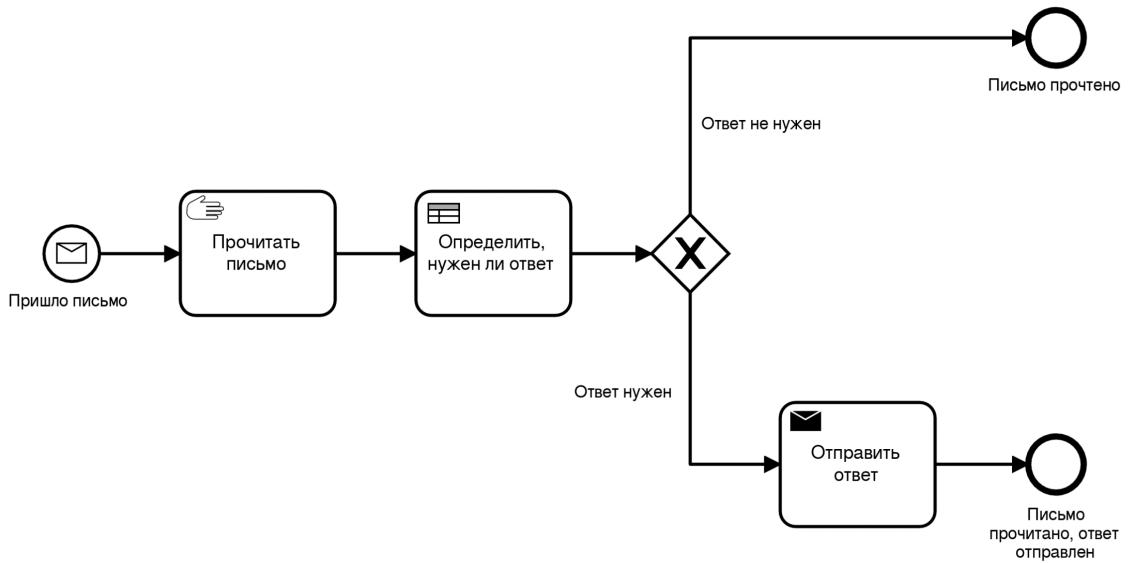
BPMN



Различают понятия модель процесса и экземпляр процесса.



BPMN





ОСНОВНЫЕ ЭЛЕМЕНТЫ

- деятельность (Activity)
- •соединитель потоков (Flow Connector)
- •событие (Event)
- •шлюз или объединение (Gateway)
- дорожка (Swimlane)
- •артефакт (Artifact)



ДЕЯТЕЛЬНОСТЬ

Простая задача

Повторяющаяся задача Параллельные задачи

Простая задача

Закрытый составной процесс

СОЕДИНИТЕЛИ ПОТОКОВ

Графический символ	Название элемента			
	простой поток			
◇ →	условный поток			
→	поток по-умолчанию			
0⊳	поток сообщений			
>	ассоциация			







Графический символ	Название элемента			
	события с триггером «сообщение»			
(a) (b)	события с триггером «таймер»			
	события с триггером «правило»			
® ®	события с триггером «исключительное событие»			
● ●	события с триггером «компенсация»			
⊗ ⊗	события с триггером «отмена»			
	события с триггером «ссылка»			
● ●	события с триггером «составное событие»			

ШЛЮЗ

Граф.	Название элемента
символ	
\Diamond	шлюз
*	шлюз на основе данных процесса с оп. XOR
•	шлюз на основе результатов наступления событий с оп. XOR
\$	шлюз на основе результатов наступления событий с оп. OR
+	шлюз с оп. AND
*	шлюз со сложным условием

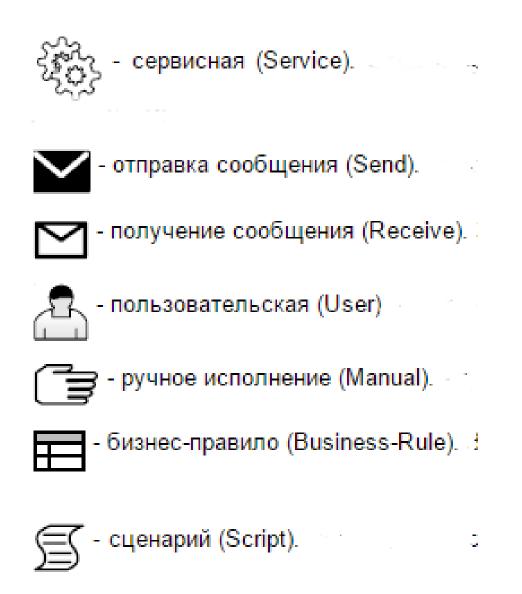
ДОРОЖКИ И ПУЛЫ

НИЯ	Отдел продаж
Компания	Отдел маркетинга

АРТЕФАКТЫ

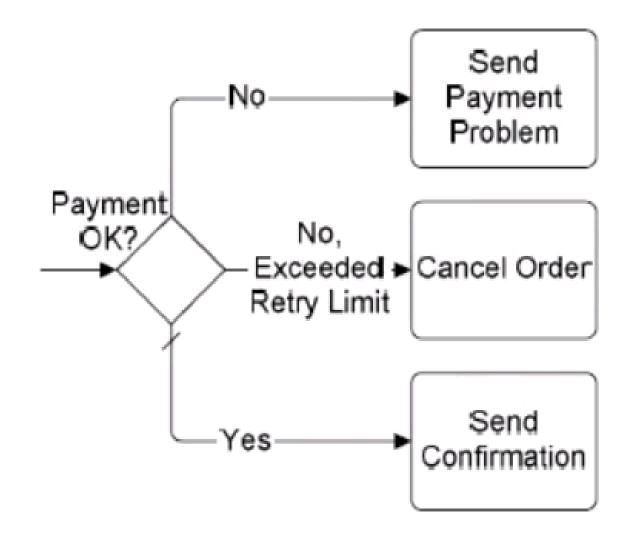
Графический символ	Название элемента			
	данные об объекте			
l J	группа			
аннотация	аннотация			

ТИПЫ ДЕЯТЕЛЬНОСТИ



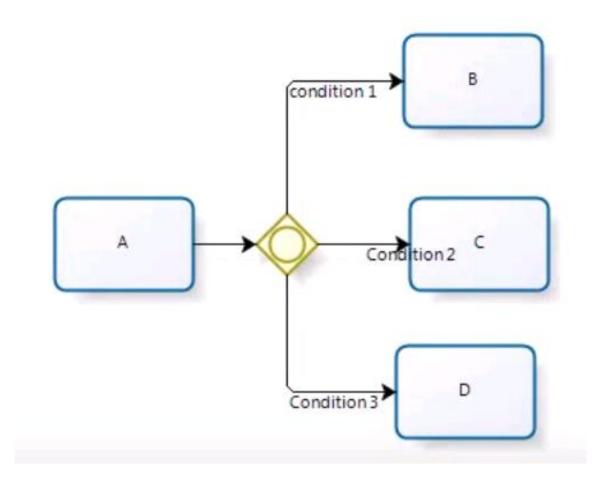
ШЛЮЗ XOR (ИСКЛЮЧАЮЩЕЕ ИЛИ)

- Две или более исходящие альтернативы
- •Толька **одна** выбирается на основе условия, проверенного в **процессе**
- Может быть ветка «по умолчанию»



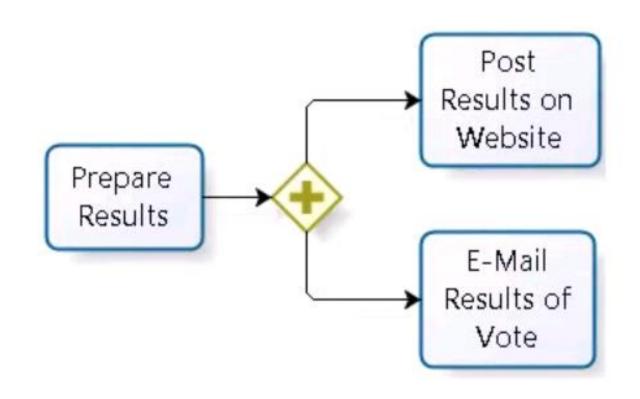
ШЛЮЗ OR (ВКЛЮЧАЮЩЕЕ ИЛИ)

- Две или более исходящие альтернативы
- Может выбираться **любое количество** альтернатив на основе условий



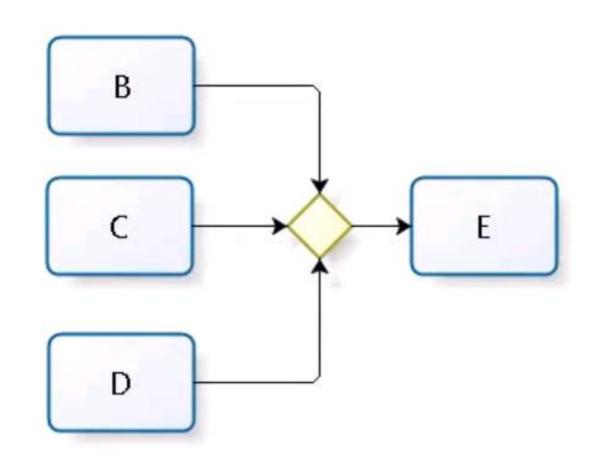
ШЛЮЗ AND (ПАРАЛЛЕЛЬНОЕ ИСПОЛНЕНИЕ)

- Две или более исходящие альтернативы
- •Выбираются **все** альтернативы (условий нет)
- •То же значение, что и несколько исходящих потоков из процесса



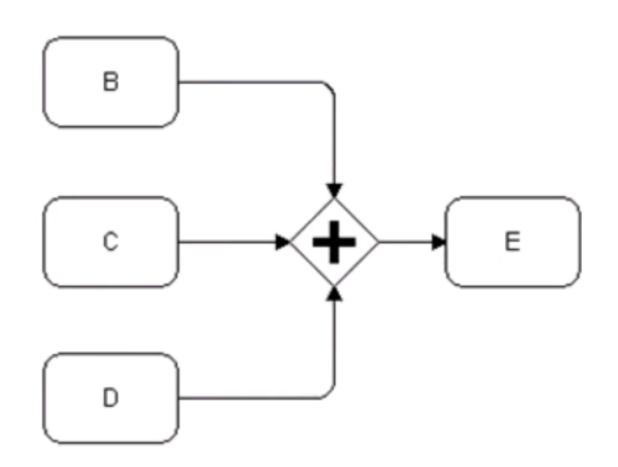
СОЕДИНИТЕЛЬНЫЙ ШЛЮЗ XOR

- •Простое объединение всех входящих путей без какой-либо синхронизации
- Пропускает каждый токен в момент прохождения через шлюз



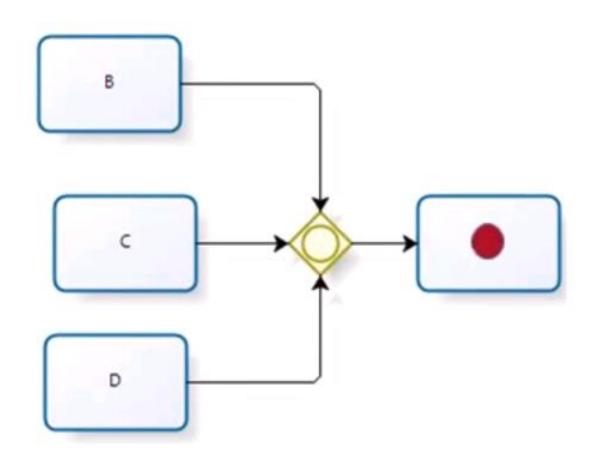
СОЕДИНИТЕЛЬНЫЙ ШЛЮЗ AND

- •Синхронизирующий шлюз
- Ожидает прихода токенов со всех веток и только потом передает один токен дальше

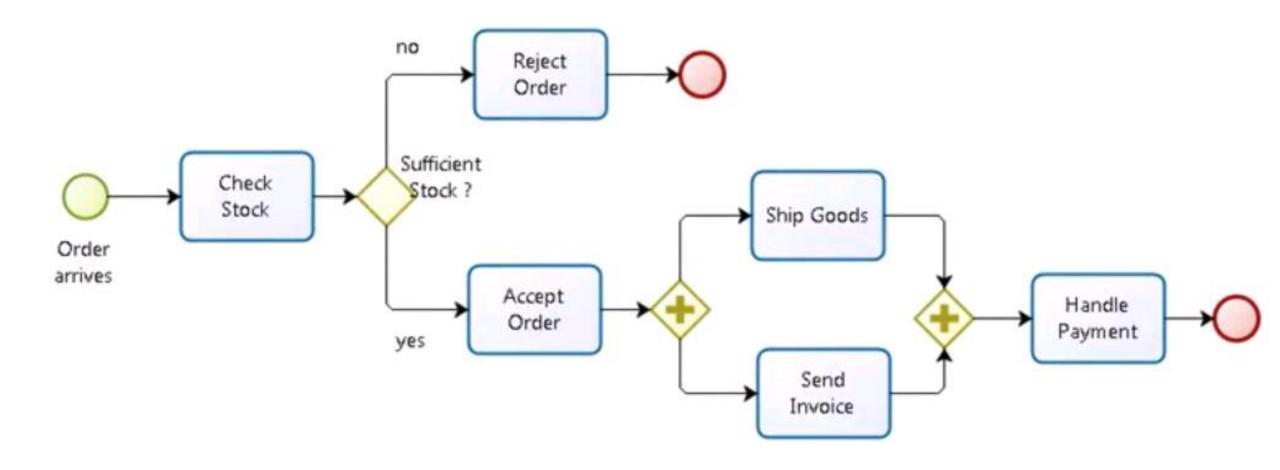


СОЕДИНИТЕЛЬНЫЙ ШЛЮЗ OR

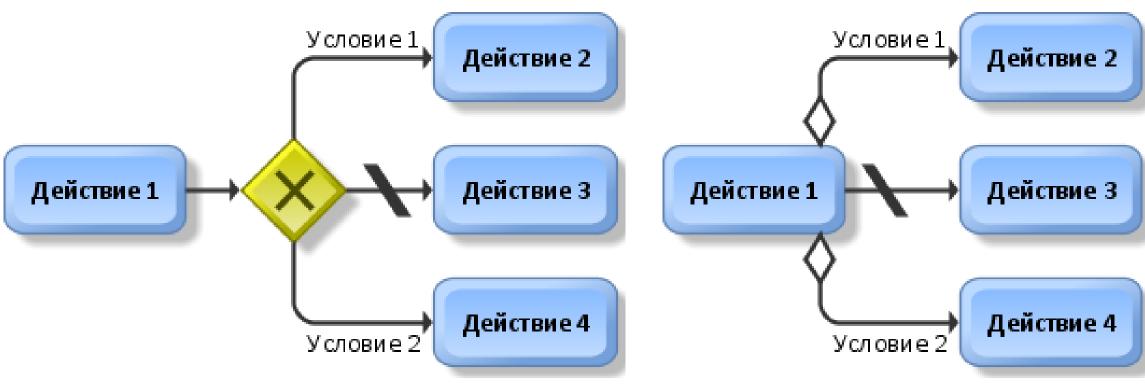
- •Синхронизирующий шлюз
- •Ожидает прихода токенов со всех веток, по которым он идет



ПРИМЕР

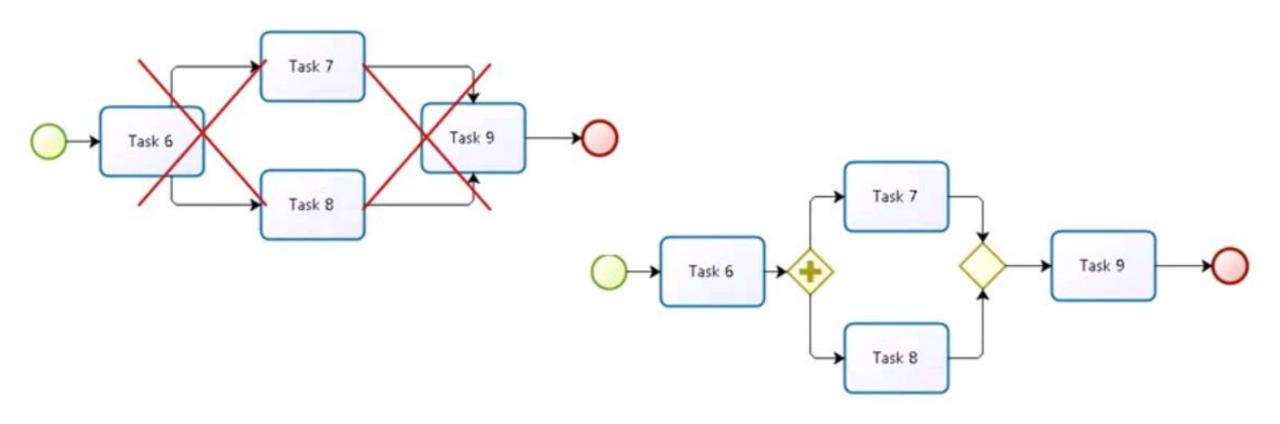


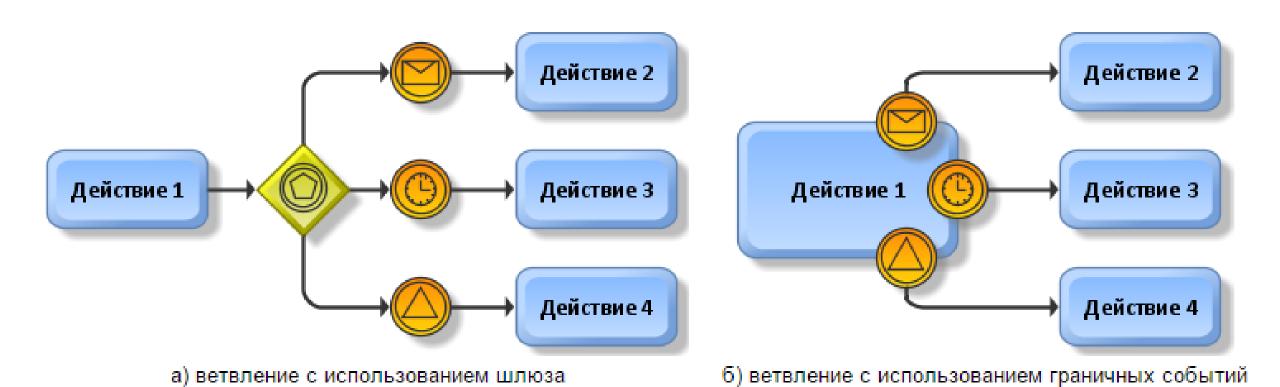
. .

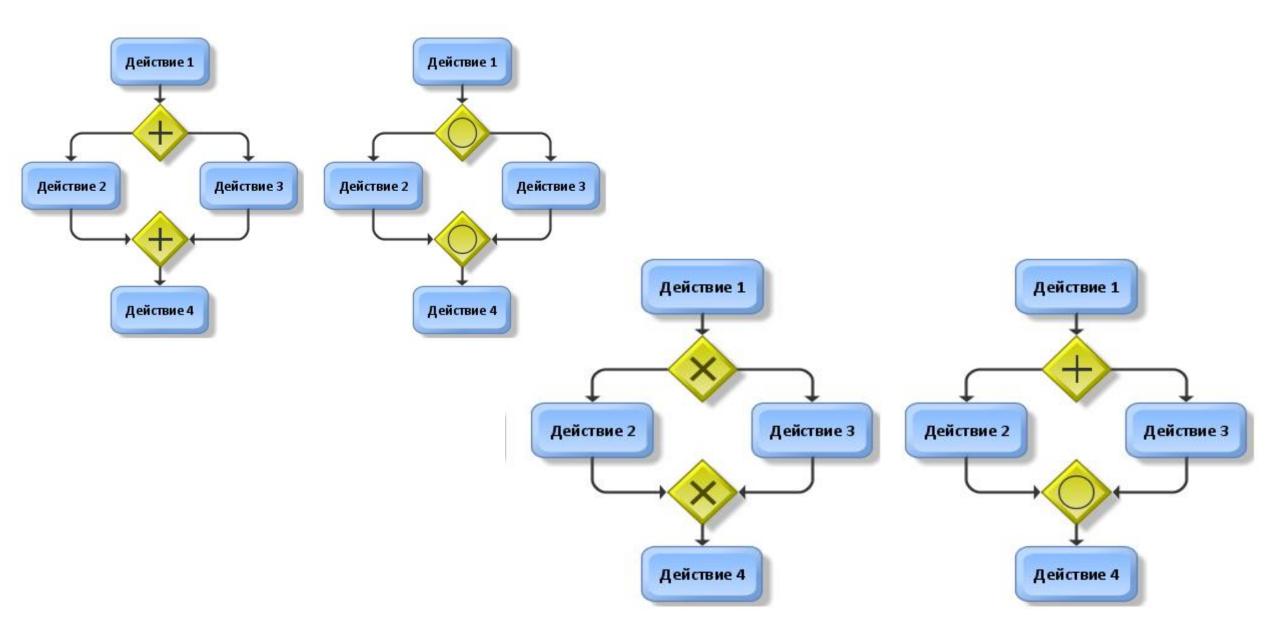


а) ветвление с использованием шлюза

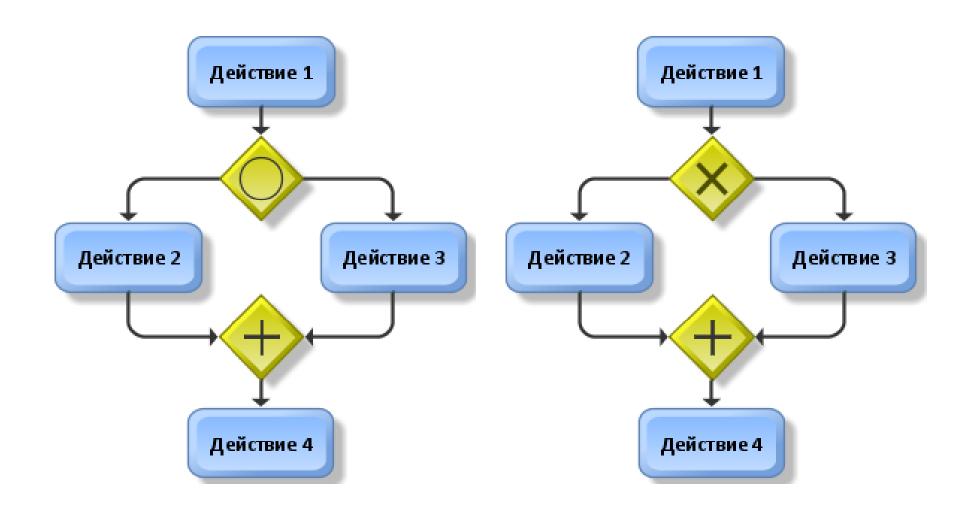
б) ветвление с использованием потоков







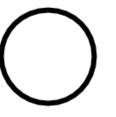
ОШИБКА



События обозначаются кружком:

- •Означают, что «что-то произошло»
- •Влияют на исполнение процесса (старт, остановка, пауза)
- •Процесс может порождать или ловить события
- •Три базовых класса: начальный, промежуточный и конечный
- •Очень много типов

Types	Start			Intermediate				End
	Top- Level	Event Sub-Process Interrupting	Event Sub-Process Non- Interrupting	Catching	Boundary Interrupting	Boundary Non- Interrupting	Throwing	
None	0						0	0
Message			(<u>D</u>)					©
Timer	(3)		(<u>(C)</u>)					
Error		⊗						⊗
Escalation		A	(<u>A</u>)			(<u>(A)</u>))		(A)







Стартовое событие:





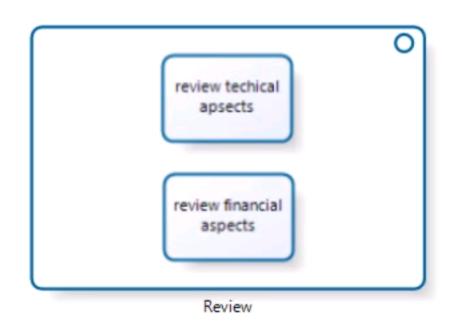
- •Тонкая одинарная окружность
- •Обозначает, где начинается процесс
- •Обычно одно на модель, но может и не быть
- Может бы непрерывающим для подпроцессов
- Может содержать символ триггера внутри то, что послужило причиной события

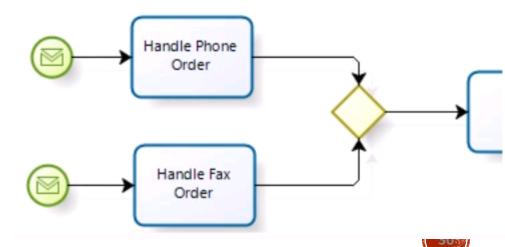
Сколько стартовых событий?

Как правило одно.

Но может и не быть (неявное стартовое событие и шлюз)

Может быть более одного – если начало зависит от того или иного канала возникновения события.













Финальное событие:



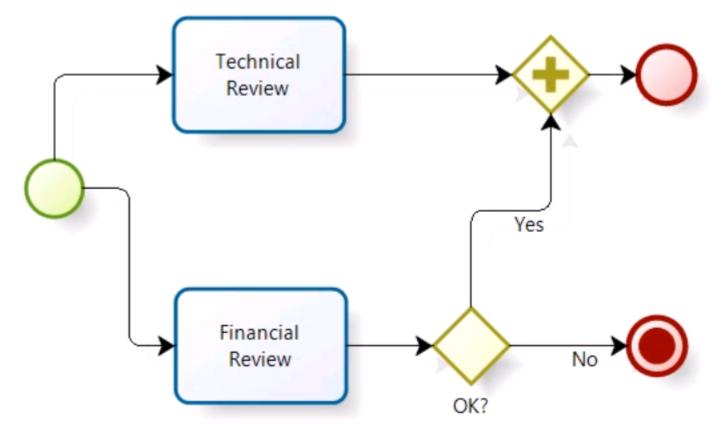




- Толстая одинарная окружность
- Обозначает, где заканчивается процесс
- •Обычно стараются указывать несколько заключительных событий, завершающих ветви процесса
- Все токены должны дойти до финальных событий, чтобы весь процесс завершился корректно
- Некоторые события могут завершить процесс досрочно
- Могут специфицировать результирующий сигнал

Пример события – «терминация» (terminate) – досрочное завершение всего процесса, даже если отдельные ветви

еще выполняются



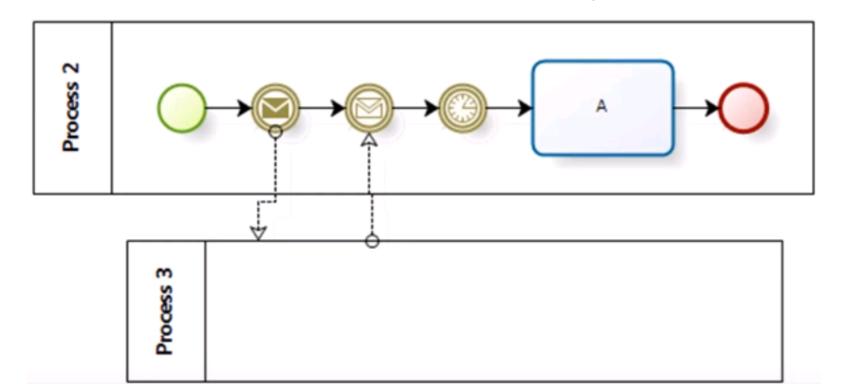
Промежуточное событие:

- •Тонкая двойная окружность
- Семантика зависит от местоположения прикреплено к процессу (пограничное) или стрелке (поточное)
- •Может быть прерывающим (сплошные линии) или непрерывающим (пунктирные)
- Может быть заполненным (порождающее событие) или пустым (слушающим)
 - •Все пограничные события только слушающие



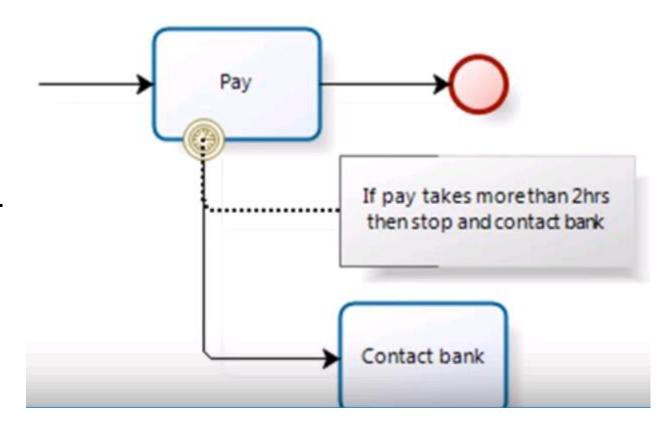
Поточное событие:

- •Генерирующее процесс генерирует сигнал
- •Слушающее процесс ожидает поступления сигнала

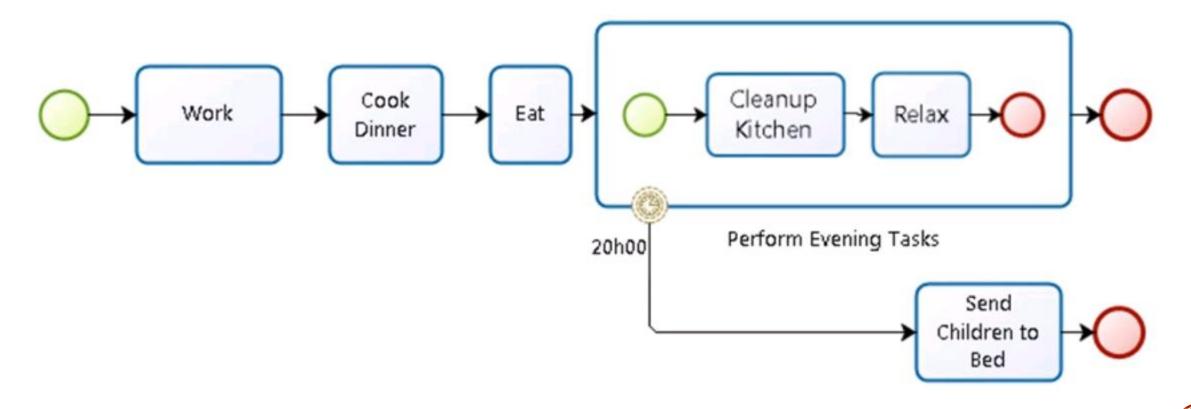


Пограничное прерывающее событие:

- Прерывает событие и запускает альтернативную ветку
- Если событие не наступило процесс продолжается как обычно
- •Процесс задает контекст, в рамках которого слушаются события

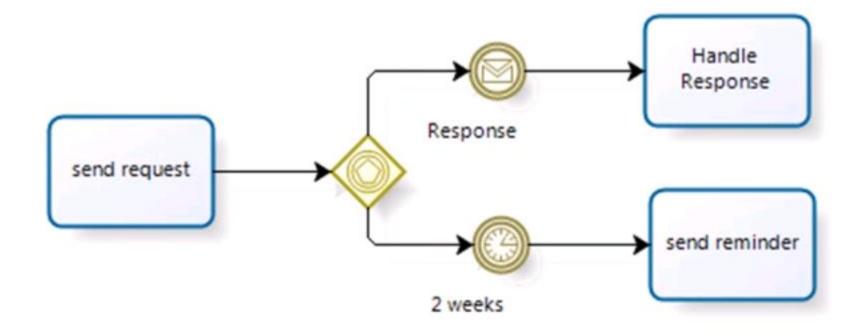


Пример задания контекста



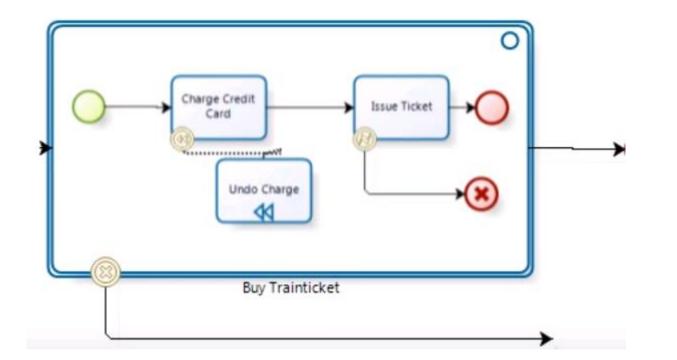
Шлюз **OR**, зависящий от события

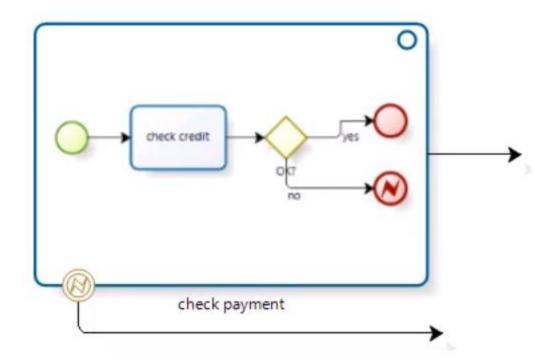
Выбор ветки зависит от того события, которое произошло раньше



Ошибка – немедленно завершает текущий процесс, посылает сообщение родительскому процессу

Эскалация – посылает сигнал родительскому процессу, не завершая текущий





Сигнал – может распространяться внутри и между пулами. Обозначается с помощью подписи.

