

# Поведенческое проектирование

# Модели поведенческого проектирования

- - схемы алгоритмов;
- - EPC-диаграммы;
- - методология VRMN.

# ГОСТ 19.701-90



- - 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:***

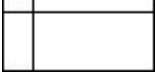



- - символы данных – указывают на наличие данных, вид носителя или способ ввода-вывода данных;
- - символы процесса – указывают операции, которые следует выполнить над данными;
- - символы линий – указывают потоки данных между процессами и/или носителями данных, а также потоки управления между процессами;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

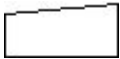


# 1. СИМВОЛЫ ДАННЫХ

## *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


# 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.




## 2. СИМВОЛЫ ПРОЦЕССА


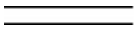
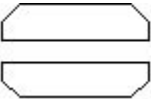
### *Основной*

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



# 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])

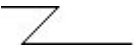
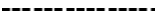
2.5		Решение	<p>Операция с одним входом и несколькими альтернативными выходами, один из которых активизируется после проверки условия, записываемого внутри символа (операторы If-Then-Else или Case)</p>
2.6		Параллельные действия	<p>Параллельное выполнения двух и более операций</p>
2.7		Границы цикла	<p>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)</p>

# 3. СИМВОЛЫ ЛИНИЙ


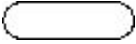
## *Основной*




3.1		Линия	Поток данных или управления
-----	---	-------	-----------------------------

# 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

# 4. СПЕЦИАЛЬНЫЕ СИМВОЛЫ

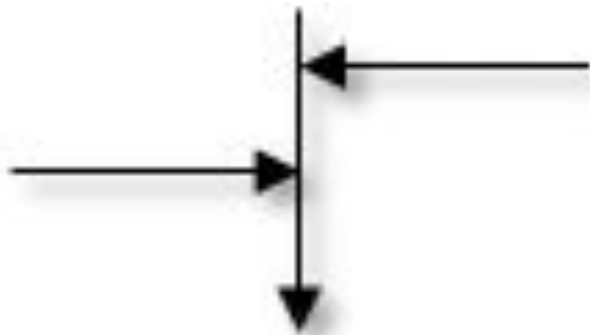
4.1	ГОСТ		Соединитель	Используется для обрыва линий и их продолжения в другом месте. Обычно используется для обозначения взаимосвязанных частей схемы на разных листах. Внутри соединителя пишется номер соединения
4.2			Терминатор	Выход во внешнюю среду или вход из внешней среды (начало и конец процесса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

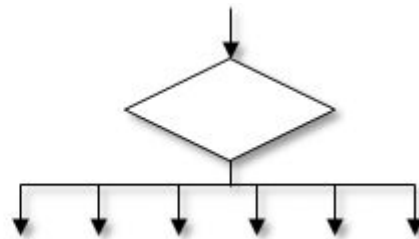
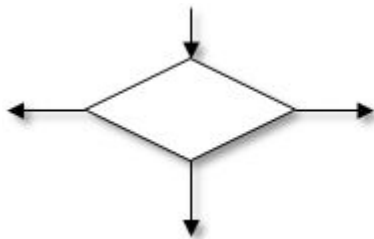
- 1. Допускается зеркально отображать символы и поворачивать их вокруг оси. В частности, запоминающие устройства с прямым доступом (таблицы на жестком диске) на схемах, как правило, поворачивают на  $90^\circ$  против часовой стрелки.
- 2. Большинство символов допускают задание внутри них текстовых пояснений. 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.

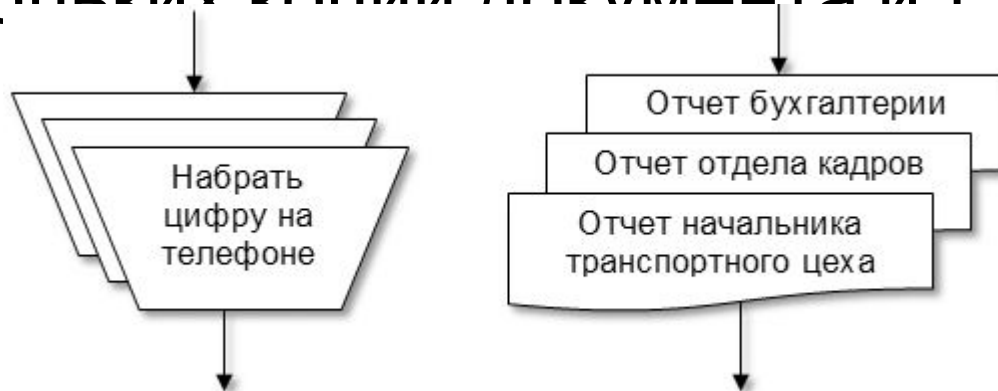




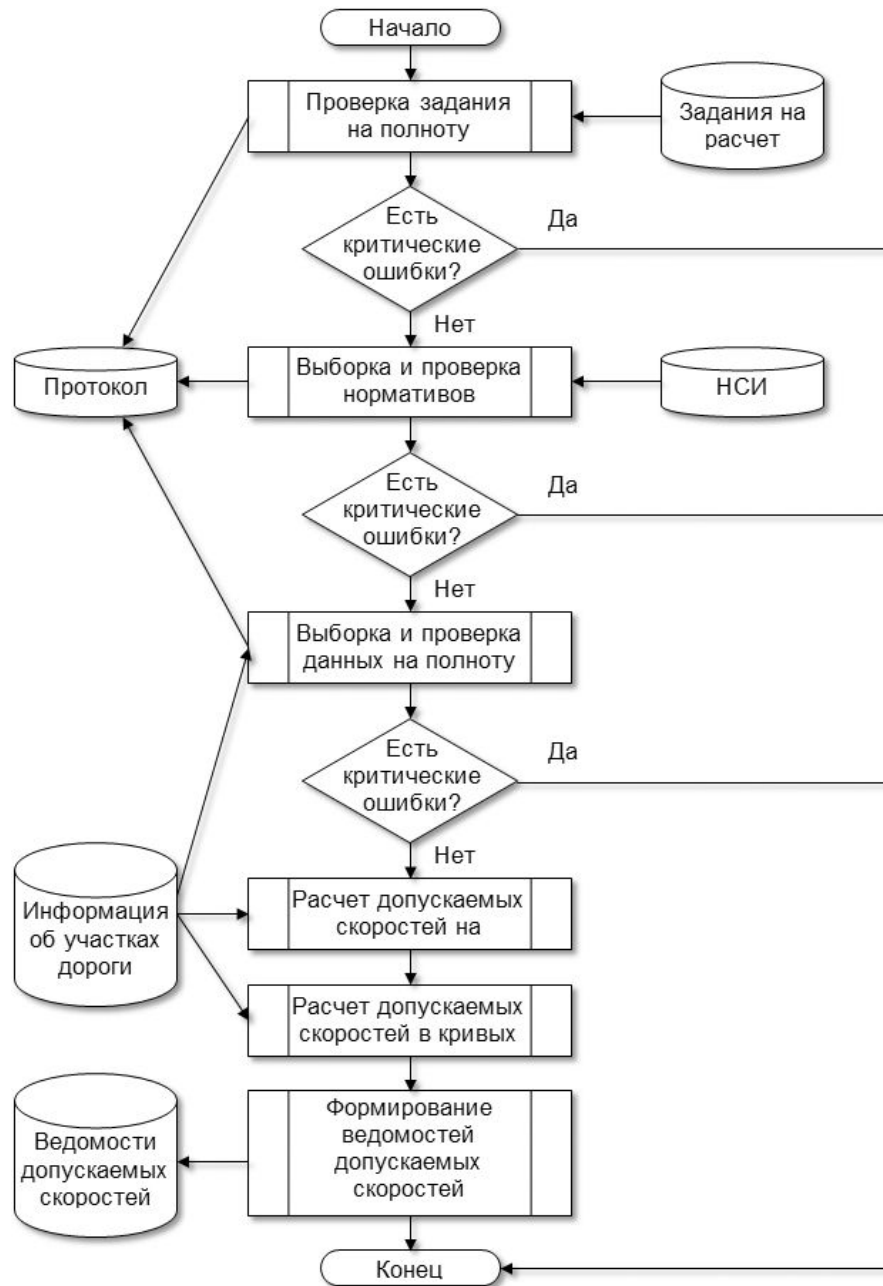
- 5. Несколько выходов из символа решения следует показывать одним из следующих способов:
- - несколькими линиями от данного символа к другим символам;
- - одной линией от данного символа, которая затем разветвляется в соответствующее число потоков.

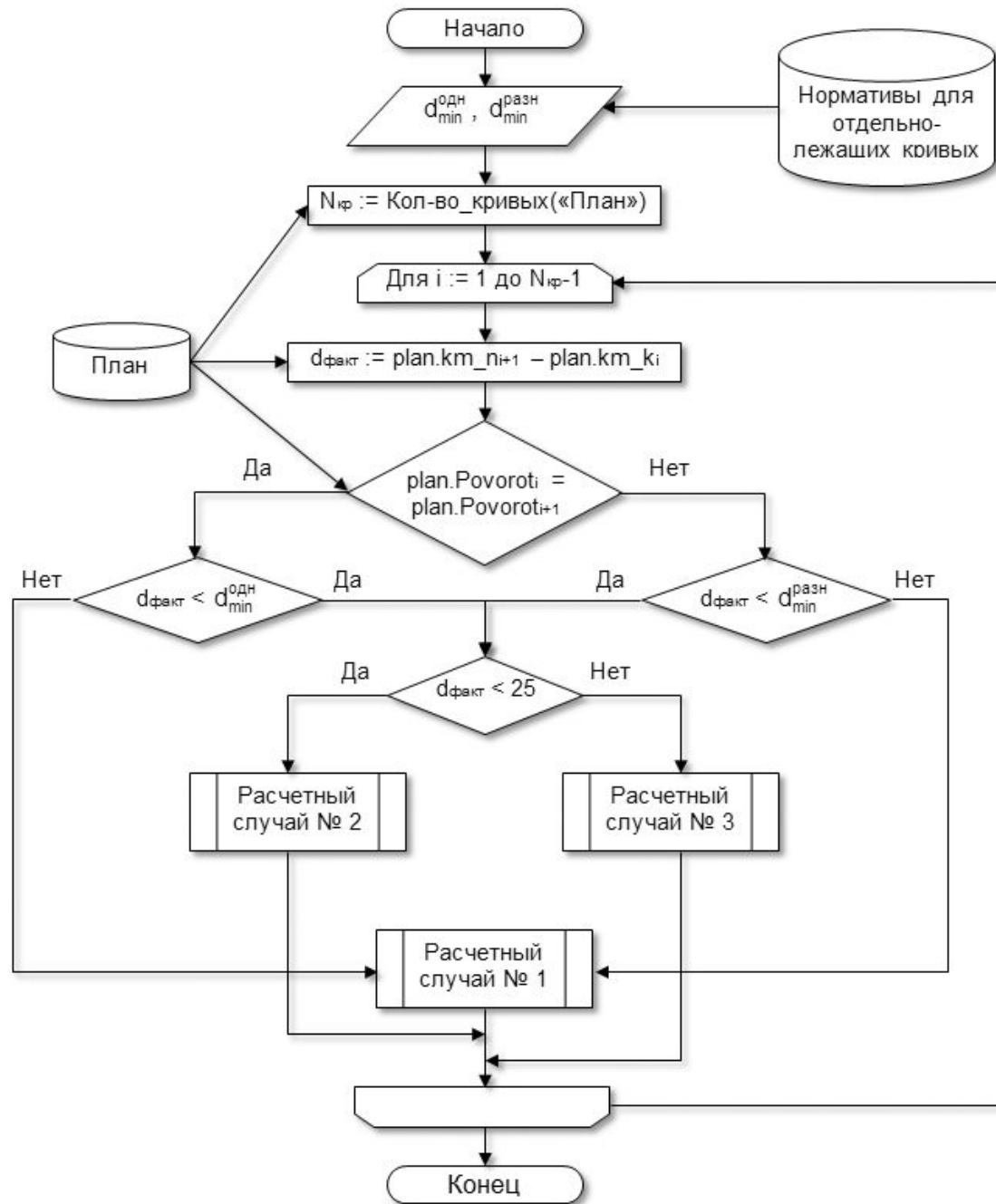


- 6. Вместо одного символа с соответствующим текстом могут быть использованы несколько символов с перекрытием изображения, каждый из которых может содержать дополнительный текст (например, запись или посылка нескольким получателям, подготовка нескольких копий документа и т. д.).



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

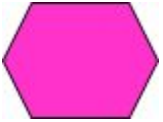

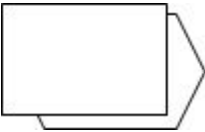




# ЕРС-ДИАГРАММЫ

- **Событийная цепочка процессов (ЕРС, event-driven process chain)**
- Август-Вильгельм Шеер (August-Wilhelm Scheer)
- ARIS (Architecture of Integrated Information Systems - Архитектура интегрированных информационных систем)
- начало 1990-х годов.





# 1. СИМВОЛЫ ПРОЦЕССА

1.1		Событие (Event)	Факт (ситуация, набор условий или обстоятельств), который активирует 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.

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

# 3. СИМВОЛЫ ИСПОЛНИТЕЛЕЙ

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

# 4. СИМВОЛЫ ПО


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


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

# 6. 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		Исключающе е «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).
-----	---	--------------------------------	---

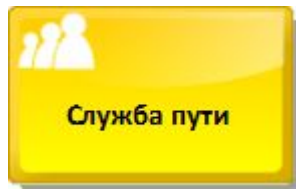


# 7. СПЕЦИАЛЬНЫЕ (ДОПОЛНИТЕЛЬНЫЕ) СИМВОЛЫ

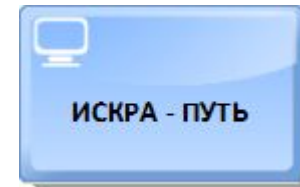
7.1		Цель (Objective )	Цель выполнения процесса (высокое качество, низкая себестоимость, сокращение издержек, увеличение прибыли).
7.2		Термин (Term)	Используется: - для обозначения данных, передаваемых между процессами или обрабатываемых при выполнении процессов (Техническое задание, Форма № 1, Ведомость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

# Отличия в цвете

а) организационная  
единица



б) информационная  
система

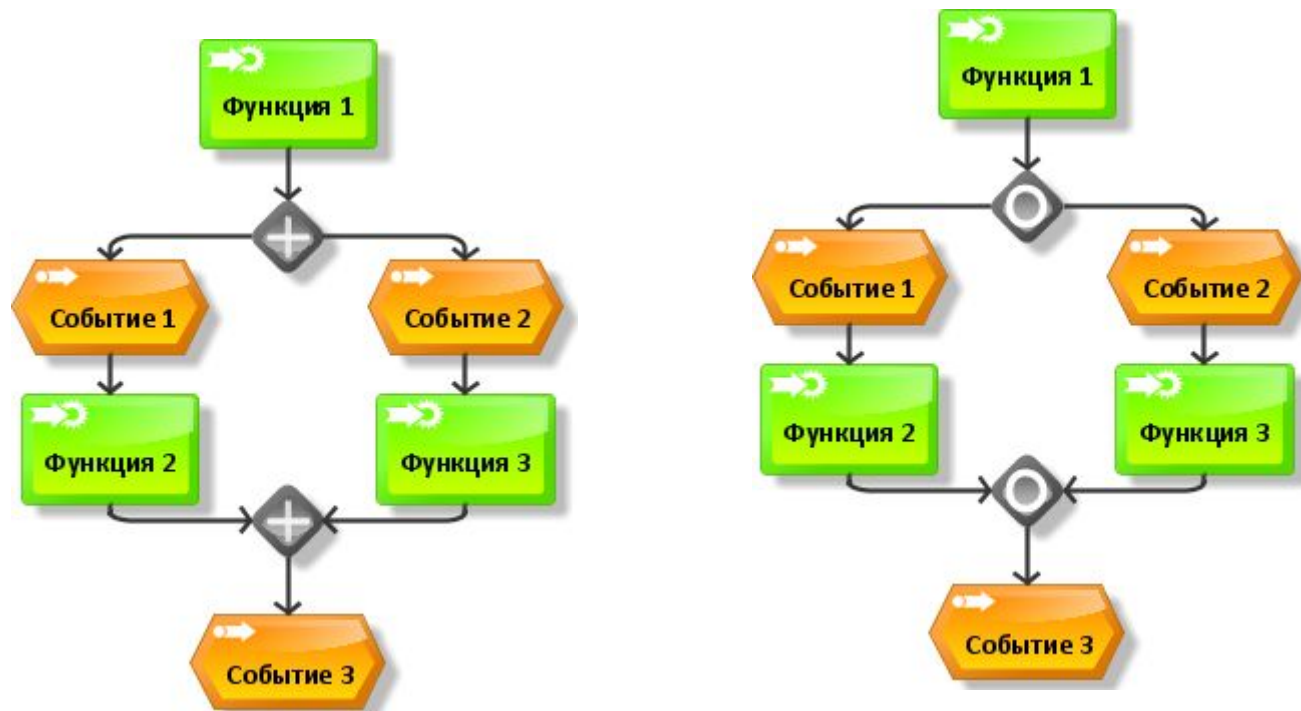


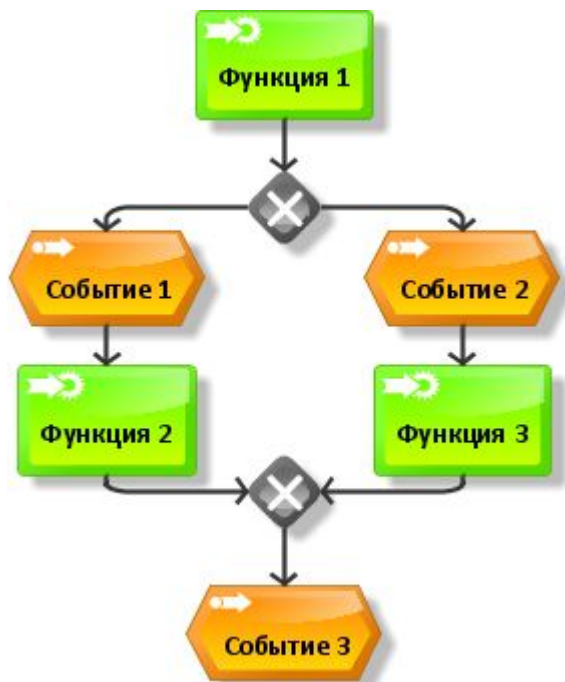
# Правила и рекомендации

- 1. Диаграмма процесса EPC 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».



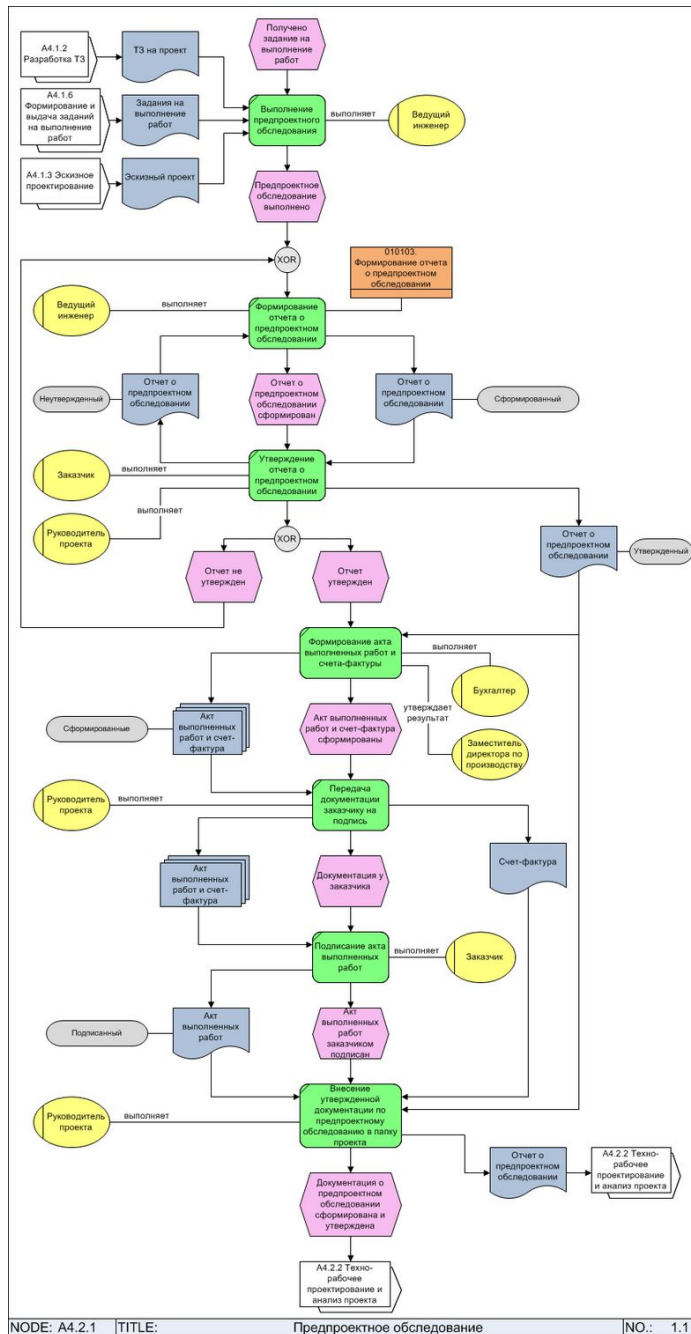


# Недопустимые ситуации



- 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

- Лучшее из:
- - 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.





# МИНУСЫ И ВАЖНЫЕ ОСОБЕННОСТИ VRMN

- **Система имеет значительное количество понятий и терминов, их нужно знать и применять грамотно.**
- **Высокий уровень вхождения.** Как и любой инструмент с широкими возможностями требует большего времени на изучение, по сравнению с другими нотациями (IDEF0, IDEF3).
- **Необходимо знание бизнес-анализа.** В VRMN модели — это не просто картинки или схемы, которые вы можете нарисовать в любом графическом редакторе. Здесь очень важна грамотная структура и четкая последовательность.

# Категории

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



# 1. 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 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		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 сообщениям и (Conversation )	Описание действия, характеризующего обмен информацией между участниками (пулами) взаимодействия.

# 2. 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.



# 3. 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.


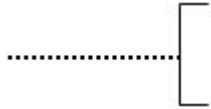
# 4. СИМВОЛЫ СОЕДИНЯЮЩИХ ЭЛЕМЕНТОВ (ЛИНИЙ)

4.1		Поток операций, поток управления (Sequence Flow)	Задает 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.

# 5. 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;

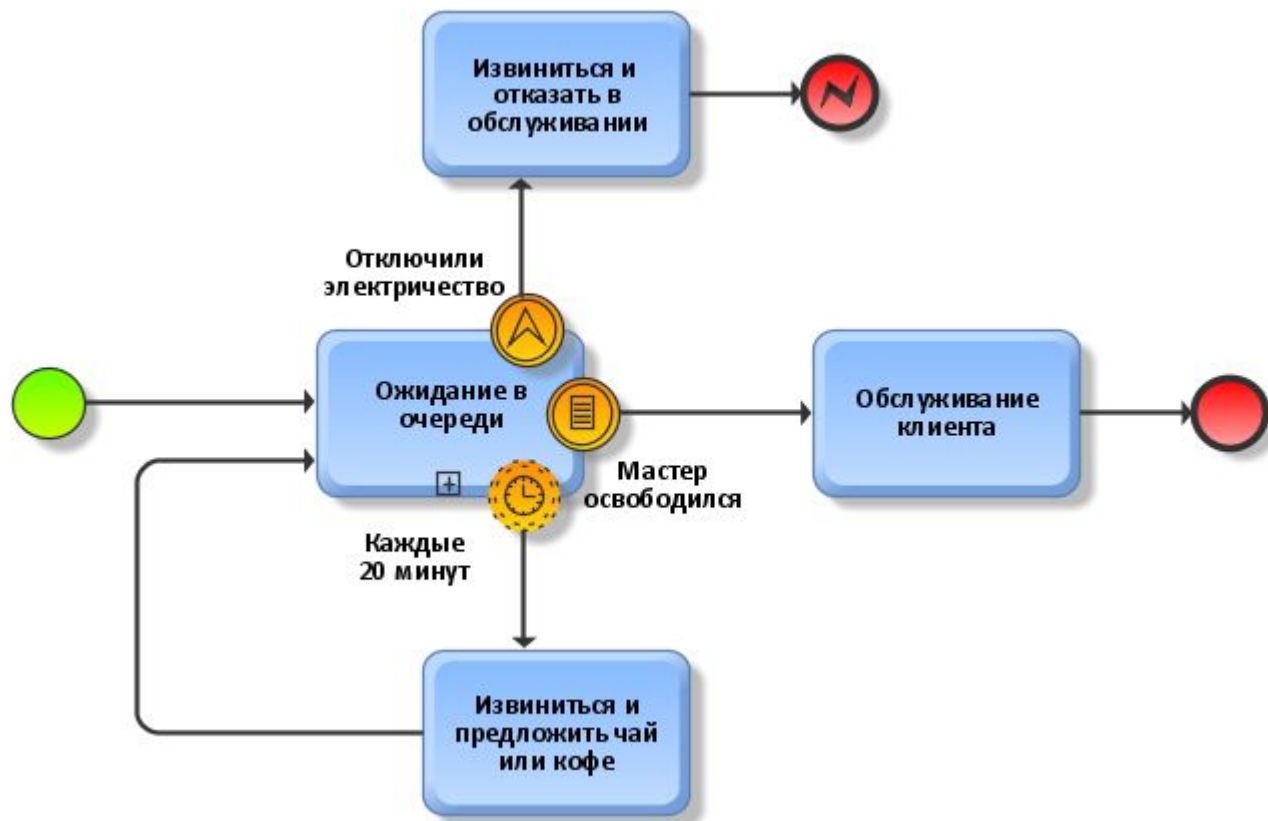
# по типу результата действия

- **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;








# Типы событий по времени возникновения и результату действия



# Типы событий по возможности прерывания выполнения действия





# Действие задача (Task)


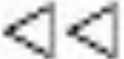
- элементарное (неделимое, атомарное) действие. Специфика (разновидность) задачи может быть отображена иконкой (маркером) в левом верхнем углу символа действия:
  -  сервисная (Service). Задача предназначена для оказания услуги, которая может являться как веб-сервисом, так и автоматизированным приложением;
  -  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 (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 + . Кроме стандартных подпроцессов, имеется еще две специфические его разновидности:
- 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 (Call). Позволяет включать в состав диаграммы повторно используемые задачи и подпроцессы. На диаграмме выделяется жирным контуром. 

# Дополнительные особенности

-  цикл (Loop). Действие выполняется в цикле с пред- (while) или пост- (repeat-until) условием;
- - ||| или ≡ - многоэкземплярность (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.

# 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;

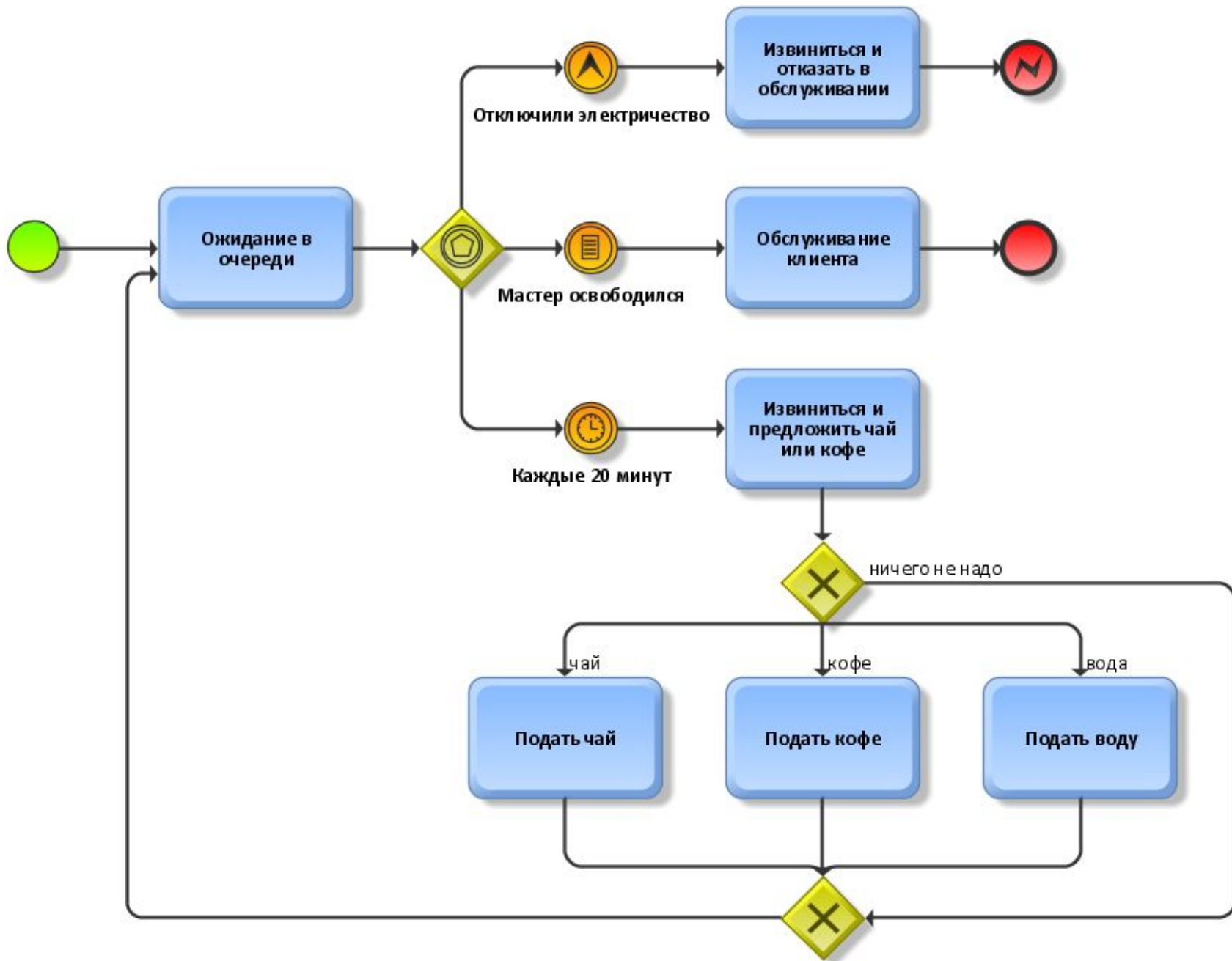


- неэксклюзивный (Inclusive, OR – логическое ИЛИ). Предназначен для разделения потока операций на несколько маршрутов, каждый из которых активируется при условии истинности связанного с ним логического выражения. Таким образом, при выполнении процесса может быть выбрано сразу несколько маршрутов, в т.ч.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

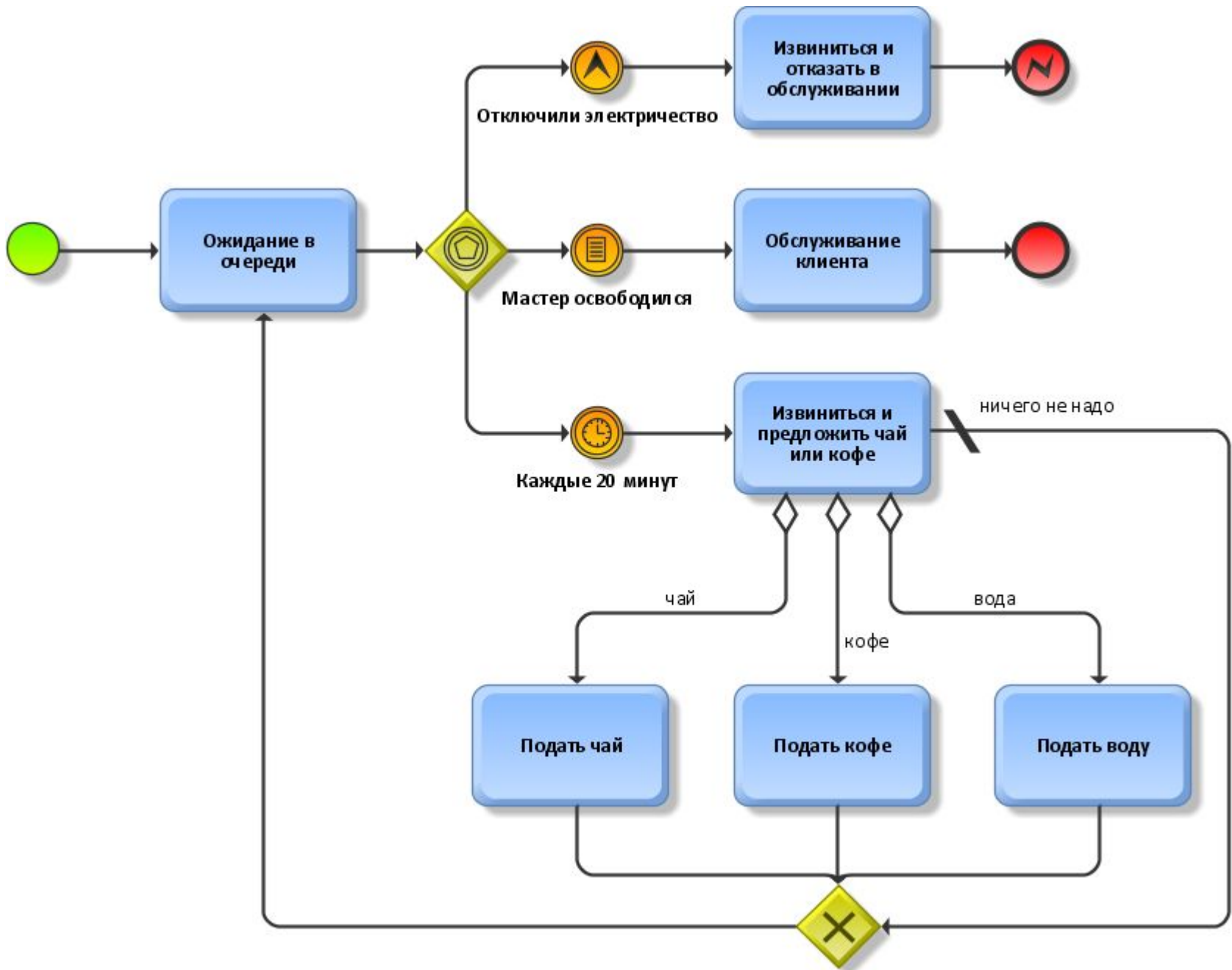


# Объект данных

- С помощью дополнительных маркеров на диаграмме может быть показана специфика использования и содержания 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.
- Связь между объектом данных и действиями отображается с помощью ассоциации.

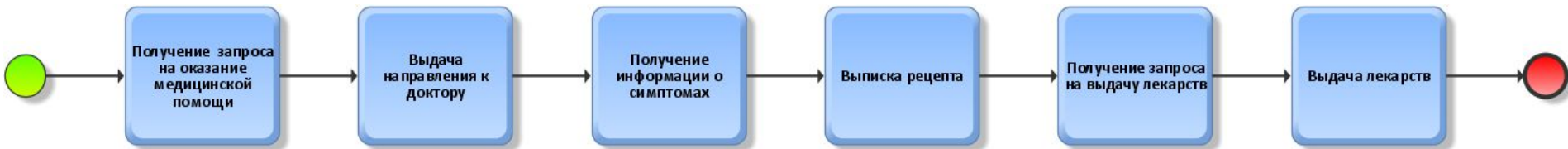
# Потоки операций

-  - 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.

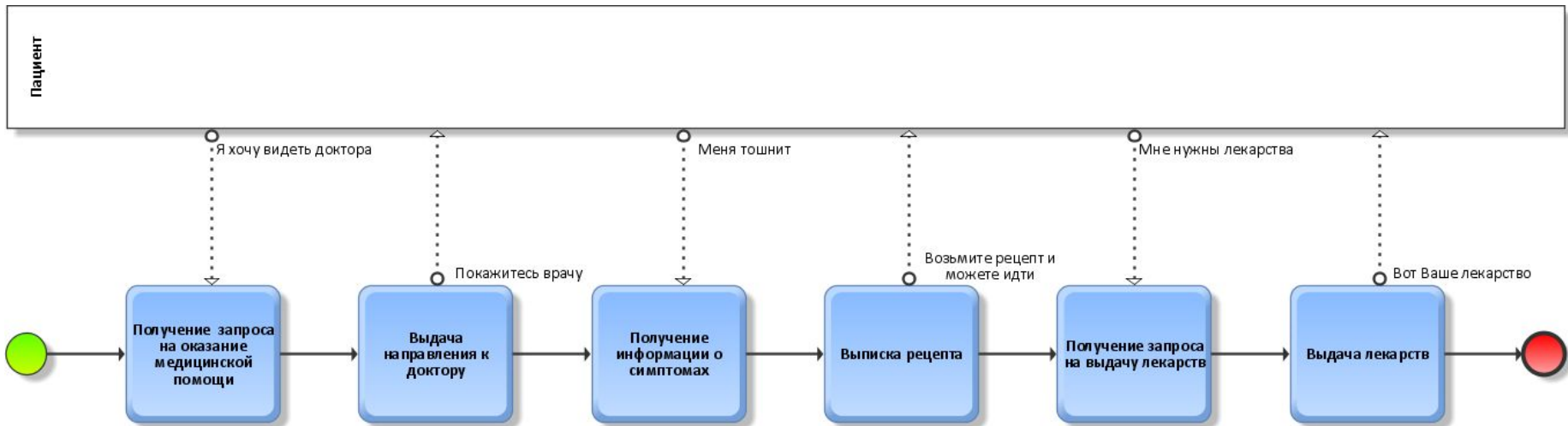




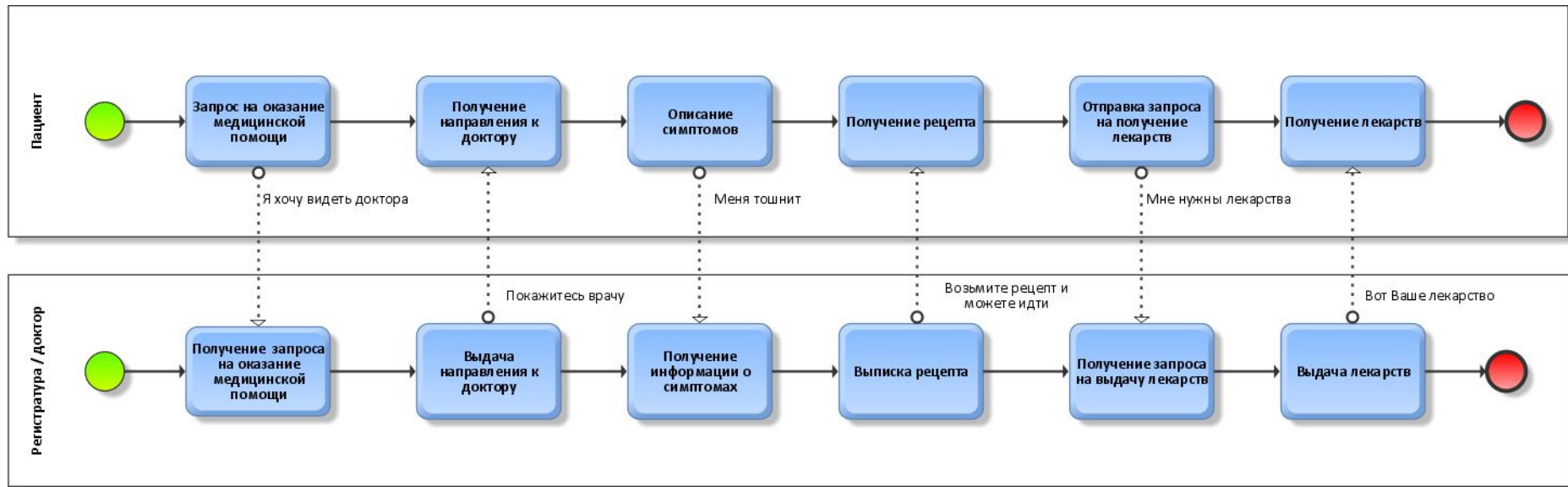
# Диаграмма внутреннего процесса



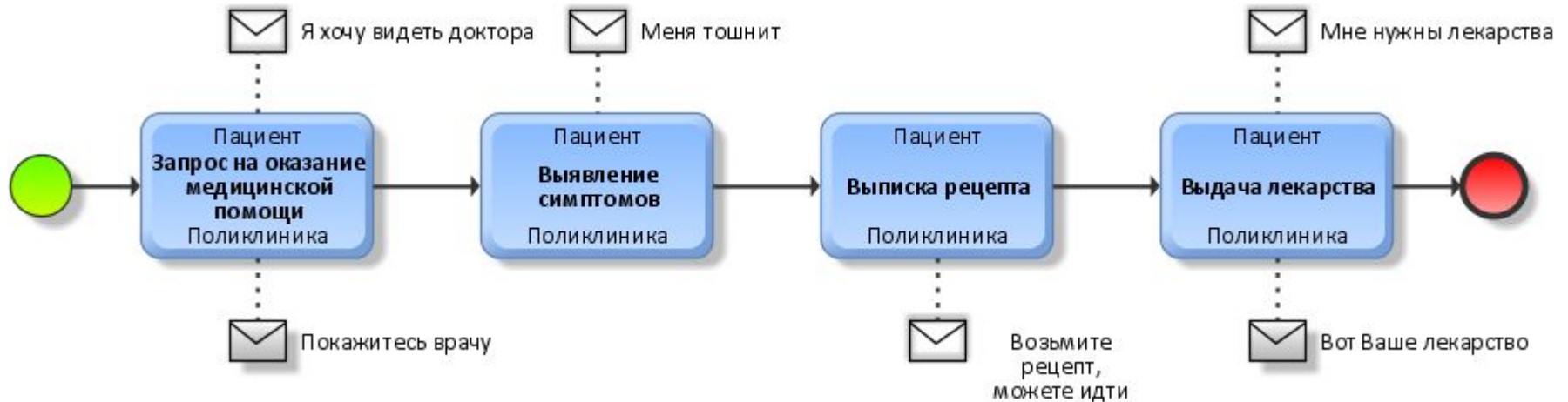
# Диаграмма открытого процесса



# Диаграмма взаимодействия процессов



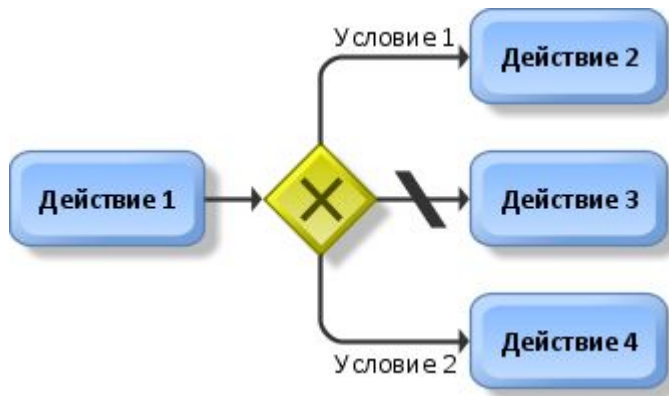
# Диаграмма хореографии



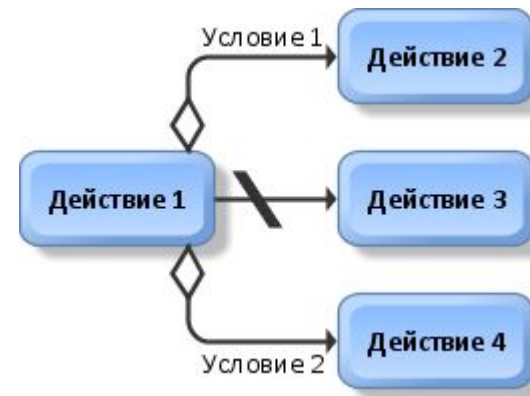
# Правила и рекомендации построения BPMN-диаграмм

- 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.

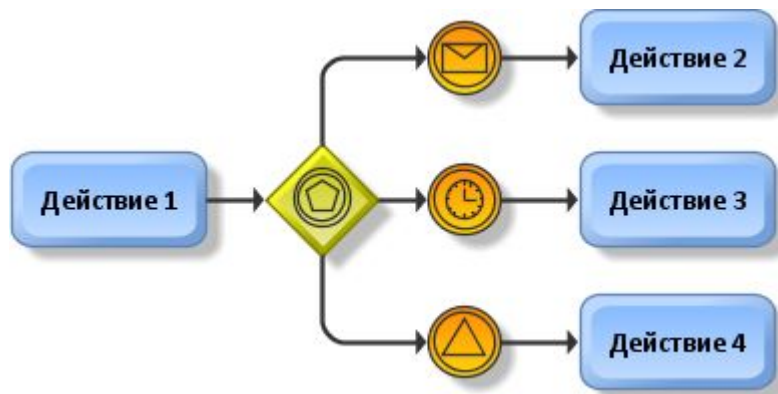


а) ветвление с использованием шлюза

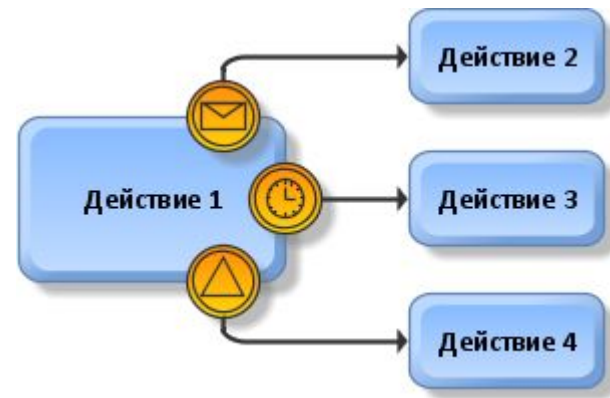


б) ветвление с использованием Потоков

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

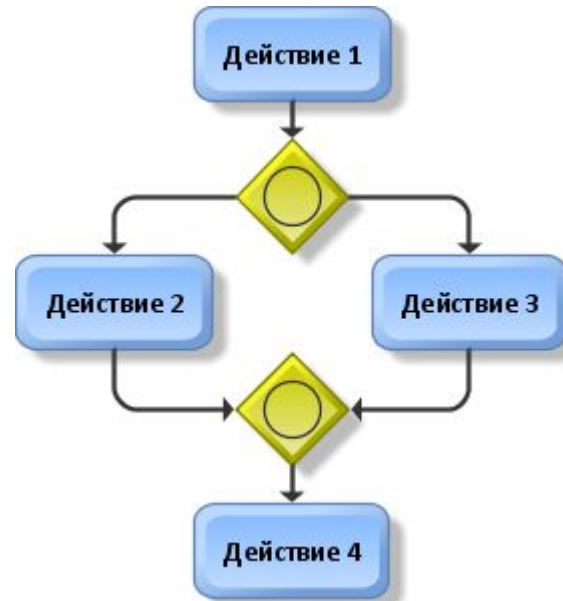
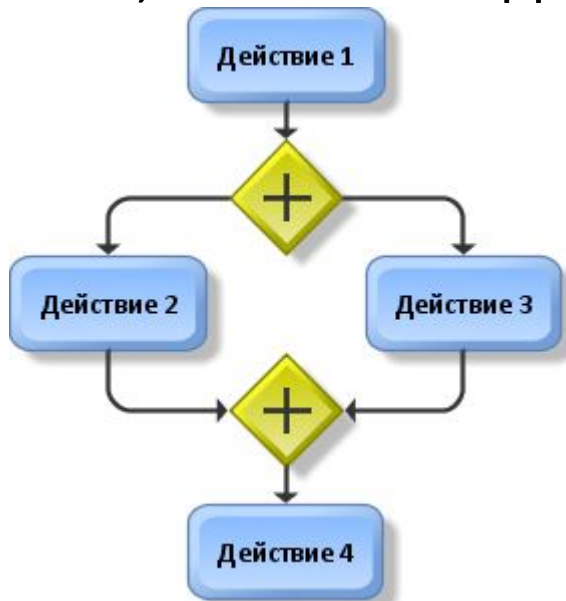


а) ветвление с использованием шлюза

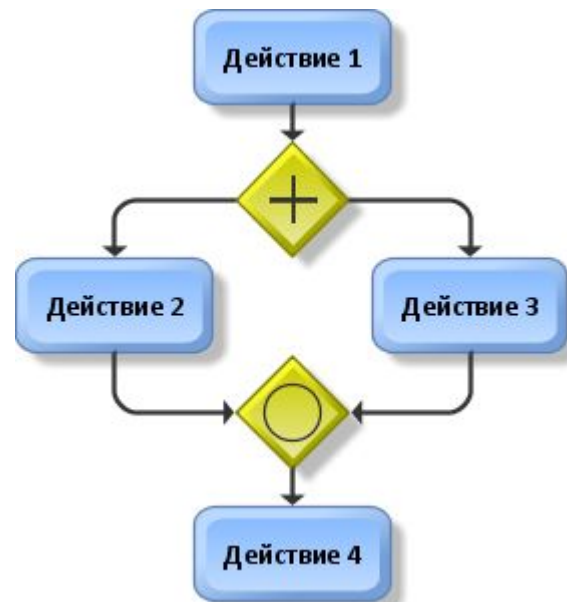
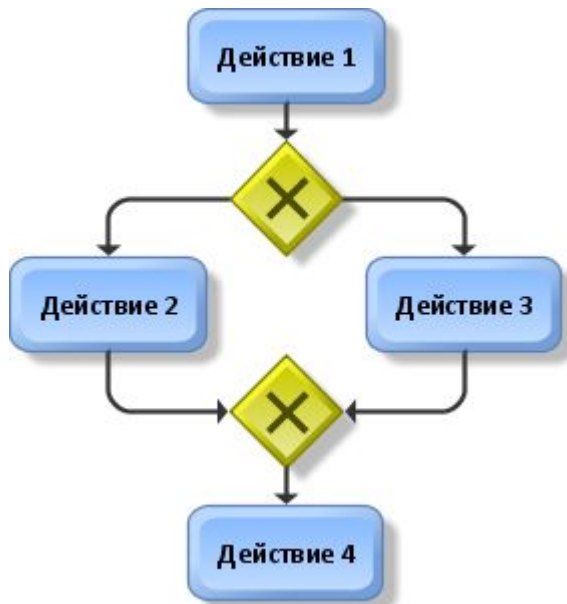


б) ветвление с использованием граничных событий

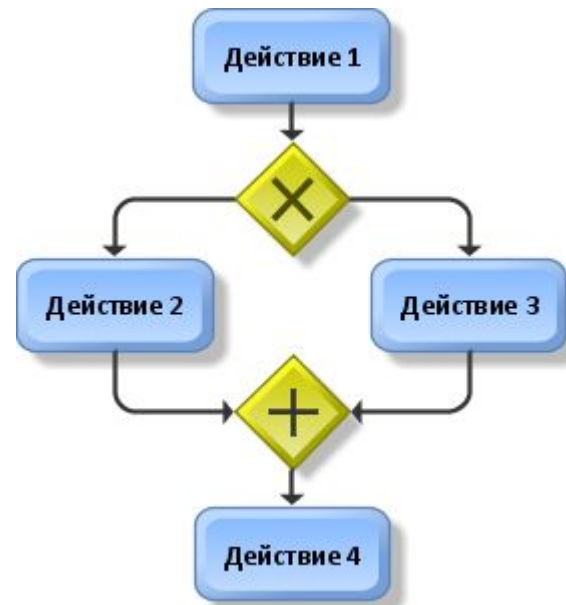
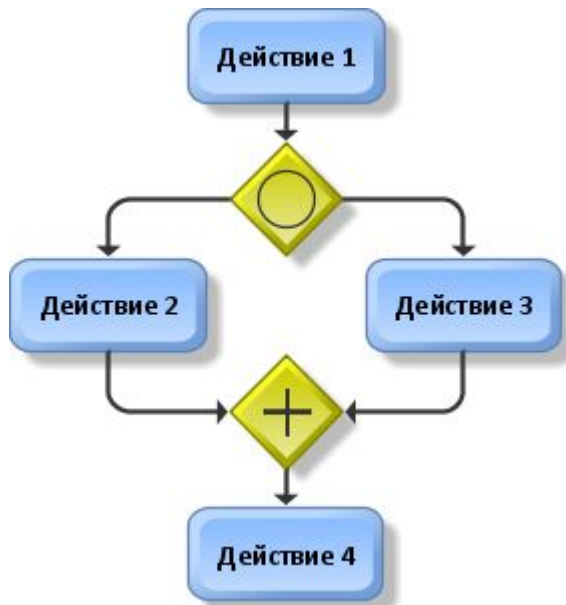
- 7. Шлюз, разветвляющий ветки, и шлюз, объединяющий эти ветки, должны совпадать. Допускается также ситуация, когда шлюз ветвления «И», шлюз объединения – «ИЛИ».



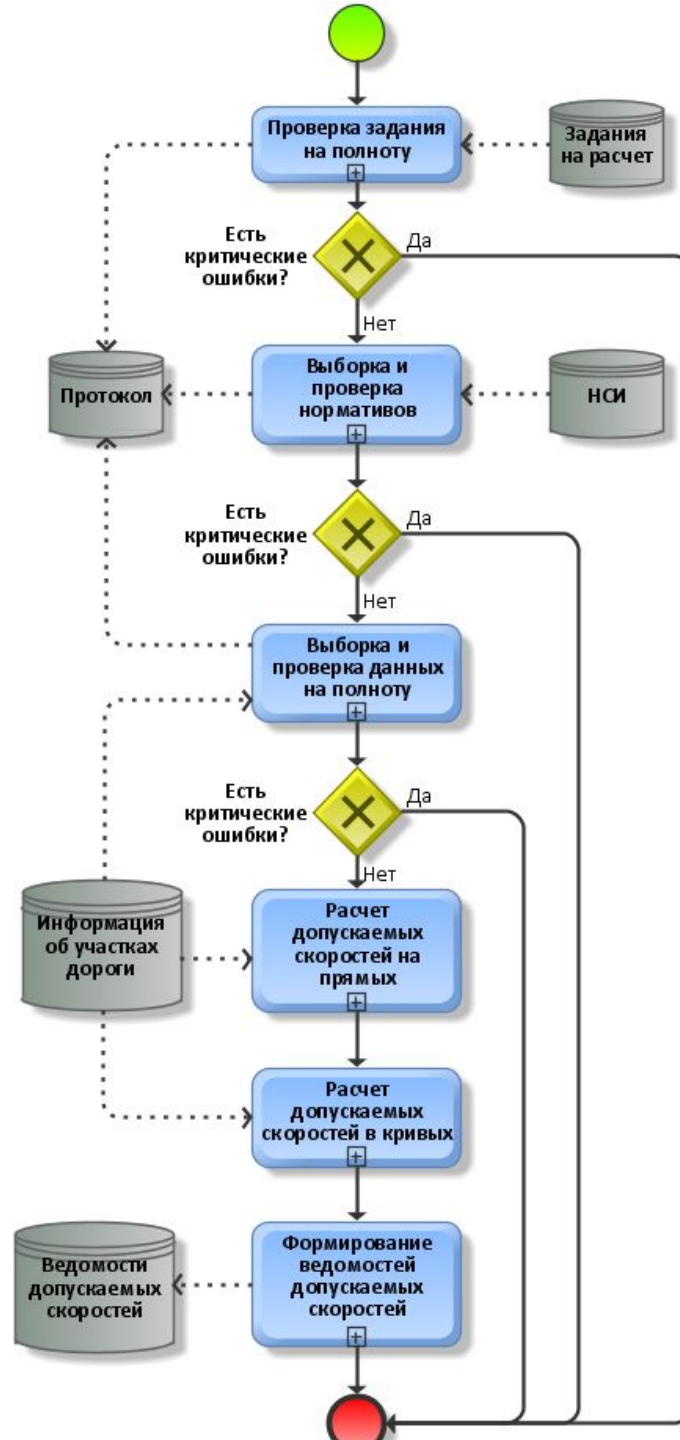


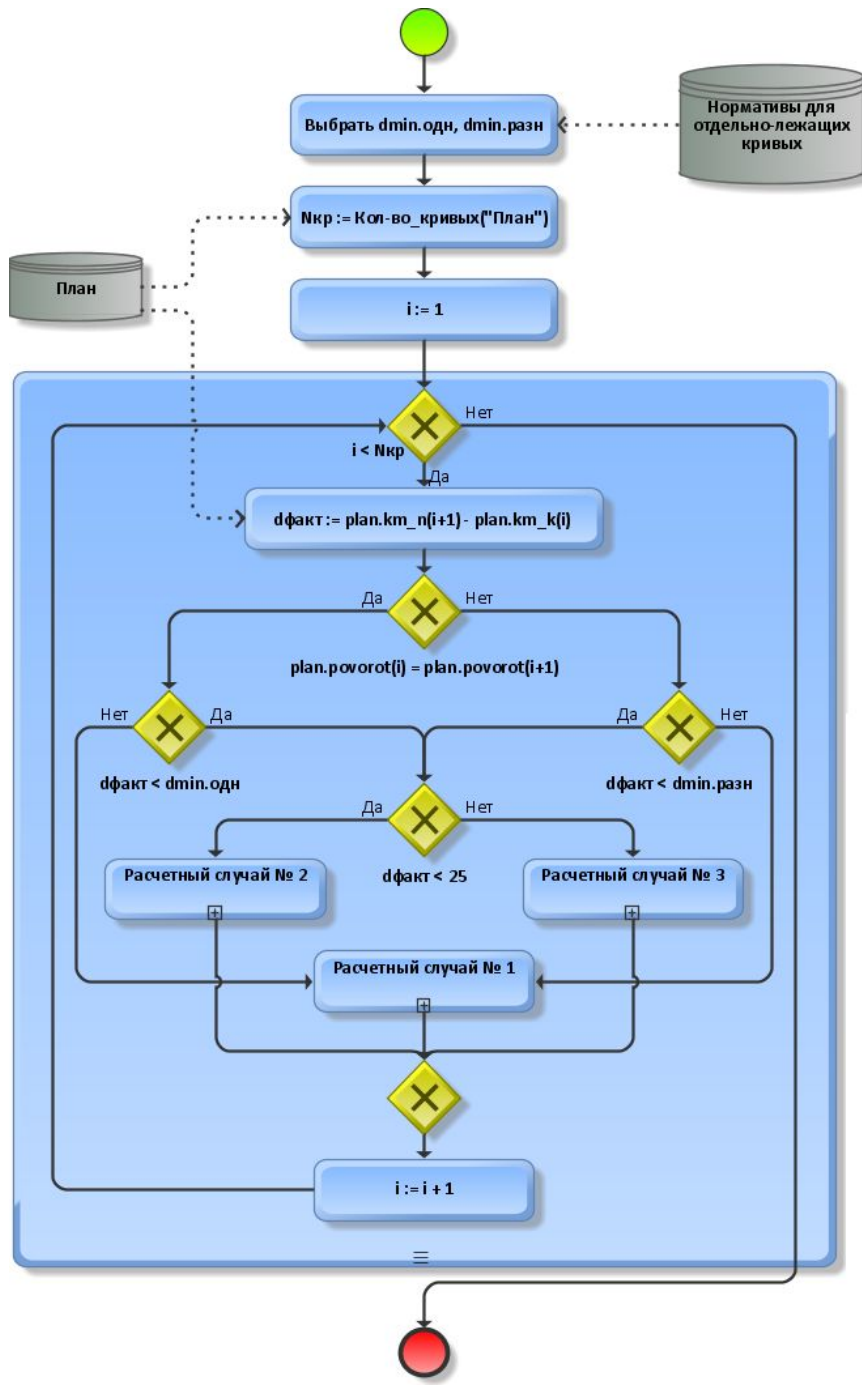


# Недопустимые ситуации



- 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.







# Операции

- **Сервисная операция**
- Операция, которая выполняется сервисом или механизмом. Иными словами, это операции выполняемые автоматически. Пример — рассчитать цену с учетом скидки. Операция выполняется программой, автоматически. Этот тип операций удобно использовать, когда вы отображаете работу программы или инструмента. Или показываете взаимодействие человека и программы. Кстати, пул, который означает сотрудника в модели, также может обозначать программу. Если задать один пул как программу, а другой как пользователя, то можно раскрыть процесс взаимодействия. Такое возможно только в нотации BPMN 2.0.

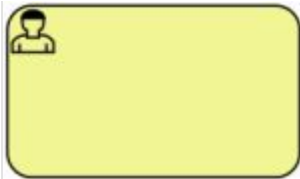


- **Отправка сообщения**
- Операция, результатом которой является отправленное сообщение. И вот первый пример гибкости нотации — операция «отправить электронное письмо», может быть, как типом «пользовательская операция», так и «отправка сообщения». Какой тип использовать — зависит от целей моделирования конкретного процесса и правил, которые вы для себя приняли.



- **Получение сообщения**
- Операция, связанная с получением сообщения. Пример — получить письмо на почте.





- **Пользовательская операция**
- Операция, которая выполняется сотрудником, с помощью сервиса, инструмента или других сотрудников. Это может быть программа, веб приложение, оборудование и так далее. Если вы обозначили операцию как пользовательская, свяжите ее с сервисом или инструментом, который в ней используется. Пример пользовательской операции — отправить электронное письмо. Сервисом будет выступать почтовое приложение. Еще пример — просверлить отверстие в стене. Инструмент — перфоратор.



- **Ручная операция**
- Операция, которая выполняется сотрудником самостоятельно, без применения каких то сервисов или инструментов. Пример — почистить апельсин.



- **Выполнение сценария**
- Операция, которая подразумевает выполнение сценария. Сценарий создается заранее и представляет собой последовательность действий. По сути, сценарий это тоже процесс. Проще всего понять сценарий, как процедуру. Например — проверить документ. Эта операция подразумевает выполнение ряда действий, т.е. мы проверяем конкретные пункты в документе. Это и есть сценарий. Если вы используете этот тип операции, то нужно обязательно указать, какой сценарий используется. Кстати, когда для проверки используется чек-лист, то такая операция явно имеет тип «выполнение сценария», а чек-лист — сценарий.
- Сценарий, это очень удобный элемент, который позволяет «облегчить» модель за счет уменьшения количества отображенных операций. Т.е. мы убираем операции, которые можно преобразовать в сценарии.



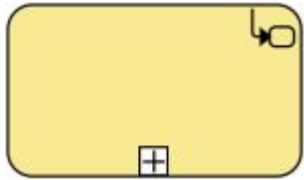
- **Выполнение бизнес правила**
- Операция, которая запускает действие какого то правила. Сценарии и правила схожи. И то, и другое создается и известно заранее. Но есть и разница. Сценарий — ряд действий, который нужно выполнить, чтобы завершить задачу. Правило же, предполагает наличие определенных условий и вариантов действий. Сценарий выполнять обязательно. Правило — дает выбор. 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. Кстати в этом примере, операция «правило», может запустить операцию «сценарий».
- Бизнес правило это уникальная штука! И с точки зрения моделирования бизнес процессов, и с точки зрения выполнения. Есть мнение, что описанные бизнес процессы, строги и не дают свободу действий сотрудникам. Это не верно. И как раз бизнес правила, дают допустимую свободу действий — модели с точки зрения развития процесса, сотруднику с точки зрения действий.



- **Операция-ссылка**
- Операция ссылается на другой процесс и фактически выполняет его. Т.е. процесс уже где то существует и здесь он тоже выполняется. В таком случае, чтобы не копировать процесс из одного места в другое, просто дается ссылка на него.

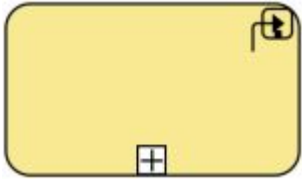


- **Операция-вызов**
- Отличается от ссылки тем, что если вы используете операцию-вызов, то еще и используете ресурсы этой операции или процесса. Т.е. «одалживаете» сам процесс, из другого места.



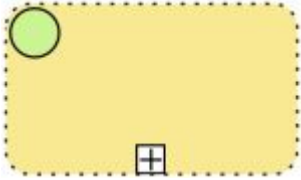
# Процессы

- **Повторно используемый процесс**
- С помощью данного элемента, определяется место в процессе, где используется сторонний подпроцесс. Например — по ходу приготовления ужина, возникает ситуация, когда для его продолжения нужно сходить в магазин. Соответственно «Сходить в магазин» будет повторно используемым подпроцессом.

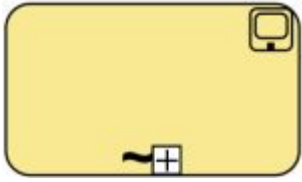


- **Процесс-ссылка**
- В некоторых ситуациях, нужно сослаться на процесс. В таком случае используется этот элемент. Например, вы можете указать что процесс-ссылка является источником некого документа или события.

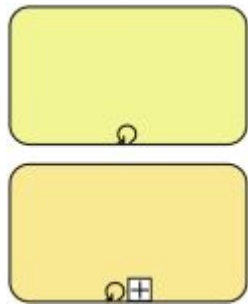




- **Событийный процесс**
- Особый вид процесса. Особенность заключается в том, что такой процесс не имеет входящих/исходящих потоков. Т.е. на диаграмме он не соединен стрелками с другими процессами/операциями. А запускается он, когда в процессе наступает событие, такое же, какое указано в событийном процессе в качестве старта. Зачем это нужно? Иногда не нужно включать процесс в основной поток. Например потому, что событие, запускающее процесс, возникает крайне редко. При этом, ход основного процесса, может как прерываться и переходить в событийный подпроцесс, так и нет. Т.е. событийный процесс выполняется параллельно. Например — в процессе обслуживания клиента в банке, оператор видит, что клиент является мошенником. В таком случае он подает сигнал службе безопасности, который служит стартом для соответствующего процесса. Однако оператор не прерывает процесс а продолжает обслуживание, с целью удержания мошенника на месте, до прибытия сотрудников.

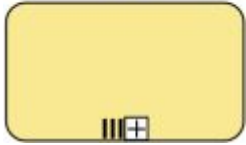
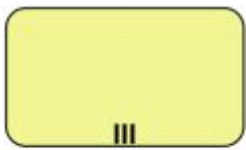


- **Спонтанный процесс (ad-hoc)**
- Еще один вид особого процесса. В спонтанном процессе, операции не имеют последовательности действий. Т.е. они могут выполняться спонтанно, в любом порядке, с любым количеством повторений. Выполнение такого процесса полностью определяется его исполнителем. Например, спонтанный процесс «Приготовить кофе», имеет в своем составе операции: «Насыпать кофе», «Насыпать сахар», «Налить воду». Порядок действий не важен, т.к. в любом случае, мы получим один результат. Что еще важно — не все действия в спонтанном процессе обязательны для выполнения. События начала и окончания, не могут включаться в спонтанный подпроцесс.

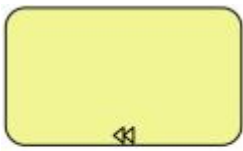


# Повторения

- Повторение может относиться как в операциям, так и к процессам.
- **Стандартное повторение**
- 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. Только не забывайте указывать условия выхода из повторения.



- **Множественное повторение**
- Этот тип повторения отличается от предыдущего тем, что 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.



- **Компенсация**
- Операция или процесс компенсации, выполняется в том случае, если в процессе, произошло что-то, требующие дополнительных действий для его завершения. Например — при согласовании документа, необходимо внести поправки. В таком случае, выполняется компенсирующие действие «внести поправки», которое позволяет завершить процесс согласования. Форма компенсации очень удобна для моделирования. Компенсация — отличительная черта нотации BPMN 2.0.

# Входящие / исходящие события

- Прежде чем приступить к тригерам, объясню, что такое входящие и исходящие события.
- Событие с входящим триггером (входящее событие), означает что оно наступает, если мы получаем какой то сигнал, сообщение и т.д. Например получение письма — входящий триггер.
- Событие с исходящим триггером (исходящее событие) означает, что событие свершается, если что то отправлено. Опять же — отправлено письмо. Исходящими событиями удобно отображать выполнение условия по передаче информации.
- **Стартовые события**, могут быть **только входящими**, а **события окончания** — **только исходящими**. Логика очень простая — стартовое событие запускает процесс, а запуск может происходить только при получении информации. Событие окончания, которое отражает факт завершения процесса, может только передавать информацию. Ведь если информация продолжает поступать, то процесс не завершен.
- Промежуточные события могут быть как входящими (получение информации), так и исходящими (отправление информации). К таким событиям относятся: сообщение, эскалация, компенсация, ссылка, сигнал, множественное.



# Спусковые крючки событий

- **Сообщение**

- Событие наступило, когда было получено или отправлено какое то сообщение.



Событие начала — поступило письмо от заказчика с подтверждением оплаты (только входящее).



- Промежуточное событие — отправлен отчет (исходящее). Получена спецификация заказа (входящее)



- Событие окончания — отправлено подтверждение отгрузки товара клиенту (только исходящее)

- **Таймер**

- Событие, обозначающее время. Это может быть конкретное время или дата — 10:00 А может быть интервал — каждый понедельник. При наступлении определенного времени процесс начнется.



- Пример — событие «последний рабочий день месяца» запускает процесс «инвентаризация». Таймер не может быть событием окончания. Это противоречит сути процессного подхода — каждый процесс производит продукт. А если процесс может быть завершен просто по времени, то это дает возможность не получить продукт в итоге.





- Сигнал об ошибке. Такой тип события всегда связан с другим процессом. Т.е. в каком то процессе возникла ошибка, которая запускает последующий процесс. Пример — событие «ошибка соединения», запускает процесс проверки интернет соединения с компьютером.
- Событие начала — «отсутствуют сопроводительные документы», является ошибкой в процессе «отгрузка товара», но запускает процесс «Повторно подготовить сопроводительные документы».
- Промежуточное событие — «клиент не подтвердил бронирование номера» направляет процесс по другой ветке и после этого события следует операция «связаться с клиентом для подтверждения брони».
- Событие окончания — из примера со стартовым событием, «отсутствуют сопроводительные документы», завершает процесс отгрузки товара.

- **Эскалация**

- Запускает процесс, который ускоряет выполнение другого процесса, если тот выполняется не правильно. Событие «Эскалация» всегда связано с другим процессом, который передает (эскалирует) выполнение процесса другому человеку или системе, а тот его принимает.



Событие начала — «на заявку клиента не был дан ответ в течении суток», запускает процесс рассмотрения заявки у руководителя отдела, вместо специалиста по обслуживанию. Событие может быть только входящее.



Промежуточное событие — может отображать тот же переход заявки от специалиста к руководителю, но в рамках одного процесса. А таком случае промежуточное событие отправки у специалиста, будет переходить в соответствующее событие приема у руководителя.



- Событие окончания — тот же пример, что и в стартовом событии, «на заявку клиента не был дан ответ в течении суток», завершает процесс и отправляет заявку другому человеку. Событие только исходящее.

- **Отмена**

- Такое событие отменяет дальнейшее выполнение процесса, при этом все, что было выполнено в процессе до этого, должно быть компенсировано. Техника и инструменты, должны быть возвращены на место, отправленные сообщения аннулированы и т.д.



- Промежуточное событие — в случае промежуточного события, процесс направляется по другой ветке. Например после события отмены клиентского заказа, необходимо еще выполнить ряд действий, связанных с указанием причин отмены.
- Событие окончания — «никто не пришел», отменяет подготовку вечеринки, и все что было для нее сделано.
- Событие отмены не может быть стартовым



- **Компенсация**
- Понятие компенсации означает, что в случае наступления события компенсации в процессе, для его завершения, должны быть выполнены дополнительные действия.
- Событие начала — запускает процесс, который должен компенсировать обстоятельства, возникшие в другом процессе. Такое событие тоже всегда связано с другим процессом. Пример — событие начала «проверить сборку узла», запускает соответствующий процесс и компенсирует процесс «сборка узла».
- Промежуточное событие, обозначающее что тесто слишком жидкое, является исходящим событием компенсации, т.к. запускает операцию «добавить еще муку». После выполнения компенсирующей операции, процесс «приготовить тесто» будет завершен. Если добавление муки выполняет другая роль в процессе, то в одной роли это будет исходящее событие, а в другое — входящее.
- Событие окончания — «клиент просит скидку за пределами лимита», заканчивает процесс продаж у менеджера и запускает процесс «рассмотрение клиентской заявки» у сотрудника, ответственного за подобные решения.

- **Состояние**

- Событие, обозначающее некое состояние объекта. Состояние относится к параметрам. Например — стартовое событие «температура процессора превысила 70 градусов», запускает процесс охлаждения процессора. Или — стартовое событие, обозначающее состояние вашей второй половины как «плохое настроение», запускает процесс приготовления его/ее любимого блюда. Тоже касается и промежуточных событий — состояние какого то объекта, определяет ветку развития процесса.
- Событие окончания не может иметь триггер состояния.



- **Сигнал**

- Событие сообщает о поступлении или отправке какого то сигнала. Сигнал похож на сообщение, но разница заключается в том, что сообщение содержит информацию, а сигнал нет.



- Событие начала — свисток чайника, это сигнал, который запускает процесс заваривания чая.



- Промежуточное событие — поднятый флажок проводника, передает сигнал машинисту о готовности к отправке. Для машиниста, соответственно, сигнал является входящим.



- Событие окончания — выключение света на сцене, передает сигнал об окончании выступления и завершает процесс.

- **Множественное**

- Событие предполагает множество вариантов, но для его наступления, достаточно хотя бы одного. Или какой то одной комбинации событий. Бывают такие случаи, когда нет необходимости конкретизировать каждое событие и можно их объединить в группу. В таком случае достаточно просто перечислить — какие варианты наступления события могут быть.



Событие начала — процесс начнется, если клиент позвонит по телефону, придёт в офис или обратится через форму связи в интернет. Все это отображается одним значком.

Список событий задается или в названии события, или с помощью текстовой аннотации, или в описании элемента диаграммы. Событие начала может быть только входящим.

- Тоже касается и промежуточного и события окончания. Единственный нюанс — промежуточное событие может быть как исходящим, так и входящим. Событие окончания может быть только исходящим.

- **Параллельное множественное**
- Тоже самое, что и множественное, но событие произойдет только если будут выполнены все заданные условия. Например, мы начнем процесс подготовки к лыжной прогулке, в том случае, если сейчас зима, на улице есть снег и температура составляет не выше 5 градусов.





- **Ссылка**

- Используется для связки процессов или частей. Если событие окончания одного процесса, ссылается на другой процесс, то это же событие является началом другого процесса. Только типы событий разные.



- Пример — событие «ужин закончился», которое является завершающим событием процесса «ужин», ссылается на процесс «помыть посуду» и запускает его. Промежуточные процессы могут ссылаться как на внешние процессы, так и на процессы внутри себя.
- Ссылки могут быть как входящими, так и исходящими.

- **Терминация**

- Событие окончания, которое говорит о том, что выполнение всех операций в процессе, должно быть немедленно завершено. Самый простой пример — срабатывание пожарной сигнализации в здании, является событием терминации, потому что надо все бросить и эвакуироваться.





- **Прерывающее и не прерывающее события**
- Ну и последнее, что надо знать о свойствах событий. Каждое событие, в нотации BPMN 2.0 может прерывать или не прерывать процесс. Это относится только к событиям начала и промежуточным событиям. Прерывание, означает что наступление события необходимо для продолжения процесса. Если же событие не влияет на ход процесса, не определяет или не изменяет его, то оно просто учитывается. Например автоматическая фиксация какого-нибудь KPI, не влияет на процесс, но тем не менее может быть отображена на схеме. Также, не прерывающие событиями можно отображать совокупность условий, которые на данный момент не влияют на процесс, но будучи выполненными вместе, собираются в Множественное или Множественное параллельное событие, которое уже влияет на процесс. Примеры подобных ситуаций я соберу в отдельной статье.

# Развилки BPMN 2.0

- Шлюз, основанный на данных (XOR)
- Развилка подобного типа означает, что дальнейшее развитие процесса зависит от определенных данных или решений. При этом процесс может развиваться только по одному пути развития событий. Самый простой пример — любой вопрос, ответ на который может быть да или нет. На улице идет дождь? Если да, то берем зонт. Если нет, то не берем.





- **Шлюз, основанный на событиях (XOR)**
- Как и в предыдущем варианте, процесс может развиваться только по одному пути, но теперь он определяется на основании событий. Т.е. в зависимости от того, какое событие произойдет дальше процесс и будет развиваться. Например — если клиент перезвонил в оговоренный срок, то дальше просто происходит беседа с ним. Если же звонка не было, то менеджеру самому необходимо связаться с клиентом.



- **Включающий шлюз (решение, объединение, OR)**
- Данная развилка используется когда необходимо показать, что процесс может развиваться в результате наступления нескольких событий. Например, вы можете пойти гулять с собакой в случае если настало время прогулки, если собака просится на улицу или если оба эти события наступили. Т.е. наступление одного события и, как следствие, развитие процесса, не исключает наступление другого события. Процесс запускается при любом сочетании событий.



- **Комплексное решение, объединение**
- В нотации BPMN 2.0, во всех случаях множественных решений, требуется заранее заданное условие, чтобы развилка сработала. Почему то в голову пришел вот такой пример — для того чтобы сработало закливание, его необходимо произнести (выполнить процесс), только в полнолуние, в апреле високосного года, или при убывающей луне 29 февраля. Т.е. вы сможете пройти развилку и выполнить операцию, только если выполнен один из наборов условий.

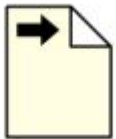


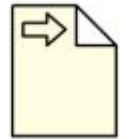
- **Параллельная развилка, объединение (AND)**
- Здесь все просто. Для того чтобы процесс стал развиваться дальше, необходимо выполнение всех условий, входящих в развилку. Если же из развилки выходит несколько потоков, то они развиваются параллельно. Пример — чтобы войти в дверь, закрытую на 3 замка, вам нужно открыть каждый из них. Дверь — это и есть развилка))



# Объекты данных VRMN 2.0

- **Исходящие данные**
- Отображает появление данных в результате выполнения процесса или операции. Например, в результате обслуживания клиента, появляются данные о времени обслуживания. С помощью этого элемента можно показывать фиксацию показателей эффективности.





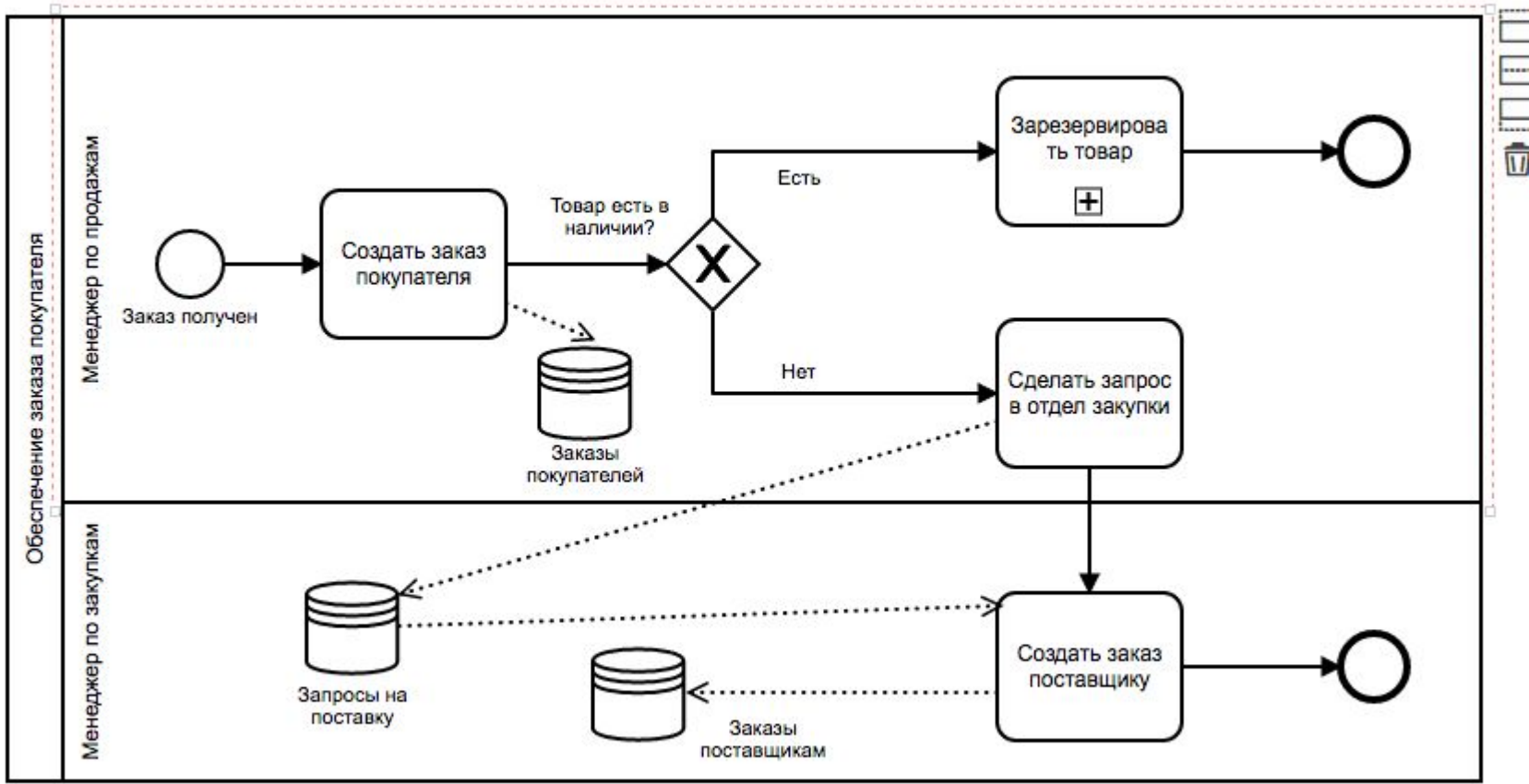
- **Входящие данные**
- Используется чтобы показать, что для выполнения процесса или операции, необходимы некоторые данные. Например, для идентификации клиента, обратившегося в кол центр, необходим номер договора.

# **Потоки VRMN 2.0**

- **Поток по умолчанию**
- Такой поток, который считается верным в процессе. Из таких потоков, складывается верный, желаемый путь всего процесса.



- **Условный поток**
- Развитие процесса происходит исходя из определенных условий. Но если вы не обозначили условия событиями или развилками, то можно воспользоваться таким элементом. Тогда в описании потока необходимо указать условия его возникновения.



- [https://www.elma-bpm.ru/journal/index.php?ELEMENT\\_ID=2900](https://www.elma-bpm.ru/journal/index.php?ELEMENT_ID=2900)