

Настройка схемы обработки инцидентов

Инструменты раздела [Ситуационный центр](#) (вкладка **Обработка инцидентов**) позволяют задать схему обработки инцидентов оператором [АРМ НЕЙРОСС Центр](#). Вы можете выполнить загрузку готовой схемы обработки или сформировать схему непосредственно из веб-интерфейса с помощью встроенного редактора.

Переход к функционалу: [Рабочий стол НЕЙРОСС](#) > [Конфигурация узлов](#) > [Ситуационный центр](#) > вкладка Обработка инцидентов.

i Возможна полностью автоматическая обработка событий, например, события технической неисправности могут автоматически адресоваться службе эксплуатации. В этом случае воспользуйтесь сервисом автоматизации [[НЕЙРОСС Автоматика](#)].

☰ Содержание:

- [Общие сведения](#)
- [Создание схемы обработки](#)
 - [Загрузка готовой схемы обработки](#)
 - [Создание схемы в редакторе](#)
- [Окно редактора схем](#)
- [Готовые элементы схемы обработки](#)

Общие сведения

Платформа НЕЙРОСС обеспечивает возможность выбора схемы обработки инцидентов в зависимости от условий возникновения события и шагов оператора, выполняющего обработку, согласно заданному сценарию, с подсказкой оператору по шагам обработки и контролем действий оператора. Предусмотрена стандартная схема обработки: оператор принимает инцидент в обработку, самостоятельно изучает состояния и события элементов, просматривает «живое» видео и видеоархив, формирует фото- и видеоматериалы, выполняет управляющие команды, формирует текстовый комментарий к инциденту и завершает его обработку.

✓ При необходимости использования более сложного сценария обработки (с разветвлением и инструкциями оператору по шагам обработки) вы можете самостоятельно разработать такой сценарий или оформить заявку на разработку нестандартного многошагового алгоритма. Для работы в режиме PSIM (обработки события по алгоритму), требуется приобретение соответствующей лицензии [[Лицензирование](#) | [Платформа НЕЙРОСС](#)].

Запуск задачи обработки осуществляется при возникновении события, удовлетворяющего конкретному фильтру. Таким образом, создавая пары [фильтр + алгоритм обработки], можно распределять инциденты по операторам и назначать разные схемы обработки разным типам событий. Обработка событий может быть частично или полностью автоматизирована. К пользовательским шагам обработки относится: отметка о выполнении заданного действия, заполнение формы, добавление комментария, вынос резолюции по инциденту, возможность скачать отчёт по инциденту. К автоматическим шагам обработки относится: возможность «привязать» архивные скриншоты, выполнить произвольный SCALA-скрипт, сформировать отчёт по инциденту и отправить на заданную почту.

Порядок настройки:

1. Рисование схемы

Для рисования базовой схемы (каркаса) рекомендуется использовать встроенный редактор BPMN-схем (с версии 20.8 Платформы НЕЙРОСС).

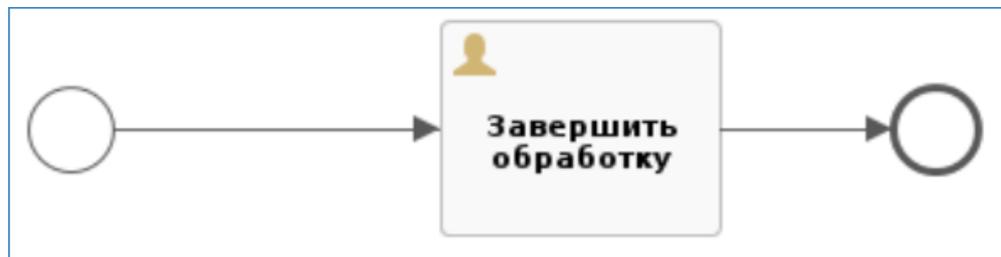
2. Дополнение схемы

Для всех автоматических и пользовательских шагов с помощью атрибутов настройте параметры шага. Для всех операторов ветвления нужно задать условия ветвления. Также необходимо указать уникальный идентификатор процесса.

3. Добавление фильтра

Требуется задать условия запуска процесса обработки по заданному сценарию — фильтр отбора целевых событий.

Каждая схема должна содержать **Стартовое событие**, которое инициализирует запуск процесса в нотации BPMN изображается в виде круга, а также **Конечное событие** — изображается в виде «жирного» круга. Для упрощения действий администратора по созданию сценария обработки, в Платформе НЕЙРОСС при создании алгоритма ему автоматически присваивается типовая схема, содержащая обязательные шаги и шаг ручного завершения оператором процедуры обработки инцидента.



Вся логика работы (ход) процесса выражается во всевозможных элементах, расположенных между Стартовым и Конечным событием. Достаточно добавить типовые шаги обработки и настроить их параметры.

Ситуационный центр

Планы Карта Обработка инцидентов Управление инцидентами Источники тревог

Алгоритмы обработки +

Стандартная схема обработки

Обработчик 1 ✕

Обработчик 2

События АвтоКПП

Сложная схема

Основные настройки

Идентификатор: custom

Название: Обработчик 1

Фильтр по событиям: Тревоги ОТС и СКУД

[Настроить фильтры]

Описание схемы [Выбрать файл] [Открыть редактор]

Сохранить

Пример схемы с автоматическим формированием фотоматериалов и выгрузкой отчёта в формате PDF

Ситуационный центр

Планы Карта Обработка инцидентов Управление инцидентами Источники тревог

Алгоритмы обработки +

Стандартная схема обработки

Обработчик 1

Обработчик 2

События АвтоКПП ✕

Сложная схема

Основные настройки

Идентификатор: AutoKPP

Название: События АвтоКПП

Фильтр по событиям: Тревоги АвтоКПП

[Настроить фильтры]

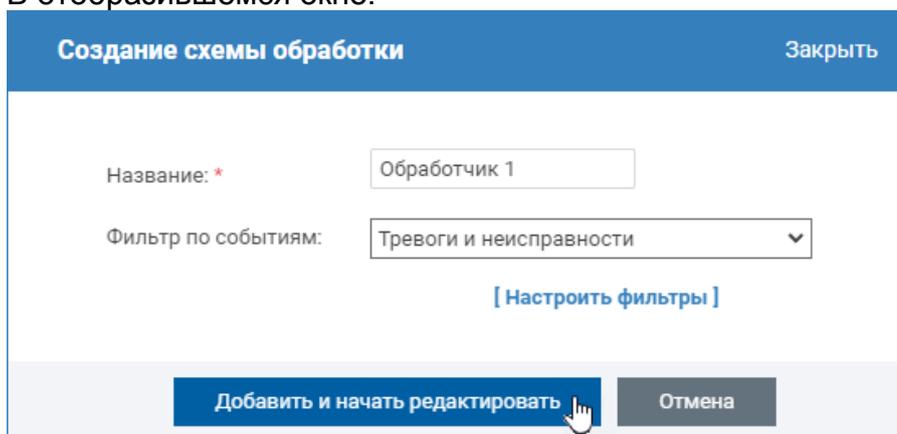
Описание схемы [Выбрать файл] [Открыть редактор]

Сохранить

Пример сложной разветвлённой схемы обработки инцидентов АвтоКПП

Создание схемы обработки

1. В блоке **Алгоритмы обработки** нажмите на кнопку  для добавления нового алгоритма.
2. В отобразившемся окне:



- a. В поле **Название** введите произвольное текстовое название нового алгоритма.
- b. В поле **Фильтр по событиям** из раскрывающегося списка выберите фильтр событий.

 Фильтр событий позволяет разделить все входящие события на несколько потоков и для каждого потока в отдельности задать свой алгоритм обработки. Фильтры настраиваются в разделе [Фильтры](#). Для перехода к разделу вы также можете воспользоваться кнопкой **Настроить фильтры** [\[Настроить фильтры\]](#).

- c. Нажмите на кнопку **Добавить и начать редактировать**. Будет создан новый алгоритм со стандартной схемой обработки.
3. Загрузите готовую схему обработки или создайте её с помощью редактора
 - a. [Загрузить готовую схему обработки](#)
 - b. [Создать схему с помощью редактора](#)

Загрузка готовой схемы обработки

Вы можете загрузить готовый XML-файл схемы, а также вставить исходный код с помощью текстового редактора.

Загрузка файла

1. В поле **Описание схемы** нажмите на кнопку **Выбрать файл** и укажите файл XML-описания алгоритма обработки. Будет загружен графический вид схемы.
2. Нажмите на кнопку **Сохранить**.

Загрузка исходного кода

1. Скопируйте исходный код схемы в буфер обмена.
2. В поле **Описание схемы** нажмите на кнопку **Открыть редактор**.
3. На вкладке **Текстовый редактор** вставьте код схемы из буфера обмена.
4. Нажмите на кнопку **Сохранить**.

i Подробная информация по работе с редактором схем обработки приведена ниже.

Создание схемы в редакторе

1. В поле **Описание схемы** нажмите на кнопку **Открыть редактор**.

Описание схемы [Выбрать файл] [Открыть редактор]



2. В отобразившемся **окне редактора схем**:
 - a. Сформируйте каркас схемы обработки [3] при помощи готовых элементов, доступных на панели элементов [2], задайте последовательность шагов. Рекомендуется использовать базовую схему с шагами запуска и завершения инцидента и дополнить её готовыми или произвольными шагами обработки. Для обеспечения «ветвления» алгоритма — выбор «ветки» в зависимости от заданных оператором условий — предусмотрен оператор ветвления.

i Перечень готовых элементов схемы обработки

[Готовые элементы схемы обработки](#)

- b. На панели свойств [5] измените, при необходимости, уникальный идентификатор каждого шага и потока, укажите параметры.
- c. Перейдите к вкладке **Текстовый редактор** [1] задайте дополнительные параметры элементов, условия ветвления.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <definitions xmlns:ultima="urn:ru:itrium:ultima:1.0" xmlns:activiti="http://activiti.org/bpmn" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://activiti.org/bpmn http://activiti.org/bpmn.xsd" >
3   <signal id="confirmedAssignmentSignal" name="Подтверждено назначение инцидента оператору" activiti:scope="processInstance"/>
4   <process id="Scheme" name="Default" processType="None" isClosed="false" isExecutable="true">
5     <startEvent id="startevent1" name="Start"/>
6     <endEvent id="endevent1" name="End"/>
7     <userTask id="finishProcessing" name="Завершить обработку" activiti:assignee="Incident:${incidentId}" activiti:exclusive="true">
8       <extensionElements>
9         <ultima:bpmnElementExtension class="extensions.vmc.incident.FinishProcessingIncidentTask"/>
10      </extensionElements>
11    </userTask>
12    <sequenceFlow id="flow1" sourceRef="startevent1" targetRef="Screenshots"/>
13    <sequenceFlow id="5" sourceRef="finishProcessing" targetRef="endevent1"/>
14    <serviceTask id="Screenshots" name="Привязать архивные скриншоты" activiti:class="extensions.vmc.incident.BindArchiveSnapshots">
15      <extensionElements>
16        <activiti:field stringValue="-5,0,5" name="relativeTimestamps"/>
17        <activiti:field name="incidentId" expression="${incidentId}"/>
18      </extensionElements>
19    </serviceTask>
20    <sequenceFlow id="flow2" sourceRef="Screenshots" targetRef="view_all_around"/>
21    <userTask id="view_all_around" name="Осуществить обзор территории" activiti:assignee="Incident:${incidentId}" activiti:exclusive="true">
22      <extensionElements>
23        <ultima:bpmnElementExtension class="extensions.vmc.incident.SimpleIncidentTask"/>
24      </extensionElements>
25    </userTask>
26    <sequenceFlow id="flow3" sourceRef="view_all_around" targetRef="fill_form"/>
27    <userTask id="fill_form" name="Заполнить форму" activiti:assignee="Incident:${incidentId}" activiti:exclusive="true">
28      <extensionElements>
29        <ultima:bpmnElementExtension class="extensions.vmc.incident.FillFormIncidentTask"/>
30      </extensionElements>
31    </userTask>
32  </process>
33 </definitions>
```

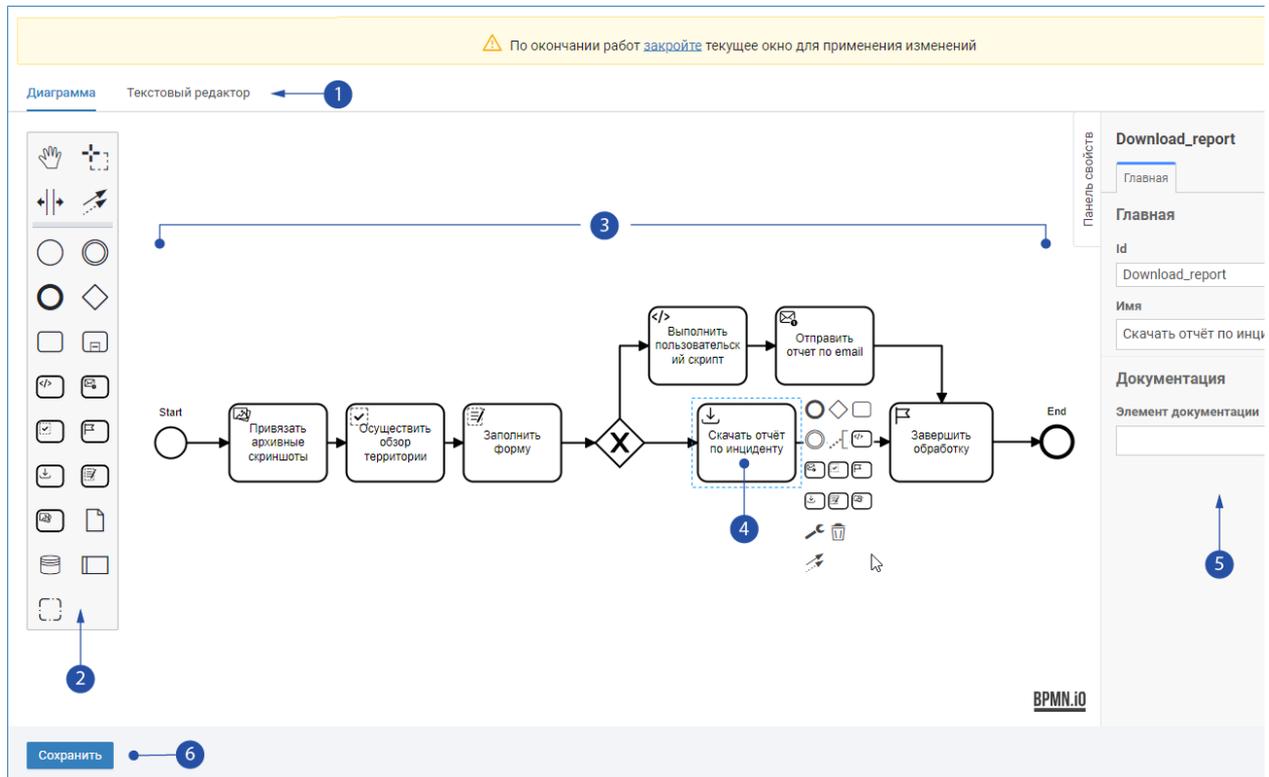
- d. Нажмите на кнопку **Сохранить**.

✓ В процессе сохранения выполняется валидация алгоритма. Если найдены ошибки, сохранение не выполняется. Проверьте алгоритм и внесите необходимые изменения, повторите попытку сохранения.

е. Закройте окно редактора схем.

⚠ По окончании работ [закройте](#) текущее окно для применения изменений

Окно редактора схем



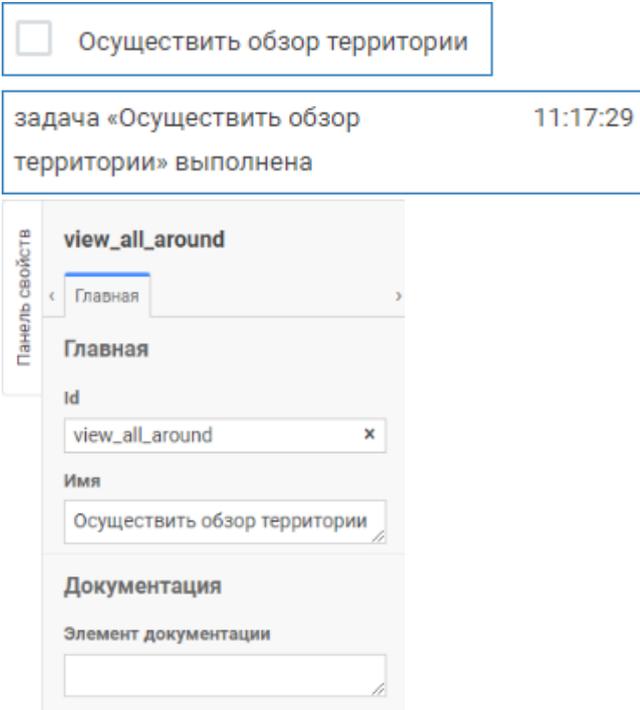
1. **Вкладки Диаграмма и Текстовый редактор:** вкладка **Диаграмма** предоставляет графический редактор для формирования каркаса схемы, вкладка **Текстовый редактор** предоставляет доступ к коду xml-схемы с подсветкой синтаксиса, позволяет задать параметры шагов схемы.
2. **Панель элементов:** [готовые элементы схемы обработки](#); достаточно выбрать требуемый элемент и нанести на схему.
3. **Схема обработки:** визуальное представление схемы с указанием последовательности обработки шага; формируется автоматически в процессе добавления новых элементов; каждому элементу и потоку присваивается уникальный идентификатор, который вы можете изменить в поле id панели свойств [5].
4. **Элемент схемы обработки:** при выделении элемента отображаются доступные инструменты по изменению типа элемента, добавлением нового потока и проч.
5. **Панель свойств:** содержит базовые свойства выделенного на схеме [3] элемента [[Готовые элементы схемы обработки](#)].
6. Кнопка **Сохранить:** сохранение схемы, в процессе сохранения выполняется валидация.

Готовые элементы схемы обработки

Платформа НЕЙРОСС предоставляет готовые элементы для построения схемы обработки инцидентов:

- пользовательские шаги: отметка о выполнении действия, заполнение формы;
- автоматические шаги: добавление медиаматериалов, формирование отчёта по инциденту и отправка его на электронную почту, выполнение произвольного скрипта.

Поддержаны операторы условного ветвления.

Пользовательские шаги User Task			
№	Шаг / Панель свойств	Описание	Пример XML
1	<p>Отметить выполнение действия</p> 	<p>Пользовательский произвольный шаг обработки, факт выполнения которого оператор фиксирует установкой флага в соответствующем поле. Задаёт алгоритм действий оператора: подсказывает последовательность шагов и контролирует их выполнение.</p> <p>Инструкция оператору задаётся в поле name.</p> <p>В карточке инцидента отображается в виде чек-бокса с текстом. Оператор знакомится с текстом инструкции, выполняет требуемые действия. По факту завершения устанавливает флаг в поле. Система фиксирует время выполнения шага и передает управление следующему шагу.</p>	<pre><userTask id="view_all_around" name="Осуществить обзор территории" assignee="\${incident.assignee}" incident="\${incident.id}" exclusive="true"> <extension base="http://bpmn.io/bpmn/extensions/extension" class="external:incident.Similarity" /> </userTask></pre>
2	Завершить обработку / пометить инцидент ложным	Завершает обработку инцидента. Выносит резолюцию	<pre><userTask id="finishProcess" name="Завершить обработку" /></pre>

✓ ЗАВЕРШИТЬ
🗑 ЛОЖНЫЙ

задача «Завершить обработку»
11:54:23

выполнена (инцидент подтверждён)

Панель свойств

finishProcessing

Главная

Главная

Id

Имя

«Ложный» / «Подтверждённый» для инцидента.

В карточке инцидента отображается в виде виде двух кнопок: **Завершить** и **Ложный**.

Шаг включён в стандартную схему обработки.

```

activiti:assign
incident:${in
exclusive="t
<exten
<ulti
bpmnEleme
class="exten
incident.
FinishProce
/>
<
/extensionE
</userTask>

```

3 Заполнить произвольную форму

Заполнить форму

Текст:

Да/Нет: *

Число:

Дата:

Выбор из списка:

Первое значение
Первое значение
Второе значение
Третье значение
Четвёртое значение

ОТПРАВИТЬ

задача «Заполнить форму» выполнена
11:50:46

Панель свойств

fill_form

Главная

Главная

Id

Имя

Позволяет оператору заполнить и отправить произвольную форму из полей типа string, long, boolean, date, enum и selectvalue.

Возможно указание списка полей, обязательных к заполнению, значений по умолчанию для полей, текстовых подсказок оператору.

В карточке инцидента отображается в виде заголовка, списка полей формы и кнопки **Отправить**.

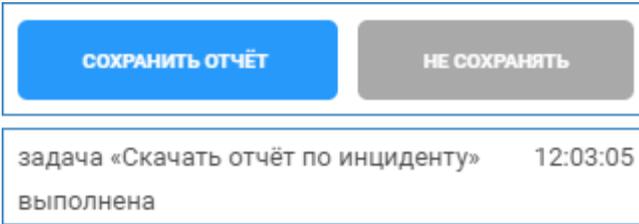
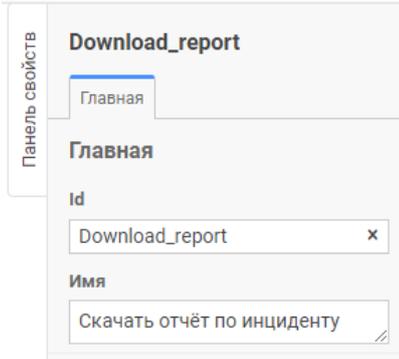
Заголовок формы задаётся в поле name для шага обработки. Параметры полей указываются отдельно: идентификатор (id), тип поля (type), заголовок поля (name), флаг обязательности (required), значение по умолчанию (default), флаг

```

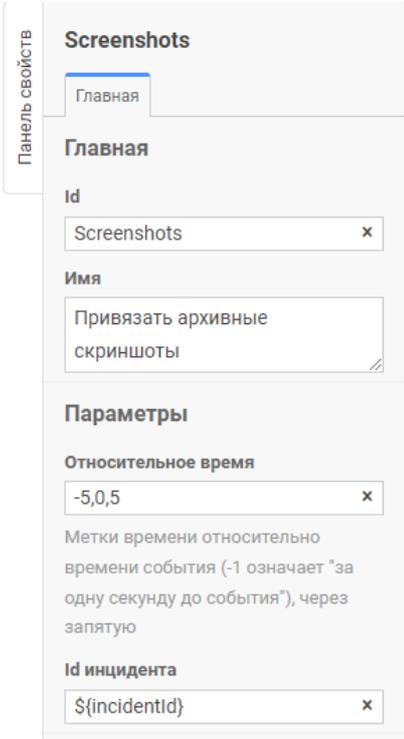
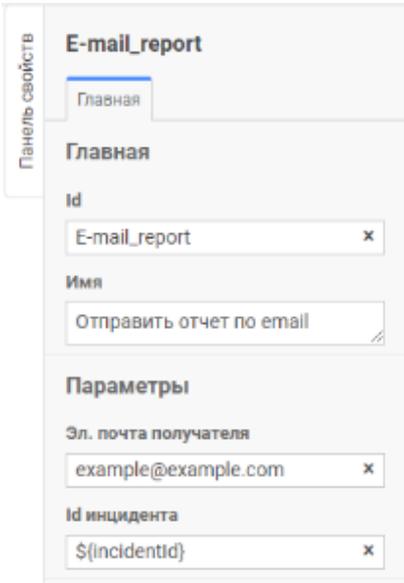
<userTask i
name="Зап
activiti:assign
incident:${in
exclusive="t
<exten
<
bpmnEleme
class="exten
incident.
UserFormIn
<
formPropert
name="Тек
default="12
<
formPropert
booleanValu
/Нет" type='
required="tr
<
formPropert
name="Чис.
longValue"
число"/>
<
formPropert
name="Дат
<
formPropert
name="Выл
type="enum

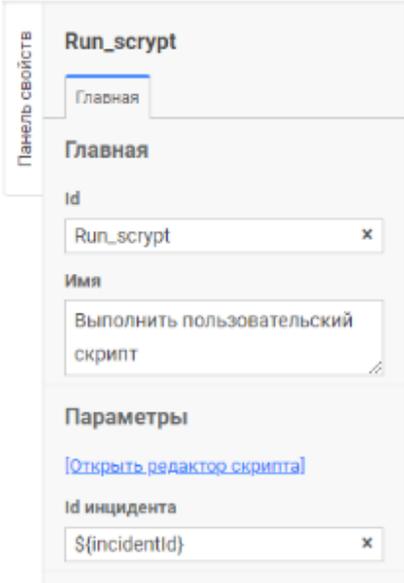
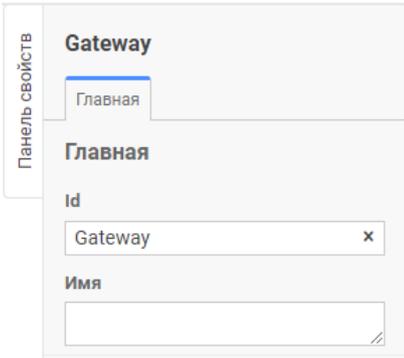
```

8

		<p>«только для чтения» (readonly).</p> <p>Система фиксирует время выполнения шага и передает управление следующему шагу.</p> <p>Все введённые значения доступны при обработке следующих шагов, в частности, — в зависимости от введённого оператором значения возможен выбор ветки алгоритма.</p>	<pre>id="value1" значение"/> id="value12 значение"/> id="value13 значение"/> id="value14 Четвёртое : formPropert < /extensionE </userTask></pre>
4	<p>Запустить процесс формирования PDF-отчета об инциденте и сохранения его в папку загрузок браузера</p>  	<p>Позволяет оператору скачать pdf-отчёт об инциденте.</p> <p>В XML-описании указывается: ключ отчёта, идентификатор источника данных и форма, которая будет использована при формировании отчёта.</p> <p>В карточке инцидента отображается в виде двух кнопок Сохранить отчёт и Не сохранять. На момент подготовки отчёта кнопки не активны. В случае возникновения ошибки создания отчёта отображается кнопка Ошибка.</p> <p>Система фиксирует время выполнения шага и передает управление следующему шагу.</p>	<pre><userTask id="Download_report" name="Скачать отчёт по инциденту" incident:\${incidentId} exclusive="true" <extension base="bpmn:ExternalTaskDefinition" class="externalTaskDefinition:DownloadReportDefinition" defaultVm="vmc-incident-reportDefinition" defaultVmC="vmc-incident-report" defaultVmCWritable="false" readable="true" dataSources="{}" default="1" <activity name="INCIDENT_DOWNLOAD_REPORT" default="\${incidentId}" writable="false" </extension> </userTask></pre>

Автоматические шаги | Service Task

1	<p>«Привязать» архивные скриншоты</p> 	<p>Добавляет в карточку инцидента скриншоты из архива «привязанных» к источнику инцидента камер. Действий пользователя не требуется. Скриншоты добавляются в отчёт по инциденту.</p> <p>В поле Относительное время (параметр relativeTimestamps в коде) через запятую задаются сдвиги относительно времени возникновения инцидента, за которые нужно «привязать» скриншоты.</p> <p>Необходимо обеспечить связь камер с источником тревог [«Привязка камер источникам тревог»] и наличие видеoarхива за требуемый период.</p>	<pre><serviceTask name="Screenshots" class="com.siemens.cti.screenshots.bindarchive.BindArchive" relativeTimeExpression="<activity:fileTime -5,0,5" name="Screenshots" relativeTimeExpression="<activity:fileTime incidentId" relativeTimeExpression="</extension" /></serviceTask"></pre>
2	<p>Отправить отчёт по e-mail</p> 	<p>Формирует отчёт об инциденте и отправляет его на заданную почту. Отчёт формируется в формате pdf.</p> <p>Необходимо настроить параметры почтового сервера в разделе Дополнительные настройки. Требуется наличие лицензий [НЕЙРОСС Автоматика].</p>	<pre><serviceTask name="mail_report" class="com.siemens.cti.mail_report.SendReport" relativeTimeExpression="<activity:fileTime incidentId" relativeTimeExpression="</extension" /></serviceTask"></pre>

		<p>E-mail-адрес получателя указывается в параметрах шага.</p>	<pre><stringValue= vmc-incident reportDefini string"/> </exter </serviceTa</pre>
3		<p>Запускает на выполнение произвольный scala-скрипт. Специальные символы в скрипте необходимо экранировать, согласно стандарту XML. Переносы строки в коде нужно заменить на ";"</p>	<pre><serviceTask name="Вып пользовате activiti:class incident.Run <exten <stringValue= common.{S SystemLog systemLogS injector.insta [SystemLog systemLogS (SystemLog ("TPEBOGA type="string < incidentId" expression= </exter </serviceTa</pre>
Оператор условного ветвления			
1	<p>«Разветвить» алгоритм</p> 	<p>Позволяет в зависимости от значения текущих переменных в рамках обработки инцидента выбрать тот или иной следующий шаг.</p> <p>На текущий момент в качестве переменных можно использовать только значения, введённые оператором в</p>	<pre><userTask i Предварите activiti:assign incident:\${in exclusive="{t <exten < bpmnEleme class="exter incident. UserFormIn <</pre>

рамках ранее
завершённых шагов
заполнения формы

```
formPropert
type="boole
Нарушител
</exter
</userTask>
<exclusiveC
Gateway"/>
<sequenceF
Flow_toForr
startevent1"
/>
<sequenceF
Flow_toGat
Form" targe
<sequenceF
flow_Gatew
Инцидент п
sourceRef='
targetRef='
>
    <condi
type="tForm
    ${
true}
</cond
</sequence
<sequenceF
flow_Gatew
Ложная тре
Gateway" ta
finishProces
    <condi
type="tForm
    ${
true}
</cond
</sequence
```