

# РАЗРАБОТКА ПОВЕДЕНЧЕСКОЙ МОДЕЛИ

- ▶ - блок-схемы алгоритмов;
- ▶ - EPC-диаграммы;
- ▶ - методология BPMN

# БЛОК-СХЕМЫ АЛГОРИТМОВ

- ▶ ГОСТ 19.701-90 под *схемой* понимается графическое представление определения, анализа или метода решения задачи. С помощью схем можно отобразить как статические, так и динамические аспекты системы. Символы, приведенные в государственном стандарте, могут использоваться в следующих *типах схем*:
  - ▶ схемы данных - определяют последовательность обработки данных и их носители;
  - ▶ схемы программ - отображают последовательность операций в программе (по сути, это и есть блок-схемы алгоритмов в традиционном понимании);
  - ▶ схемы работы системы - отображают управление операциями и потоки данных в системе;
  - ▶ схемы взаимодействия программ - отображают путь активации программ (модулей) и их взаимодействие с соответствующими данными;
  - ▶ схемы ресурсов системы - отображают конфигурацию блоков данных и обрабатывающих блоков.

# Элементы графической нотации

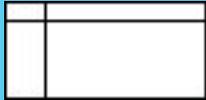
- ▶ символы данных - указывают на наличие данных, вид носителя или способ ввода-вывода данных;
- ▶ символы процесса - указывают операции, которые следует выполнить над данными;
- ▶ символы линий - указывают потоки данных между процессами и/или носителями данных, а также потоки управления между процессами;
- ▶ специальные символы - используются для облегчения написания и чтения схем.

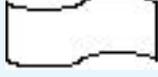
# Условные обозначения на блок- схемах

# СИМВОЛЫ ДАННЫХ. Основные

1.1		Данные	Данные, носитель которых не определен
1.2		Запоминающее устройство (ЗУ)	Данные, хранимые в виде, пригодном для автоматической обработки, носитель не определен

# СИМВОЛЫ ДАННЫХ. Специфические

1.3		ОЗУ	Данные, хранящиеся в ОЗУ (оперативная память)
1.4		ЗУ с последовательным доступом	Данные, хранящиеся на магнитной ленте (магнитная лента, магнитофонная кассета)
1.5		ЗУ с прямым доступом	Данные, хранящиеся на жестких или гибких магнитных дисках, CD, DVD, ZIP и т.д.
1.6		Документ	Данные, представляемые не в компьютерном виде (на бумаге, на пленках и т. д.)

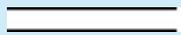
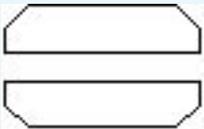
1.7		Ручной ввод	Данные, вводимые вручную с помощью клавиатуры, мыши, светового пера и т. д.
1.8		Карта	Данные на перфокартах, магнитных картах, картах со считываемыми метками и т.д.
1.9		Бумажная лента	Данные на бумажной ленте
1.10		Дисплей	Данные, отображаемые на экране монитора, сигнальные индикаторы и т.д.

# СИМВОЛЫ ПРОЦЕССА. Основной

2.1		Процесс	Элементарная (атомарная) операция по обработке данных (например, $n:=n+1$ )
-----	---	---------	---

# СИМВОЛЫ ПРОЦЕССА. Специфические

2.2		Предопределенный процесс (процедура)	Процесс, состоящий из операций, описанных в другом месте (на другой схеме)
2.3		Ручная операция	Операция, выполняемая вручную
2.4		Подготовка	Подготовительные операции, выполняемые с целью модификации последующих операций (цикл с параметром [For-To-Do])

2.5		Решение	Операция с одним входом и несколькими альтернативными выходами, один из которых активизируется после проверки условия, записываемого внутри символа (операторы If-Then-Else или Case)
2.6		Параллельные действия	Параллельное выполнения двух и более операций
2.7		Границы цикла	Начало и конец цикла. Особенности работы цикла (инициализация, приращение, условие) записывается в начале или конце, в зависимости от того, где осуществляется его проверка (циклы с пред- или постусловием)

# СИМВОЛЫ ЛИНИЙ. Основной

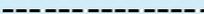
3.1



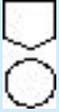
Линия

Поток данных  
или  
управления

# СИМВОЛЫ ЛИНИЙ. Специфические

3.2		Канал связи	Передача данных по каналу связи
3.3		Пунктирная линия	Альтернативная связь между двумя и более символами или для обводки комментируемого участка схемы

# СПЕЦИАЛЬНЫЕ СИМВОЛЫ

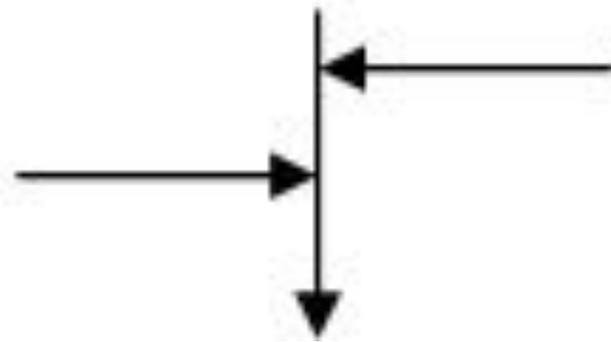
4.1	ГОСТ		Соединитель	Используется для обрыва линий и их продолжения в другом месте. Обычно используется для обозначения взаимосвязанных частей схемы на разных листах. Внутри соединителя пишется номер соединения
	ИСО			
4.2			Терминатор	Выход во внешнюю среду или вход из внешней среды (начало и конец процесса обработки данных [в этом случае внутри пишут "начало" или "конец"], источник или пункт назначения данных, начало и конец работы predetermined процесса)

4.3		Получатель - отправитель	По функциональному назначению аналогичен символу "Терминатор"
4.4		Комментарий	Символ используется для внесения пояснительных записей
4.5		Пропуск	Пропуск символа или группы символов

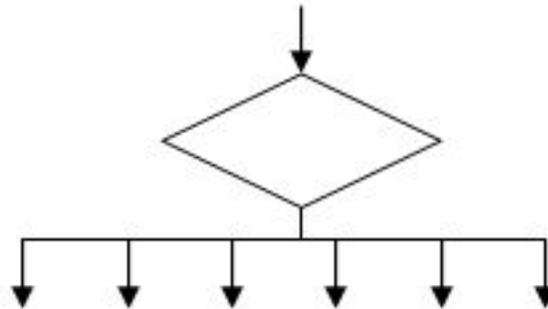
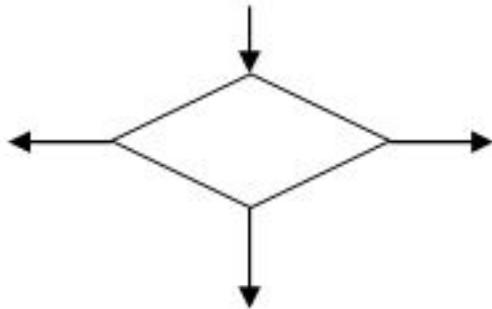
# Правила и рекомендации построения блок-схем

1. Допускается зеркально отображать символы и поворачивать их вокруг оси. В частности, запоминающие устройства с прямым доступом (таблицы на жестком диске) на схемах, как правило, поворачивают на  $90^\circ$  против часовой стрелки.
2. Большинство символов допускают задание внутри них текстовых пояснений. Если текст не помещается внутри символа, то лучше его приводить, используя комментарии.
3. Количество пересечений линий следует минимизировать. При этом считается, что пересекающиеся линии не имеют логической связи друг с другом. Другими словами потоки данных или управления в местах пересечений не меняют своего направления.

- ▶ 4. Если две или более линий объединятся в одну, то место объединения должно быть смещено.

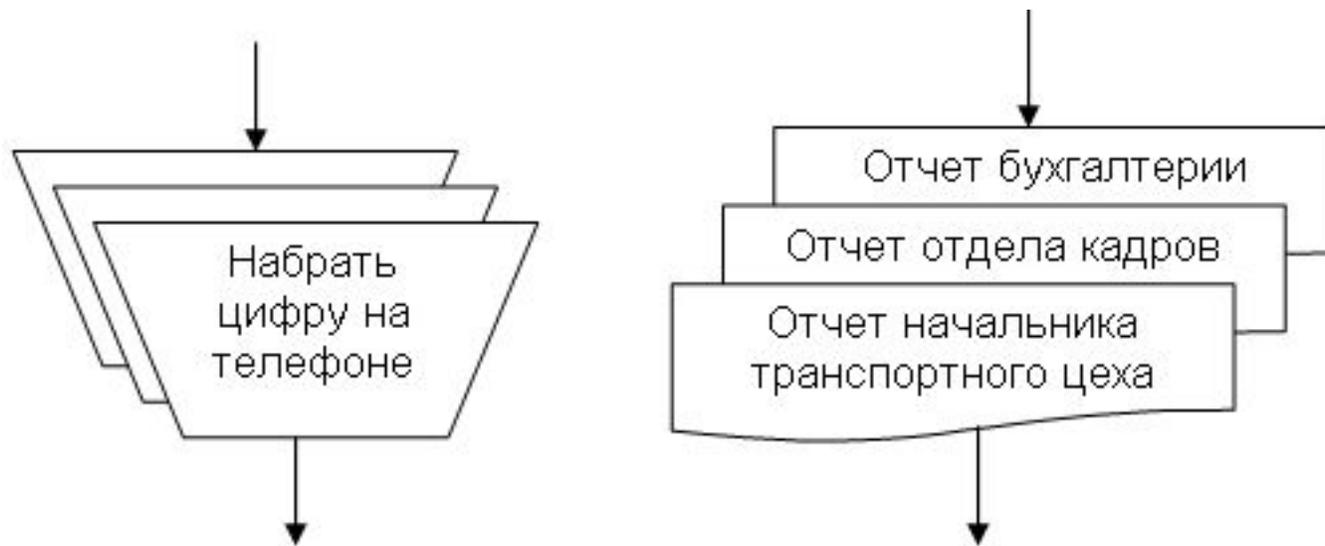


- ▶ 5. Несколько выходов из символа решения следует показывать одним из следующих способов:
  - ▶ несколькими линиями от данного символа к другим символам;
  - ▶ одной линией от данного символа, которая затем разветвляется в соответствующее число потоков.

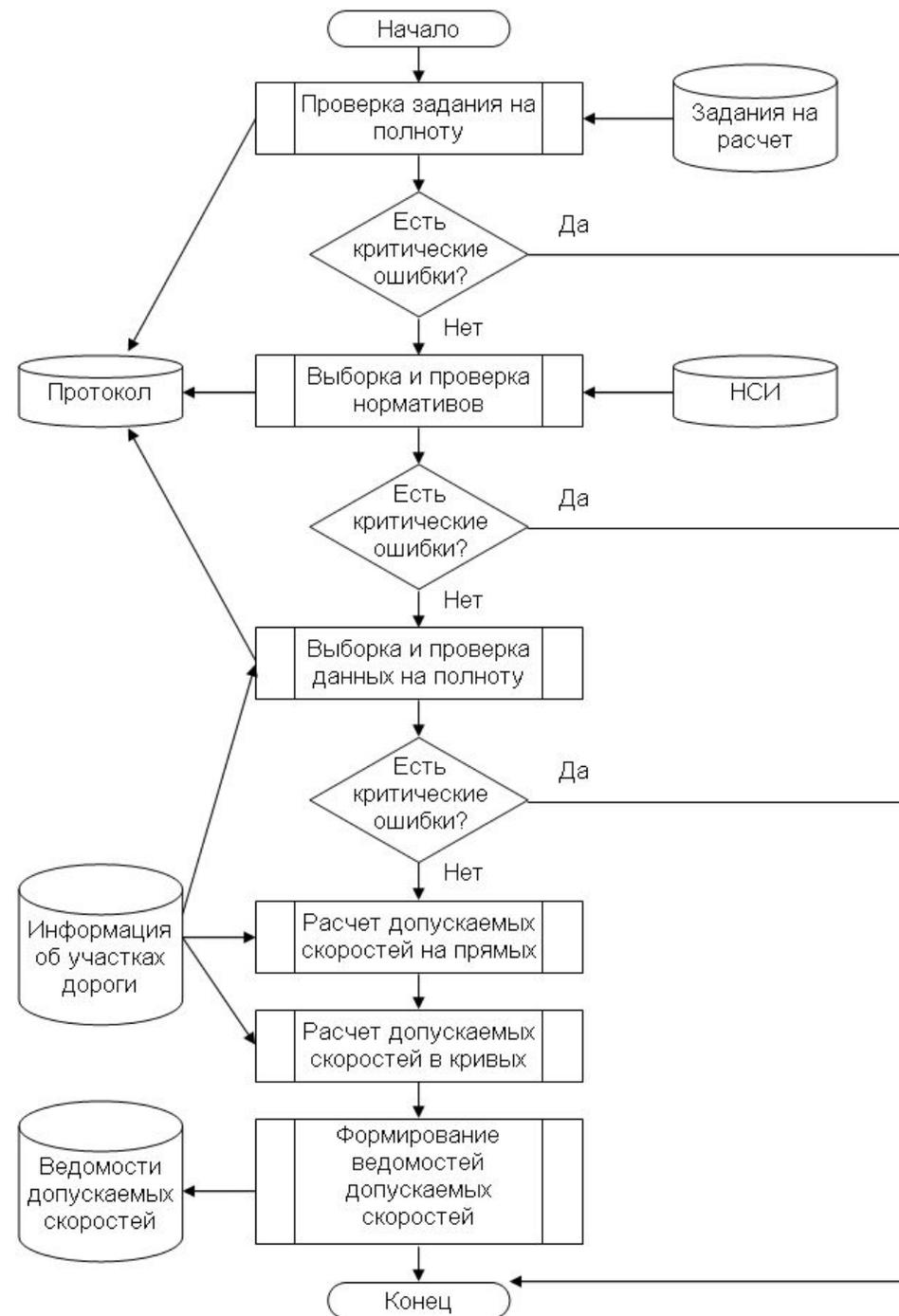


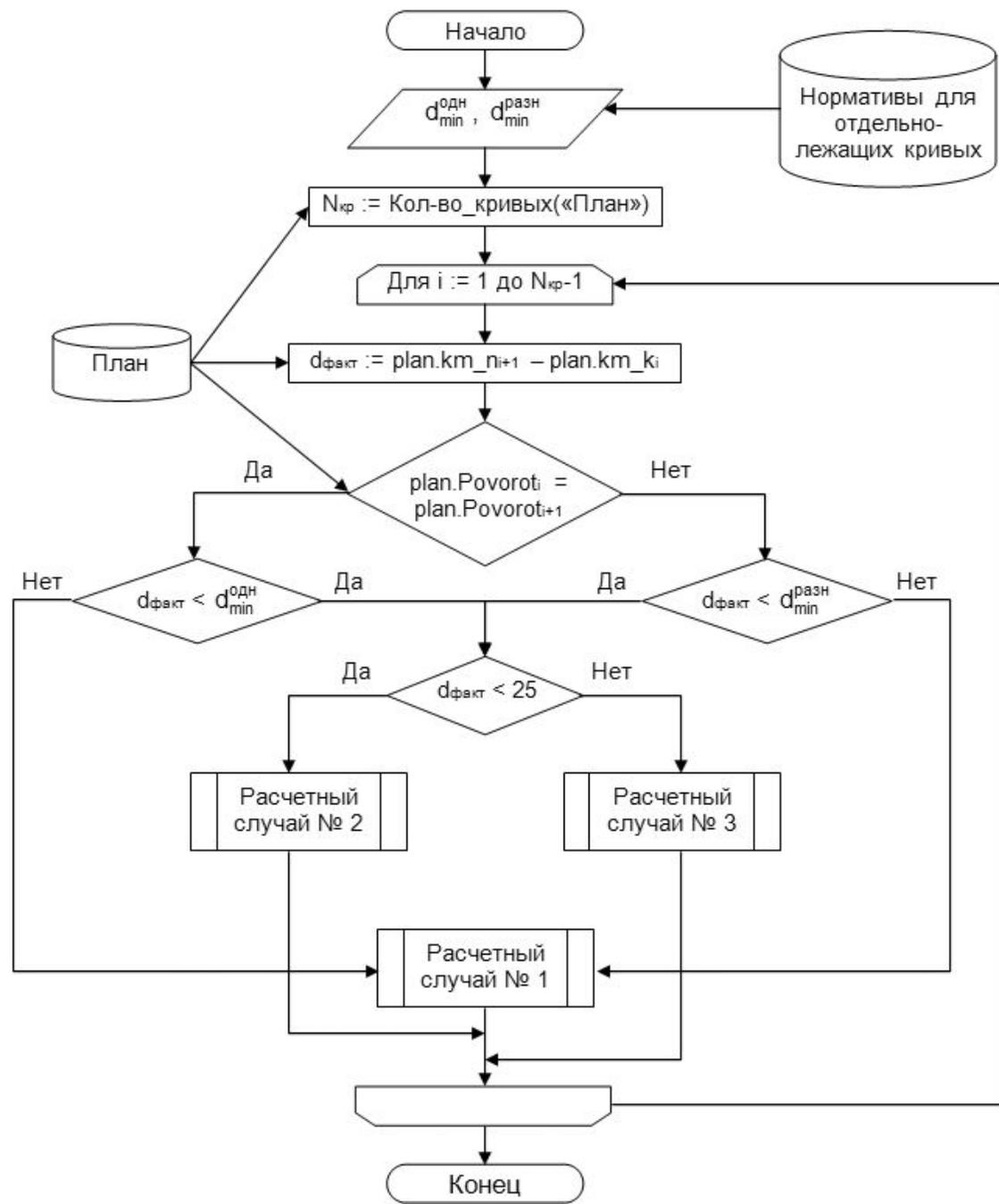
В случае ветвления каждый выход из символа должен сопровождаться либо записью условия ("Сравнить A и B", 3 выхода:  $A > B$ ,  $A < B$  и  $A = B$ ), условие " $A = B$ ", 2 выхода: Да и Нет).

- ▶ 6. Вместо одного символа с соответствующим текстом могут быть использованы несколько символов с перекрытием изображения, каждый из которых может содержать дополнительный текст (например, запись или посылка нескольким получателям, подготовка нескольких копий документа и т. д.).



- ▶ 7. Если направление стрелки не указано, то направление потока считается сверху вниз, слева направо.

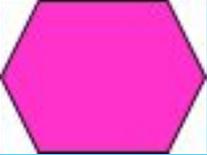




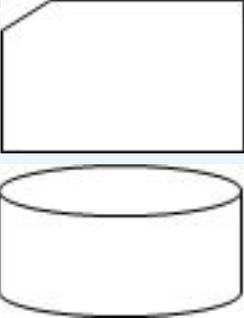
# ЕРС-ДИАГРАММЫ

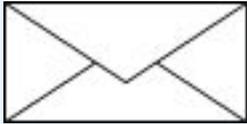
- ▶ **Событийная цепочка процессов (EPC, event-driven process chain)** - тип диаграмм, используемых для моделирования, анализа и реорганизации бизнес-процессов.
- ▶ В тоже время EPC-диаграммы могут использоваться для моделирования поведения отдельных частей системы при реализации функций и служить заменой традиционных блок-схем
- ▶ EPC-метод был разработан **Августом-Вильгельмом Шеером (August-Wilhelm Scheer)** в рамках работ над созданием методологии ARIS (Architecture of Integrated Information Systems - Архитектура интегрированных информационных систем) в начале 1990-х годов.
- ▶ Диаграмма процесса (функции) в нотации EPC представляет собой упорядоченную комбинацию событий и функций. Для каждой функции могут быть определены начальные и конечные события, участники, исполнители, материальные и информационные потоки, сопровождающие её, а также проведена декомпозиция на более низкие уровни.

# СИМВОЛЫ ПРОЦЕССА

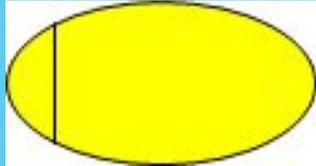
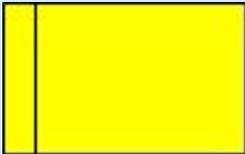
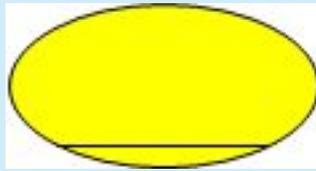
1.1		Событие (Event)	Факт (ситуация, набор условий или обстоятельств), который активизирует или оказывает влияние на дальнейшее развитие одного или более процессов. События инициируют действия или являются их результатами. В отличие от функции, выполнение которой занимает определенный промежуток времени, событие относится к конкретной точке во времени.
1.2		Функция, деятельность (Activity)	Действие или набор действий, выполняемых над объектом (документом, ТМЦ и т.п.) с целью получения заданного результата.
1.3		Интерфейс процесса (Process Interface)	Внешний (по отношению к текущей диаграмме) процесс или функция. Используется для указания взаимосвязи процессов: - обозначает предыдущий или следующий процесс по отношению к текущему процессу (диаграмме); - обозначает процесс, откуда поступил или куда передается объект.

# СИМВОЛЫ ОБЪЕКТОВ

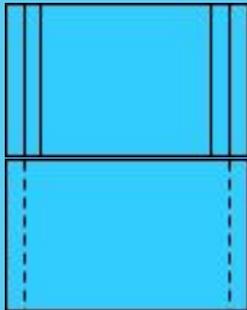
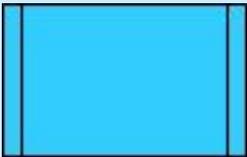
2.1		ТМЦ, информация (Information, Entity)	Товарно-материальные ценности (ТМЦ) или информация, используемые или получаемые в результате действий. Может использоваться вместо элемента «Вход / Выход».
2.2		Документ (Document)	Информация, представляемая не в компьютерном виде (на бумаге, пленках, слайдах).
2.3		Файл, база данных (File, Database)	Информация, представляемая в компьютерном виде (файл, таблица, БД, электронный документ).

2.4		Контекстные данные, кластер (Cluster)	Набор данных, необходимых для выполнения функции (модель, диаграмма, заказ).
2.5		Набор объектов, картотека (CardFile)	Набор ТМЦ или документов.
2.6		Сообщение (Message)	Требование отправителя к получателю на создание ТМЦ, предоставление информации или оказание услуги.
2.7		Вход / Выход, продукт (Input / Output, Product)	Объект, необходимый для выполнения процесса (план работ, заказ, материалы) или являющийся результатом процесса (документация, изделие, выполненная услуга).

# СИМВОЛЫ ИСПОЛНИТЕЛЕЙ

3.1		Организационная единица (Organizational unit)	Структурное подразделение, которому поручено выполнение действия (фирма, организация, отдел, служба).
3.2		Должность, тип исполнителя (Position, Role, Person type)	Должность исполнителя или роль субъекта, которому поручено выполнение действия. Составная часть организационной единицы.
3.3		Исполнитель (Person)	Конкретный исполнитель, которому поручено выполнение действия (имя исполнителя). Экземпляр должности.
3.4		Местоположение (Location)	Местоположение объекта, выполнения действия или возникновения события (фирма, организация, отдел, служба, завод, здание, комната, адрес).

# СИМВОЛЫ ПО

4.1		Приложение, прикладная система (Application)	Информационная система (программный продукт), помощью которой выполняется функция.
4.2		Модуль (Module)	Составная часть информационной системы.

# СИМВОЛЫ ЛИНИЙ

5.1		Поток управления (Control Flow Arrow)	Задаёт последовательность (до- после) возникновения событий и выполнения действий.
5.2		Организационный поток (Organizational Flow Arrow)	Иерархическая связь между однотипными элементами (организационная единица - должность - персона).
5.3		Поток ресурсов (Resources Flow Arrow)	Связь между действием и ресурсами, необходимыми для его выполнения (организационными единицами, персонами, приложениями, модулями и т.п.).

5.4		Информационный поток (Information Flow Arrow)	Связывает действие и элемент, являющийся источником и/или приемником информации (приложение, кластер).
5.5		Поток информационных услуг (Information output Flow Arrow)	Связь между действием и информационным входом/выходом.
5.6		Поток ТМЦ (Material output Flow Arrow)	Связь между действием и материальным входом/выходом.

# ЛОГИЧЕСКИЕ СИМВОЛЫ

6.1		Логическое «И» (AND)	Используется для обозначения слияния и/или ветвления параллельно выполняемых действий и событий. При слиянии выполнение (наступление) следующего действия (события) возможно только при завершении выполнения (наступления) всех предыдущих действий (событий). При ветвлении должны выполняться (наступать) все последующие действия (события).
-----	---	----------------------	--

6.2



Логическое «ИЛИ»  
(OR)

Используется для обозначения слияния и/или ветвления параллельно выполняемых действий и событий. При слиянии выполнение (наступление) следующего действия (события) возможно при завершении выполнения (наступления) хотя бы одного из предыдущих действий (событий). При ветвлении должно выполняться (наступить) хотя бы одно из последующих действий (событий).

6.3



Исключающее «ИЛИ»  
(XOR)

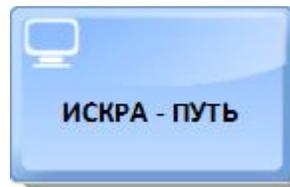
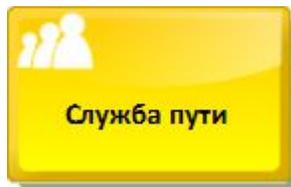
Используется для обозначения слияния и/или ветвления действий и событий. При слиянии выполнение (наступление) следующего действия (события) возможно при завершении выполнения (наступления) только одного из предыдущих действий (событий). При ветвлении должно быть выполнено (наступить) только одно из последующих действий (событий).

# СПЕЦИАЛЬНЫЕ (ДОПОЛНИТЕЛЬНЫЕ) СИМВОЛЫ

7.1		Цель (Objective)	Цель выполнения процесса (высокое качество, низкая себестоимость, сокращение издержек, увеличение прибыли).
7.2		Термин (Term)	Используется: - для обозначения данных, передаваемых между процессами или обрабатываемых при выполнении процессов (Техническое задание, Форма № 1, Ведомость ЦДЛ №3, пинкод); - для обозначения статусов бумажных или электронных документов (подписанный, утвержденный).

# Условные обозначения элементов графической нотации EPC в ARIS

- ▶ а) организационная единица    б) информационная система



# Правила и рекомендации

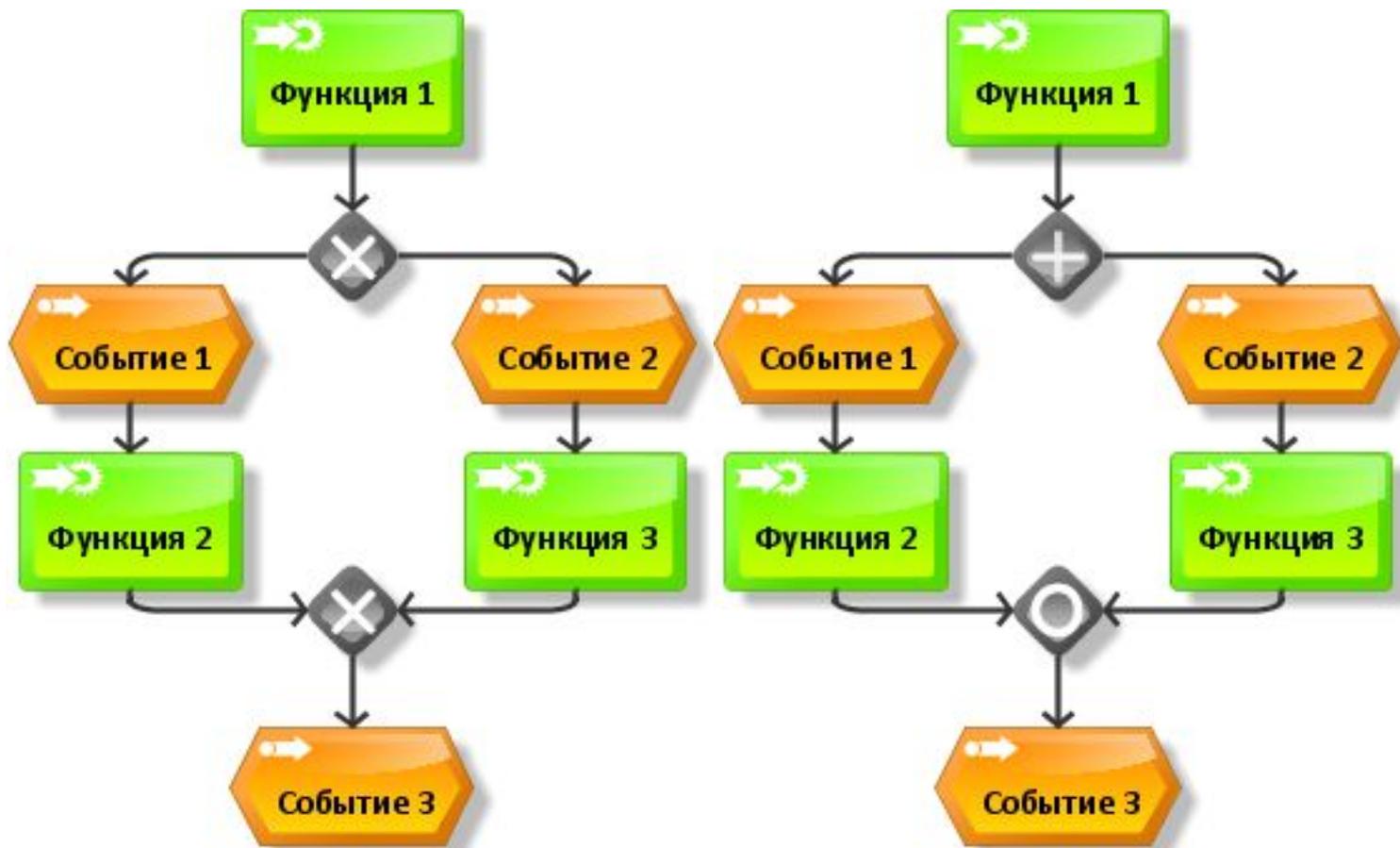
- ▶ 1. Диаграмма функции ЕРС должна начинаться как минимум одним стартовым событием (стартовое событие может следовать за интерфейсом процесса) и завершаться как минимум одним конечным событием (конечное событие может предшествовать интерфейсу процесса).
- ▶ 2. События и функции по ходу выполнения процесса должны чередоваться.
- ▶ 3. Рекомендуемое количество функций на диаграмме - не более 20. Если количество функций диаграммы значительно превышает 20, то существует вероятность, что неправильно выделены процессы на верхнем уровне и необходимо произвести корректировку модели.

- ▶ 4. События и функции должны содержать строго по одной входящей и одной исходящей связи (поток управления), отражающей ход выполнения процесса.
- ▶ 5. На диаграмме не должны присутствовать элементы без единой связи. Исключение может составлять элемент «цель» всего процесса (диаграммы).
- ▶ 6. События и логические операторы, окружавшие функцию на вышележащей (родительской) диаграмме, должны быть начальными/результатирующими событиями и операторами на диаграмме декомпозиции функции.

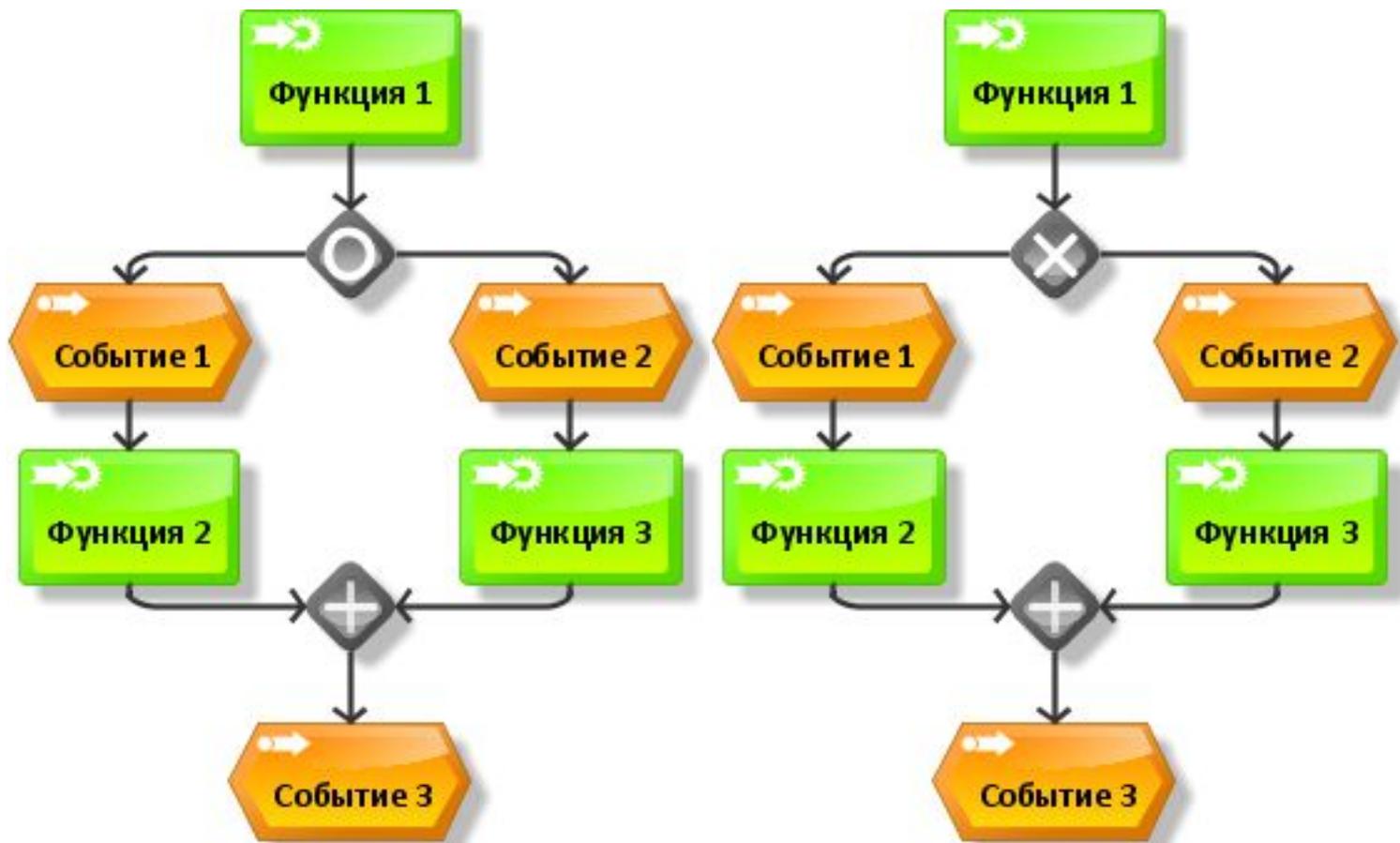
- ▶ 7. Каждый оператор слияния должен обладать минимум двумя входящими связями и только одной исходящей, оператор ветвления - только одной входящей связью и минимум двумя исходящими. Операторы не могут обладать одновременно несколькими входящими и исходящими связями.
- ▶ 8. Логические операторы могут объединять или разветвлять только функции или только события. Одновременное объединение/ветвление функции и события невозможно.
- ▶ 9. Логический оператор, разветвляющий ветви, и оператор, объединяющий эти ветви, должны совпадать. Допускается также ситуация, когда оператор ветвления «И», оператор объединения - «ИЛИ».

# допустимые ситуации

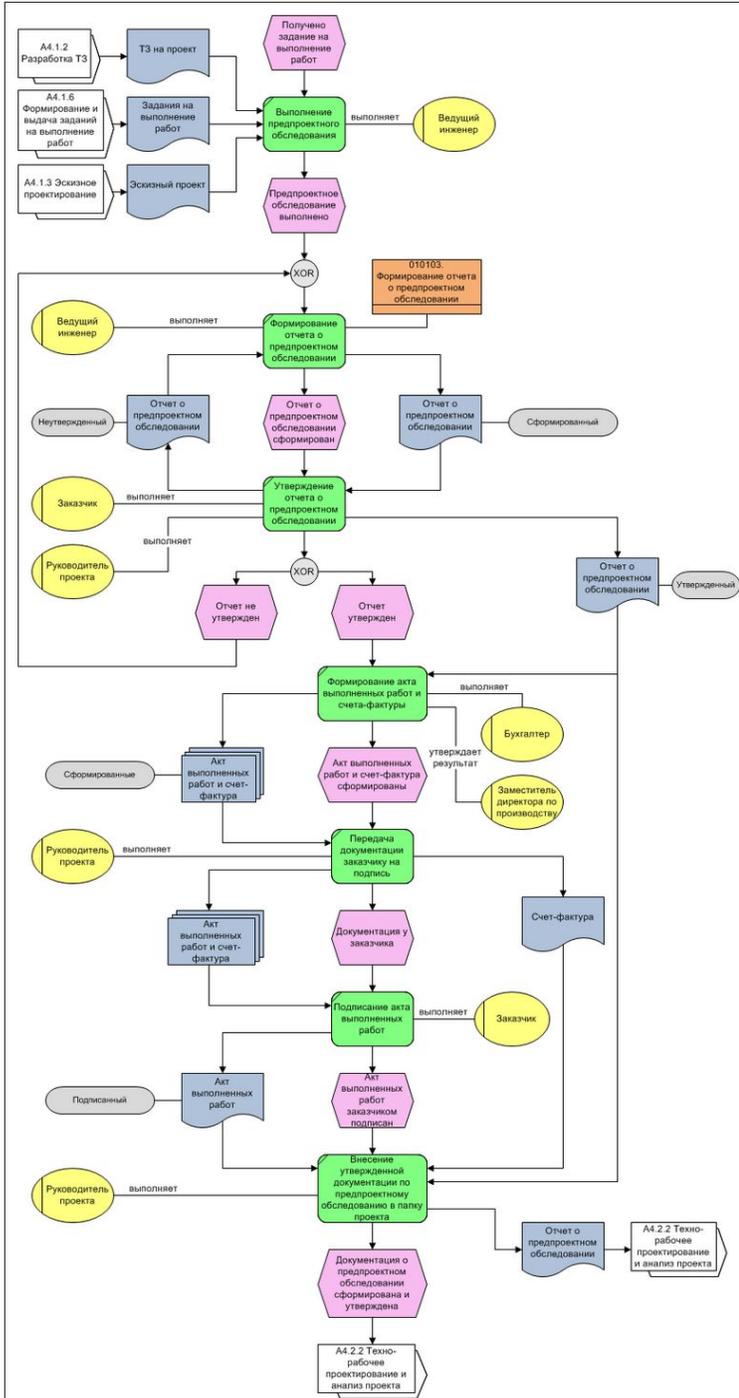




# недопустимые ситуации



- ▶ 10. Количество пересечений линий следует минимизировать. При этом считается, что пересекающиеся линии не имеют логической связи друг с другом. Другими словами, потоки в местах пересечений не меняют своего направления.



# МЕТОДОЛОГИЯ ВРМН

- ▶ **Модель и нотация бизнес-процессов (BPMN, Business Process Model and Notation)** - методология моделирования, анализа и реорганизации бизнес-процессов. Разработана Business Process Management Initiative (BPMI), с 2005 г. поддерживается и развивается **Object Management Group (OMG)**. В отличие от других методологий бизнес-моделирования, имеющих статус «фирменного» (EPC) или «национального» (IDEF0) стандарта, BPMN получила «международный» статус - Международная организация по стандартизации опубликовала стандарт «ISO/IEC 19510:2013. Information technology - Object Management Group. Business Process Model and Notation».
- ▶ Основной целью BPMN является обеспечение доступной нотацией описания бизнес-процессов всех пользователей: от аналитиков, создающих схемы процессов, и разработчиков, ответственных за внедрение технологий выполнения бизнес-процессов, до руководителей и обычных пользователей, управляющих этими бизнес-процессами и отслеживающих их выполнение. Таким образом, BPMN нацелен на устранение расхождения между моделями бизнес-процессов и их реализацией.

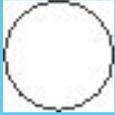
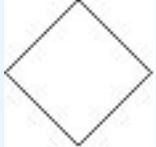
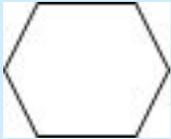
# Идеи

- ▶ - UML (Unified Modeling Language, Унифицированный язык моделирования):
  - ▶ о Activity Diagram (диаграмма деятельности);
  - ▶ о EDOC (Enterprise Distributed Object Computing, корпоративная распределенная обработка объектов) - Business Processes (бизнес-процессы);
- ▶ - IDEF (SADT);
- ▶ - ebXML (Electronic Business eXtensible Markup Language, расширяемый язык разметки для электронного бизнеса) BPSS (Business Process Specification Schema, схемы спецификации бизнес-процессов);
- ▶ - ADF (Activity-Decision Flow, поток «деятельность-результат») Diagram;
- ▶ - RosettaNet;
- ▶ - LOVEM (Line of Visibility Engineering Methodology, визуальная методология проектирования);
- ▶ - EPC.

# Элементы (символы) графической нотации BPMN по назначению объединены в категории

- ▶ - объекты потока (Flow Objects);
- ▶ - данные (Data);
- ▶ - зоны ответственности (Swimlanes);
- ▶ - соединяющие элементы (Connecting Objects);
- ▶ - артефакты (Artifacts).
- ▶

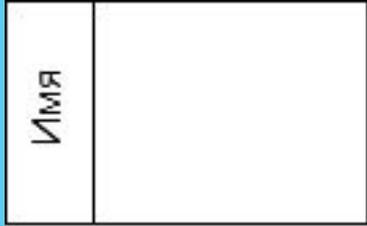
# СИМВОЛЫ ОБЪЕКТОВ ПОТОКА

1.1		Событие (Event)	Факт (ситуация, набор условий или обстоятельств), который активирует или оказывает влияние на дальнейшее развитие одного или более процессов. Событие инициирует действия или является их результатами. В отличие от функции, выполнение которой занимает определенный промежуток времени, событие относится к конкретной точке во времени.
1.2		Действие, деятельность (Activity)	Действие или набор действий, выполняемых исполнителем в ходе процесса. Помимо наименования действия вверху и внизу символа могут указываться имена участников.
1.3		Шлюз, логический оператор (Gateway)	Используется для обозначения слияния и/или ветвления потока событий и действий.
1.4		Обмен сообщениями (Conversation)	Описание действия, характеризующего обмен информацией между участниками (пулами) взаимодействия.

# СИМВОЛЫ ДАННЫХ

2.1		Объект данных (Data Objects)	Товарно-материальные ценности (ТМЦ) или информация, используемые или получаемые в результате действий.
2.2		Хранилища данных (Data Stores)	База данных или ее фрагмент, содержащий информацию для выполнения действий.
2.3		Сообщение (Message)	Отражает факт передачи информации между участниками процесса.

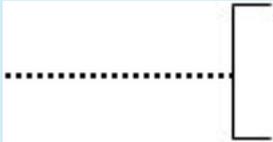
# СИМВОЛЫ ЗОНЫ ОТВЕТСТВЕННОСТИ

3.1		Пул, участник (Pool, Participant)	Структурное подразделение, которому поручено выполнение действия (фирма, организация, отдел, служба).
3.2		Дорожка (Lane)	Должность исполнителя или роль субъекта, которому поручено выполнение действия. Составная часть организационной единицы.

# СИМВОЛЫ СОЕДИНЯЮЩИХ ЭЛЕМЕНТОВ (ЛИНИЙ)

4.1		Поток операций, поток управления (Sequence Flow)	Задаёт последовательность (до-после) возникновения событий и выполнения действий.
4.2		Поток сообщений (Message Flow)	Отражает информационный обмен между участниками процесса. Обычно соединяет действия и/или пулы двух участников процесса.
4.3	 	Ассоциация (Association)	Отражает связь между данными (артефактами) и объектами потока.
4.4		Ссылка на обмен сообщениями (Conversation Link)	Указывает на обмен сообщениями между участниками взаимодействия.

# СИМВОЛЫ АРТЕФАКТОВ (СПЕЦИАЛЬНЫЕ СИМВОЛЫ)

5.1		Группа (Group)	Используется для группировки графических элементов, принадлежащих одной и той же категории.
5.2		Комментарий, текстовая аннотация (Text Annotation)	Примечание (дополнительная информация), связанная с отображенным элементом.

# События по времени наступления

- ▶ **стартовое событие** - инициирует начало процесса (диаграммы). Из стартового события поток управления может только исходить, а поток сообщений - как входить, так и исходить. На диаграмме процесса, как правило, отображается только одно стартовое событие, но оно может отсутствовать или их может быть несколько при отображении процесса с пулами, дорожками или развернутыми подпроцессами. Контур события отображается одинарной тонкой линией;
- ▶ **конечное событие** - является результатом выполнения процесса. В конечное событие поток управления может только входить, а поток сообщений - как входить, так и исходить. В конечное событие может только входить поток (стрелка). На диаграмме конечное событие, как и стартовое, может быть одно, несколько (даже при отсутствии пулов и дорожек) или ни одного. Контур события отображается одинарной жирной линией;
- ▶ **промежуточное событие** - все остальные события, возникающие в ходе выполнения процесса. В промежуточное событие обязательно должен входить и выходить один поток. Исключение составляет граничные (Boundary) события, возникающие и обрабатываемые непосредственно либо в самом начале действия либо в его конце. Такие события отображаются на границе (контуре) действия и у них может быть только либо входящий либо исходящий поток. Контур события отображается двойной тонкой линией;

# ПО ВОЗМОЖНОСТИ прерывания выполнения действия (подпроцесса)

- ▶ **непрерывающее событие** - стартовое или промежуточное событие, возникающее в ходе выполнения действия, но инициирующее связанный с событием исходящий поток только после завершения действия. Контур события отображается штриховой линией;
- ▶ **прерывающее событие** - событие, возникающее до или после стандартного выполнения действия или требующее его немедленного прекращения в исключительных ситуациях. Например, при отсутствии всей необходимой информации или возникновении ошибки в ходе ее обработки, необходимости выполнения дополнительных действий и т.д. Контур события отображается сплошной линией;

# по типу результата действия

- ▶ **событие-инициатор обработки** - стартовое или промежуточное событие, возникшее в результате выполнения действия и требующее его последующей обработки. Отображается незакрашенной иконкой;
- ▶ **событие-результат обработки** - промежуточное или конечное событие, возникшее в результате выполнения действия и являющееся итоговым результатом стандартного или нестандартного выполнения процесса. Отображается закрашенной иконкой;

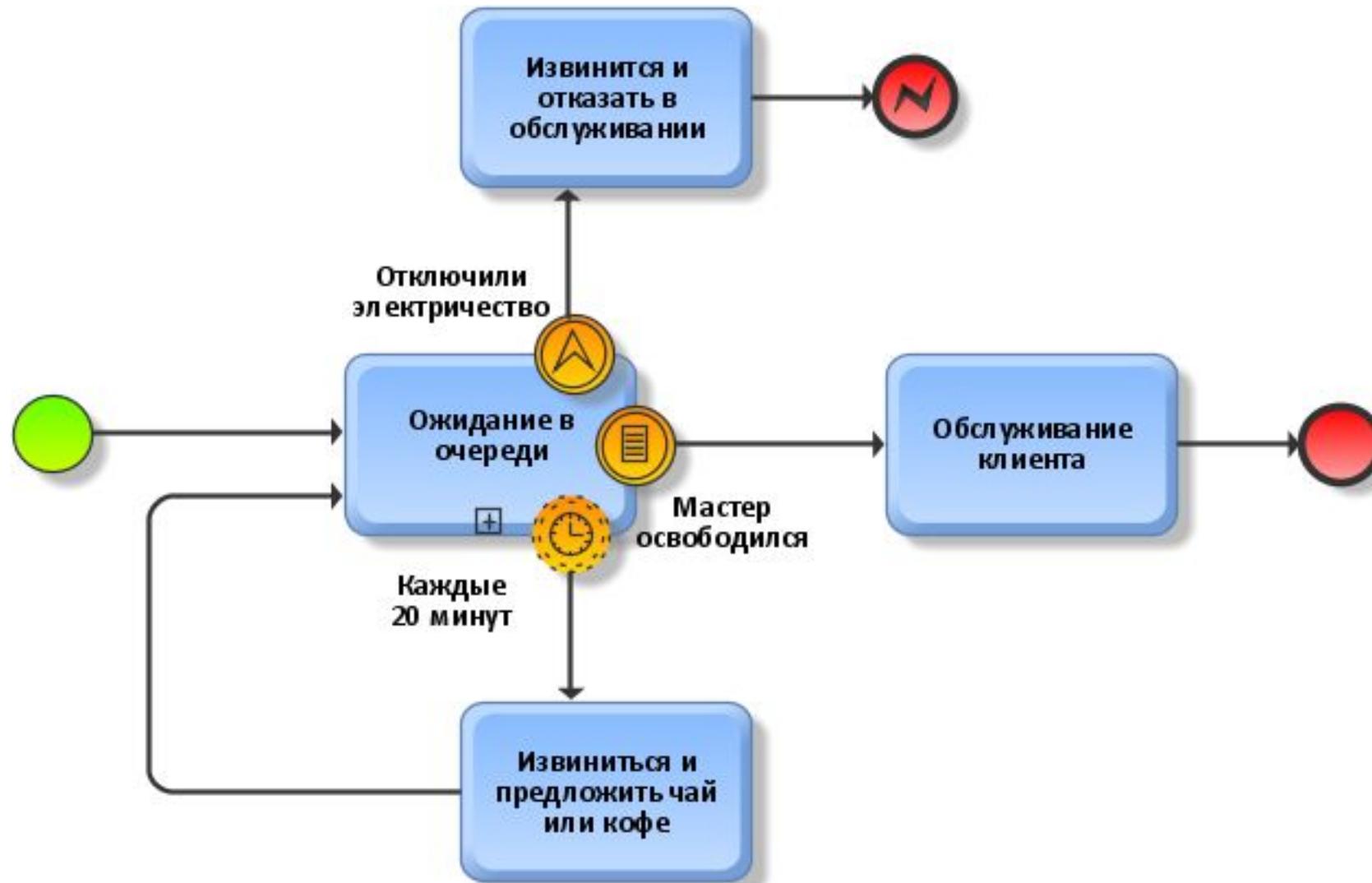
# по причине возникновения (триггеру)

▶ См далее 😊

# Пример использование различных типов событий по времени возникновения и результату действия



Пример использование различных типов событий по возможности прерывания выполнения действия



# Действие

- ▶ задача (Task) - элементарное (неделимое, атомарное) действие. Специфика (разновидность) задачи может быть отображена иконкой (маркером) в левом верхнем углу символа действия
- ▶ подпроцесс (Sub-Process) - составное действие, включающее в себя другие действия, шлюзы, события и потоки операций. Части подпроцесса могут непосредственно отображены на диаграмме внутри символа действия или вынесены на отдельную диаграмму декомпозиции. Во втором случае на родительской диаграмме в центре нижнего края действия (подпроцесса) отображается символ + .
- ▶ вызов (Call). Позволяет включать в состав диаграммы повторно используемые задачи и подпроцессы. На диаграмме выделяется жирным контуром.

# Задача

	сервисная (Service). Задача предназначена для оказания услуги, которая может являться как веб-сервисом, так и автоматизированным приложением;
	отправка сообщения (Send). Задача считается выполненной, если сообщение послано хотя бы один раз;
	получение сообщения (Receive). Задача считается выполненной, если сообщение получено хотя бы один раз;
	пользовательская (User). Характерная задача, выполняемая исполнителем при содействии других людей или программного обеспечения;
	ручное исполнение (Manual). Характерная задача, выполняемая исполнителем без каких-либо средств автоматизации;
	бизнес-правило (Business-Rule). Задача, технология выполнения которой зависит от текущих обстоятельств и выбирается на основе заданного бизнес-правила;
	сценарий (Script). Задача, порядок выполнения операций которой описан на языке, распознаваемом исполнителем. Обычно используется для задач, выполняемых автоматическими средствами;

# Подпроцесс



событийный подпроцесс (Event Sub-Process). Запускается каждый раз, когда происходит одно из стартовых событий. На диаграмме событийный подпроцесс не связан с другими действиями потоками операций. Контур подпроцесса отображается точками;



транзакция (Transaction). Действие, состоящее из составных операций, удачное завершение (получение конкретного положительного результата) которого возможно при удачном завершении всех его составляющих. В случае возникновения проблем при выполнении подпроцесса (невозможности выполнения одной из операций или высокой вероятности ее некорректного выполнения) результаты предыдущих операций отменяются (событие отмена) или компенсируются (событие компенсация). Контур подпроцесса отображается двойной сплошной линией;

# ВЫЗОВ



вызов (Call). Позволяет включать в состав диаграммы повторно используемые задачи и подпроцессы. На диаграмме выделяется жирным контуром.

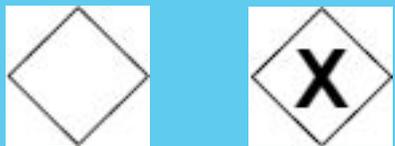
# Дополнительные особенности реализации или выполнения

	цикл (Loop). Действие выполняется в цикле с пред- (while) или пост- (repeat-until) условием;
или ≡	многоэкземплядность (Multi-Instance). Параллельное или последовательное выполнение нескольких экземпляров однотипных действий. При последовательном выполнении действие можно рассматривать как цикл с параметром (for);
	компенсация (Compensation). Действие выполняется взамен стандартного при невозможности его удачного завершения;
~	настраиваемый подпроцесс (Ad-Hoc). Указывается только для подпроцессов. Конкретный состав и последовательность входящих в него действий определяется исполнителем в процессе его выполнения.

**Дополнительные особенности** реализации или выполнения действия могут быть указаны с помощью маркеров, отображаемых у нижнего края символа

# Шлюз

Шлюз предназначен для указания специфики пропуска потока операций по альтернативным или параллельным ветвям. Шлюз может не иметь входящих или исходящих потоков, но должен иметь, как минимум, два и более либо входящих либо исходящих потока. Тип шлюза задается маркером, указываемым внутри его символа



эксклюзивный (Exclusive, XOR - исключающее ИЛИ). Предназначен для разделения потока операций на несколько альтернативных маршрутов, т.е. в ходе выполнения процесса может быть активирован только один из предложенных маршрутов. Условия пропуска по исходящему маршруту задается рядом с соответствующей линией в виде логического выражения;



неэксклюзивный (Inclusive, OR - логическое ИЛИ). Предназначен для разделения потока операций на несколько маршрутов, каждый из которых активируется при условии истинности связанного с ним логического выражения. Таким образом, при выполнении процесса может быть выбрано сразу несколько маршрутов, в т.ч. и ни одного в случае ложности всех выражений;



комплексный (Complex). Аналогичен неэксклюзивному шлюзу. Отличие заключается в том, что с ним связано одно выражение, которое определяет, какие из потоков операций будут активированы;



параллельный (Parallel, AND - логическое И). Предназначен для слияния/ветвления одновременно (параллельно) выполняемых потоков операций;



экслюзивный, основанный на событиях (Exclusive Event-Based). Предназначен для разделения потока операций на несколько альтернативных маршрутов. Единственный маршрут, по которому будет продолжен процесс, выбирается не на основе логического выражения, а в зависимости от произошедших событий, которые указываются по соответствующему маршруту;

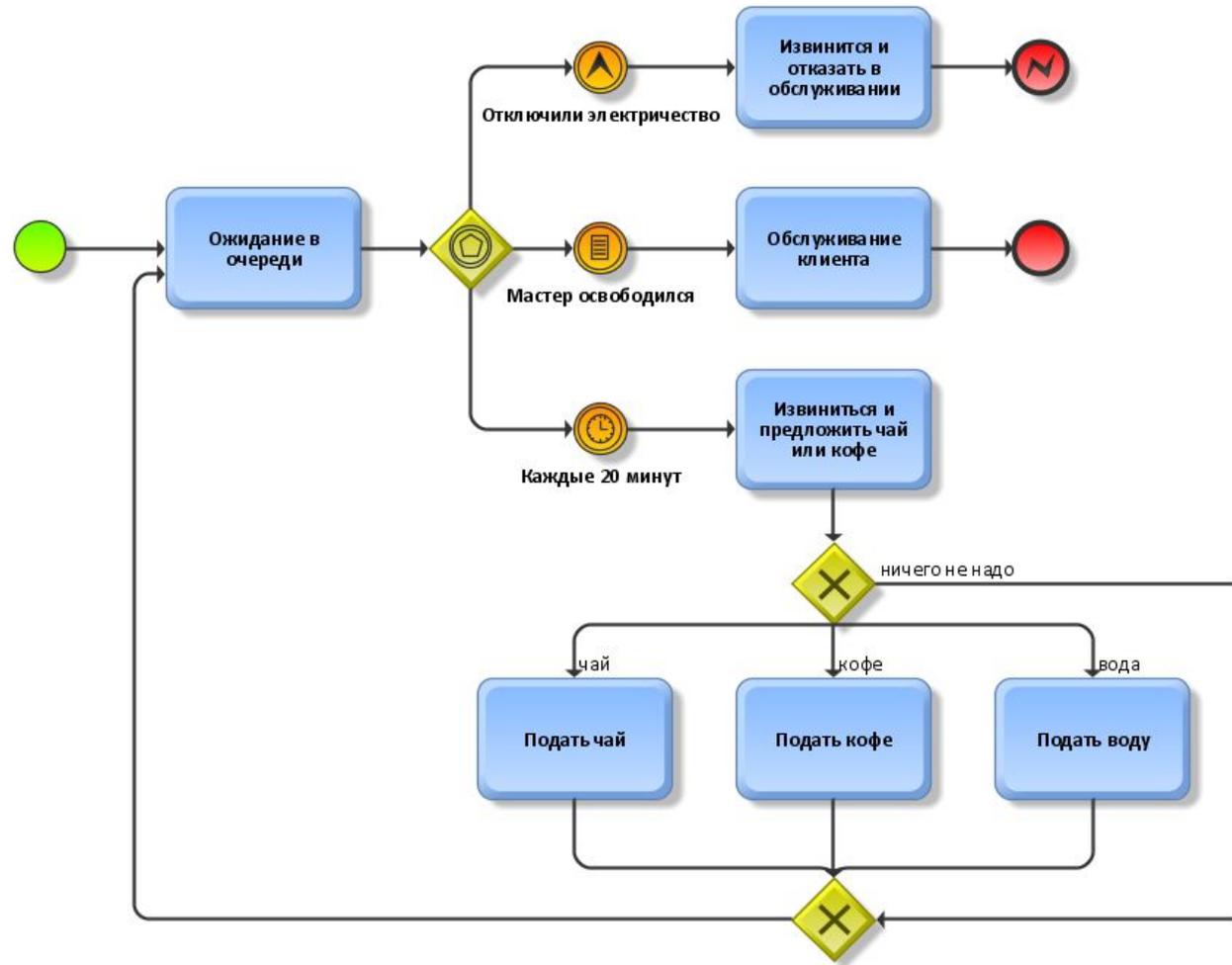


экслюзивный, основанный на событиях, запускающий процесс (Exclusive Event-Based Gateway to start a Process). Аналогичен предыдущему, но используется в качестве начального символа процесса (подпроцесса). Не имеет входящих потоков;

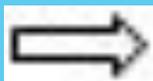


параллельный, основанный на событиях, запускающий процесс (Parallel Event-Based Gateway to start a Process). Аналогичен предыдущему, но возможна активация сразу нескольких маршрутов в случае срабатывания событий, с которыми они связаны. Возможно асинхронное выполнение маршрутов (связанных потоков операций и действий). Т.е. после активации и начала выполнения одного из маршрутов, другие маршруты тоже могут быть активированы и выполнены, пока не наступил момент завершения процесса (подпроцесса). Не имеет входящих потоков.

# Пример использование различных типов шлюзов



# Объект данных



входные данные (Data Inputs). Исходные ТМЦ или информация для выполнения действий. Отображается у верхнего края символа;



выходные данные (Data Outputs). Результат действия. Отображается у верхнего края символа;



набор данных (Data Collection). Коллекция или массив однотипных данных. Отображается у нижнего края символа.

# Потоки операций

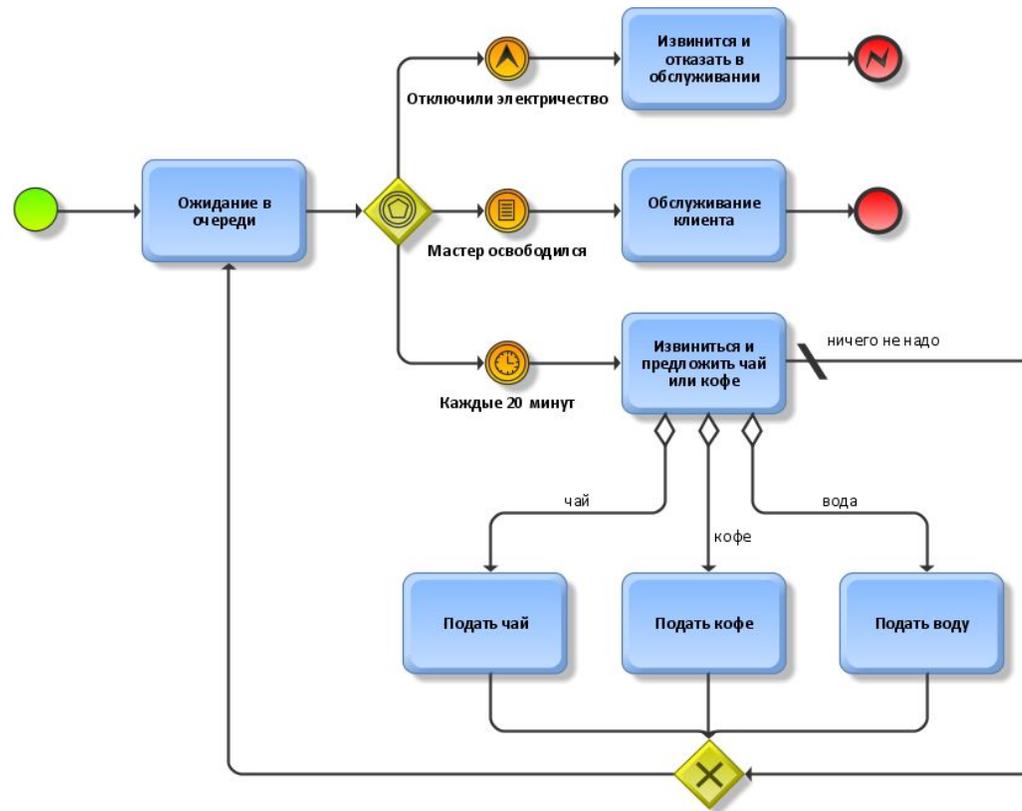


условный поток операций (Conditional Sequence Flow).  
Используется при ветвлении потоков. Обычно отображается исходящим из действия, чтобы не отображать на диаграмме шлюз. Условия активации потока задается рядом в виде логического выражения;

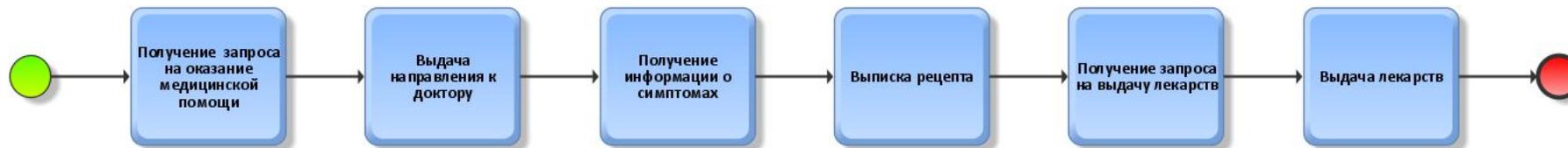


поток операций по умолчанию (Default Sequence Flow).  
Используется при ветвлении потоков. Может исходить из действия или шлюза. Не связан ни с каким логическим выражением. Поток по умолчанию активируется, если активация других потоков в соответствии с их логическими выражениями или событиями невозможна.

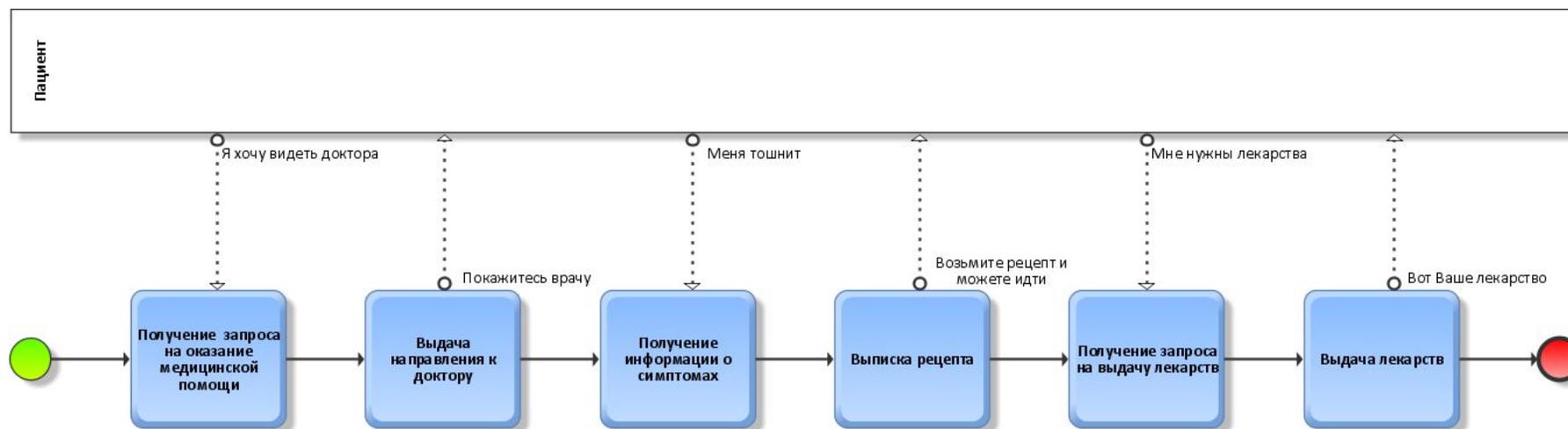
# Пример использование специфических потоков управления



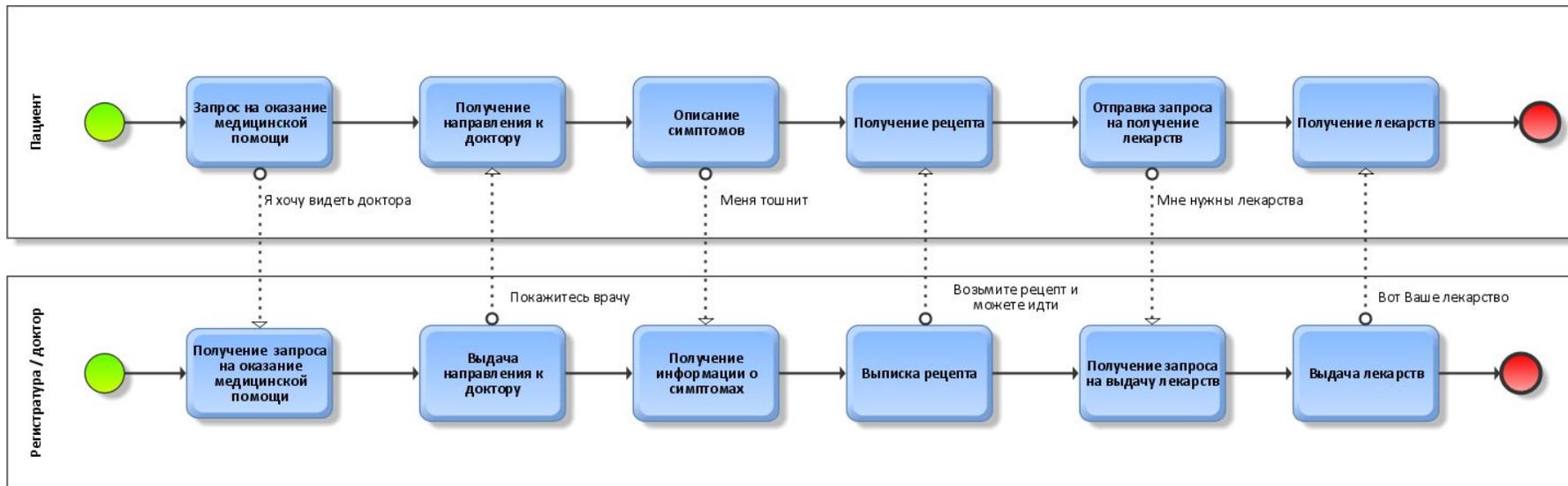
# Диаграмма внутреннего процесса



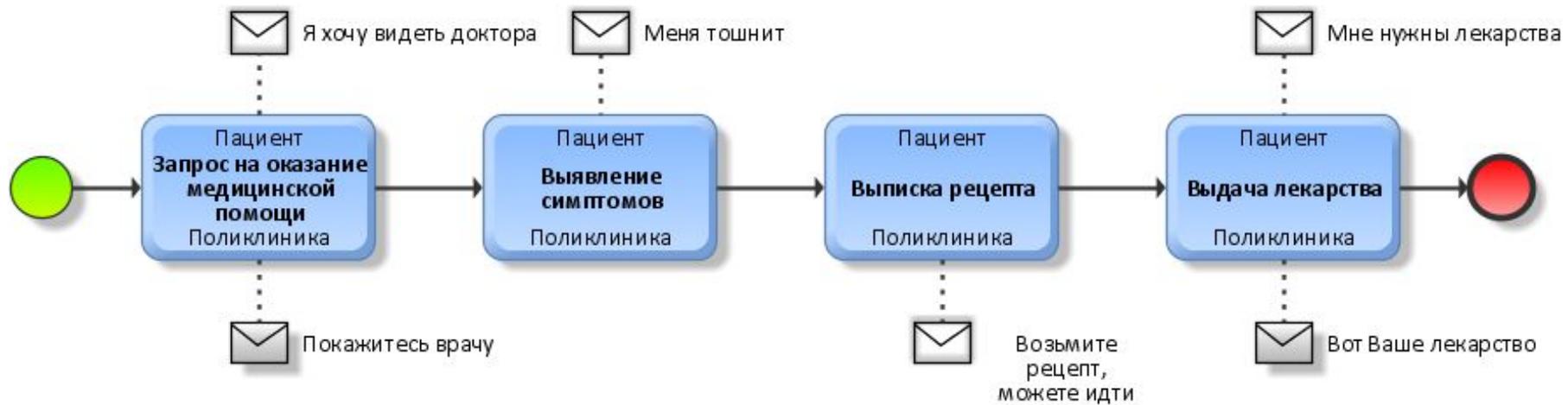
# Диаграмма открытого процесса



# Диаграмма взаимодействия процессов



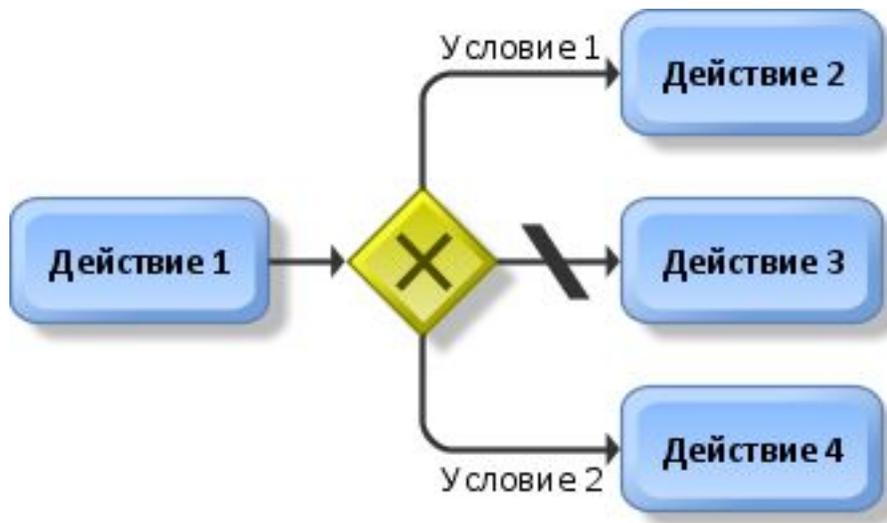
# Диаграмма хореографии



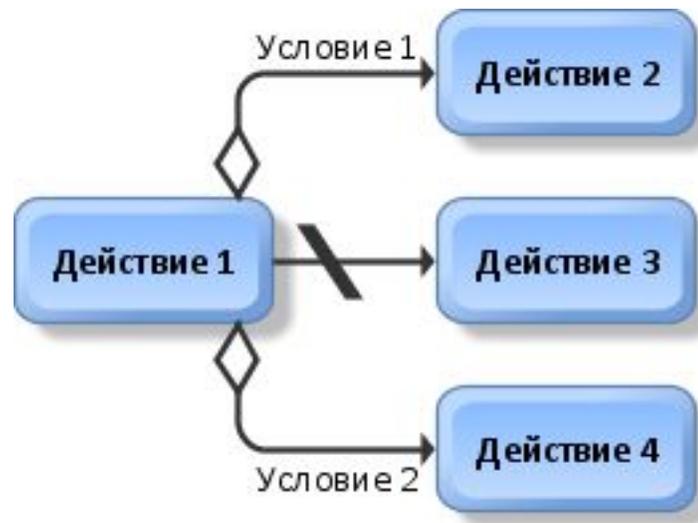
# Правила и рекомендации

- ▶ 1. Несмотря на тот факт, что события - необязательные элементы на диаграммах, рекомендуется отображать начальные и конечные события. У одного процесса (пула, дорожки, развернутого подпроцесса) должно быть только одно начальное событие, но может быть несколько конечных событий.
- ▶ 2. На диаграмме не должны присутствовать элементы без единой связи.
- ▶ 3. В отличие от EPC-диаграмм, допускается последовательное следование нескольких событий или процессов подряд.
- ▶ 4. Каждый шлюз слияния должен обладать минимум двумя входящими связями, шлюз ветвления - минимум двумя исходящими.

- ▶ 5. Ветвление на альтернативные потоки по логическим выражениям («исключающее ИЛИ» или логическое «ИЛИ») можно отобразить через соответствующий шлюз (эксклюзивный, неэксклюзивный или комплексный) или с использованием специфических потоков операций.

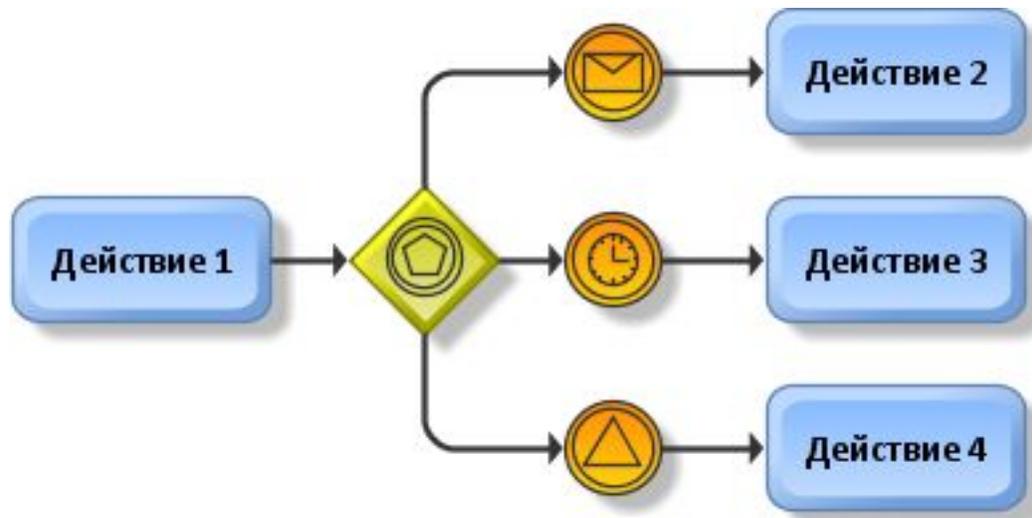


ветвление с использованием шлюза

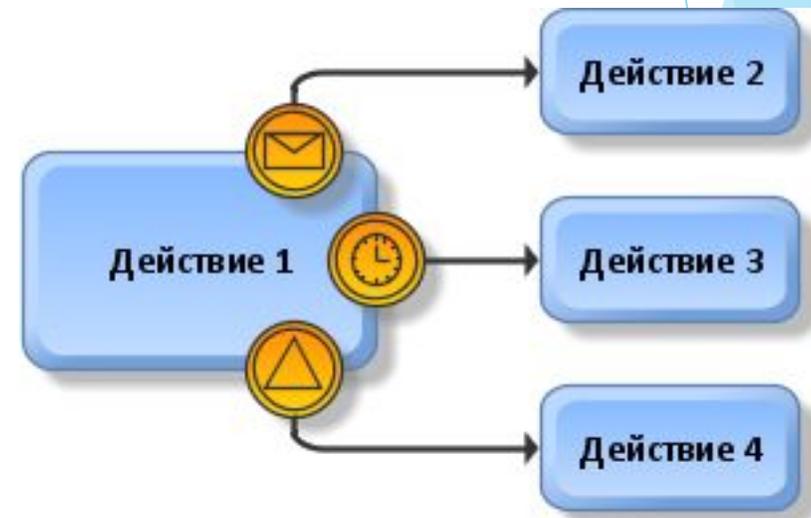


ветвление с использованием потоков

- ▶ 6. Ветвление на альтернативные потоки в зависимости от произошедших событий можно отобразить через эксклюзивный шлюз, основанный на событиях, или с использованием граничных событий.



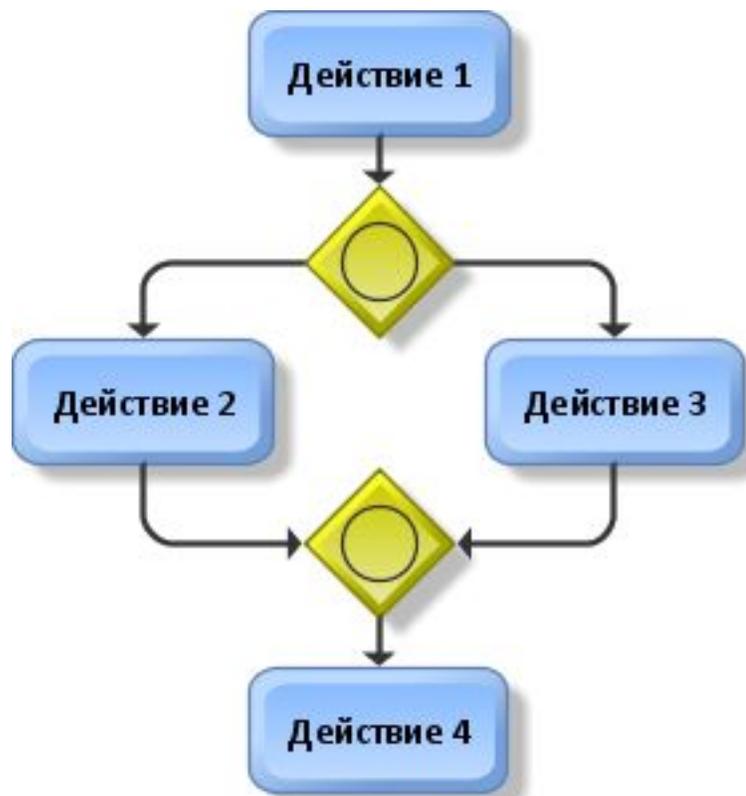
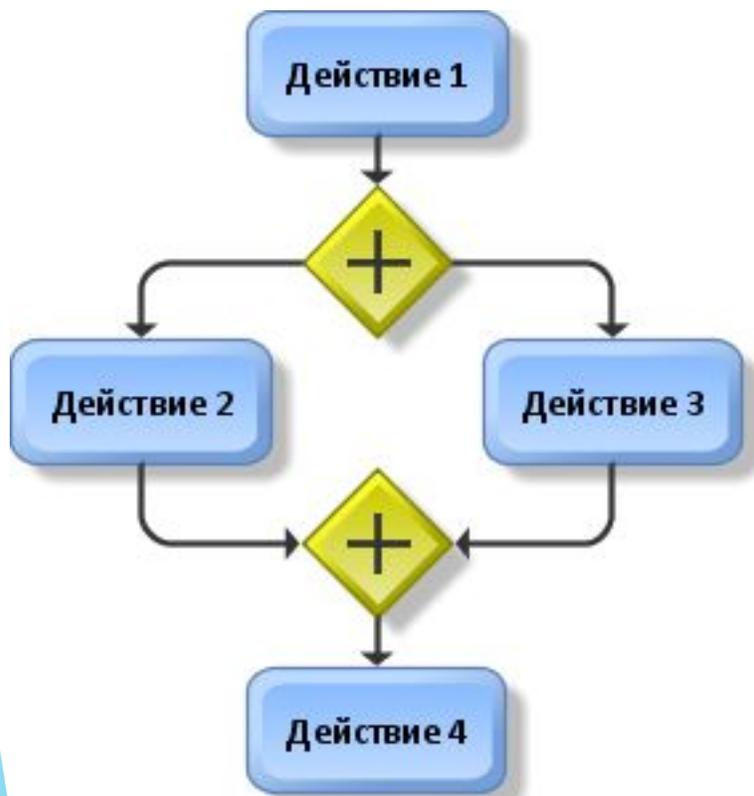
ветвление с использованием шлюза

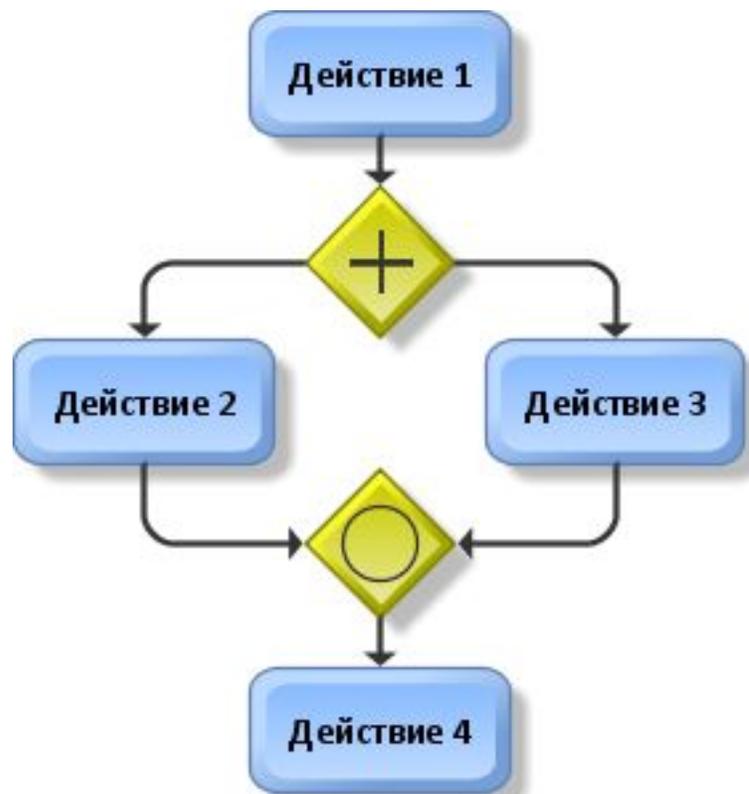
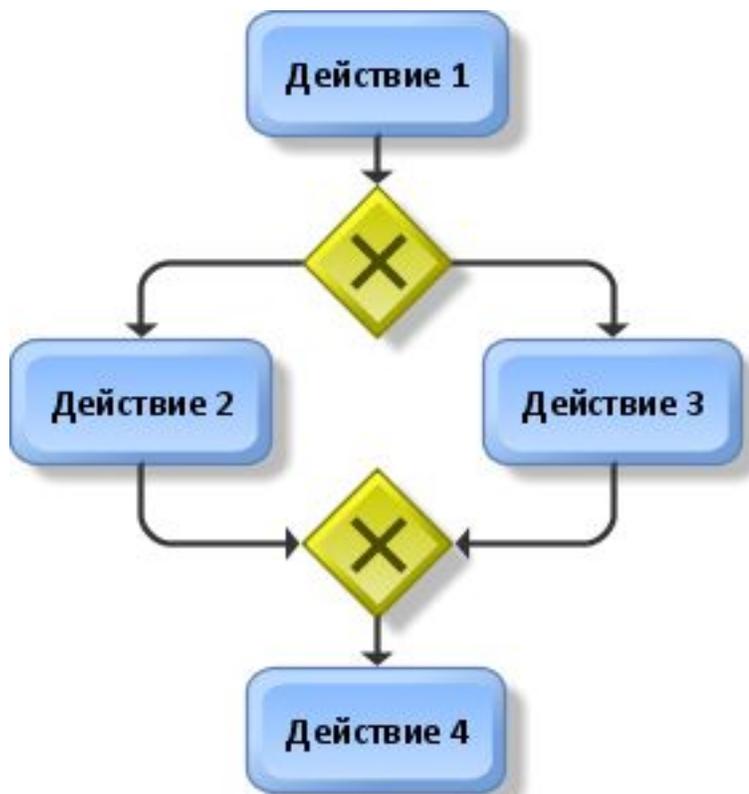


ветвление с использованием граничных событий

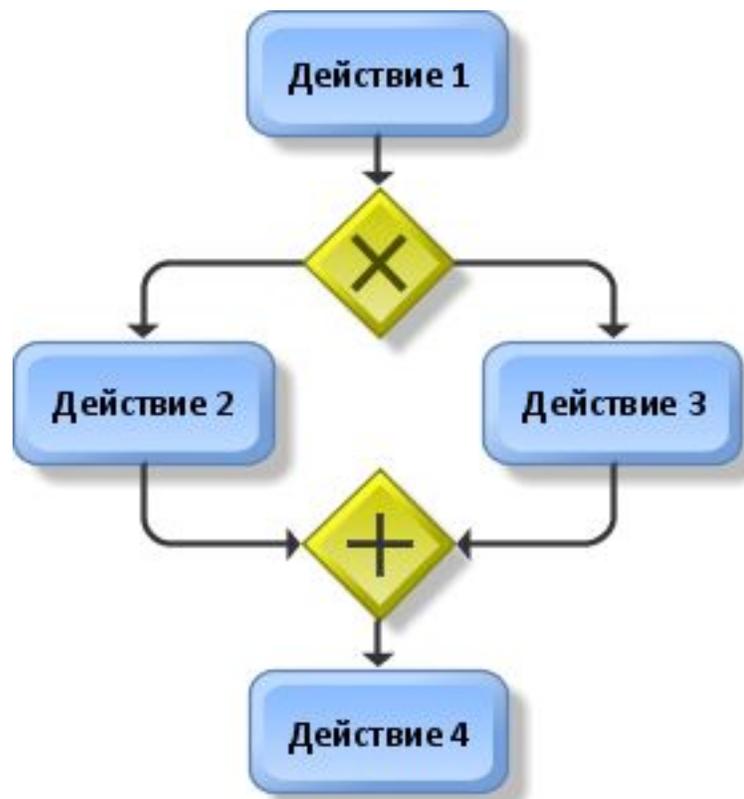
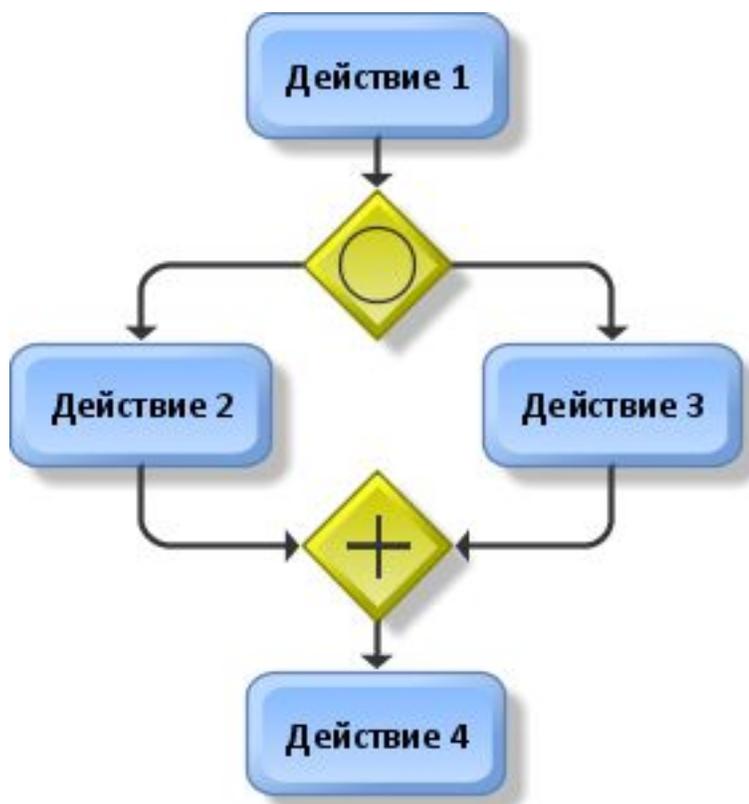
- ▶ 7. Шлюз, разветвляющий ветки, и шлюз, объединяющий эти ветки, должны совпадать. Допускается также ситуация, когда шлюз ветвления «И», шлюз объединения - «ИЛИ».

# допустимые ситуации





# недопустимые ситуации



- ▶ 8. Количество пересечений линий следует минимизировать. При этом считается, что пересекающиеся линии не имеют логической связи друг с другом. Другими словами, потоки в местах пересечений не меняют своего направления.

