

## Лабораторная работа №2. Моделирование процесса Командировка.

**Бизнес-процесс** — это регулярно повторяющаяся последовательность взаимосвязанных действий, направленных на создание определённого продукта или услуги для потребителей. При этом потребителями могут быть не только внешние заказчики, но и внутренние.

### Постановка задачи

Смоделируем бизнес-процесс «Командировка», который устанавливает четкий порядок действий сотрудников при организации командировок – от момента подачи заявления на командировку до получения отчета о поездке руководителем.

Сотрудник создает элемент приложения *Командировка*, заполняет все необходимые данные, после сохранения заявки запускается процесс оформления командировки.

Далее *Согласующий руководитель* должен согласовать или отказать в командировке. Если в командировке отказано, то командируемый сотрудник получает оповещение в системе с комментарием отказа.

Если заявка на командировку согласована, то командируемый сотрудник также получает уведомление в системе о том, что командировка согласована. Также в приложении *График отпусков* будет создано событие, в котором указывается какой сотрудник и в какие даты, по какой причине будет отсутствовать.

В том случае, если сотрудник в заявке указал необходимость получения наличных, то после согласования заявки будет запущен процесс *Получение денежных средств*.

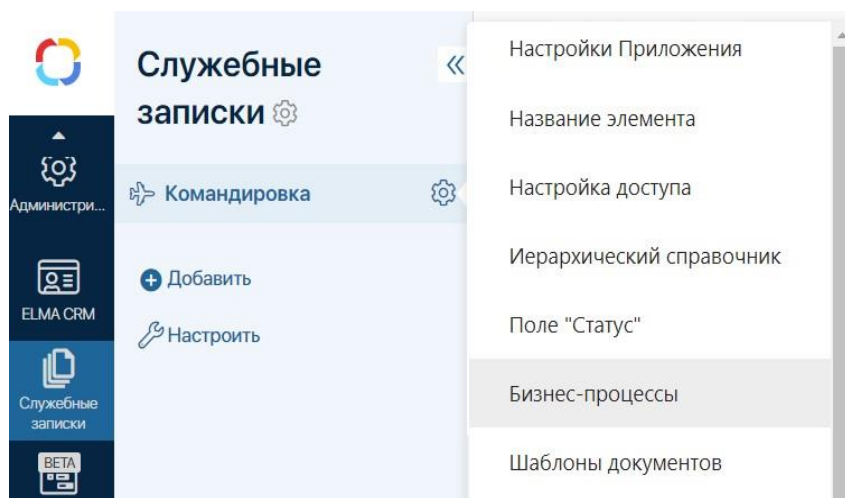
Если в заявке было указано, что необходимо купить билеты, то следующим шагом *Офис - менеджеру* приходит задача, в которой указана информацию кому, куда, на какие даты должны быть куплены билеты. После чего билеты в электронном формате отправляются на электронную почту командируемому сотруднику.

Далее необходимо приостановить процесс до даты окончания командировки, для этого используется *Таймер*.

После окончания командировки командируемому сотруднику поступает задача – Отчитаться о командировке, сотрудник заполняет отчет и отправляет его на проверку в бухгалтерию. Когда *Бухгалтер* принимает отчет, командировка считается завершенной. Процесс завершается.

## Создание бизнес-процесса

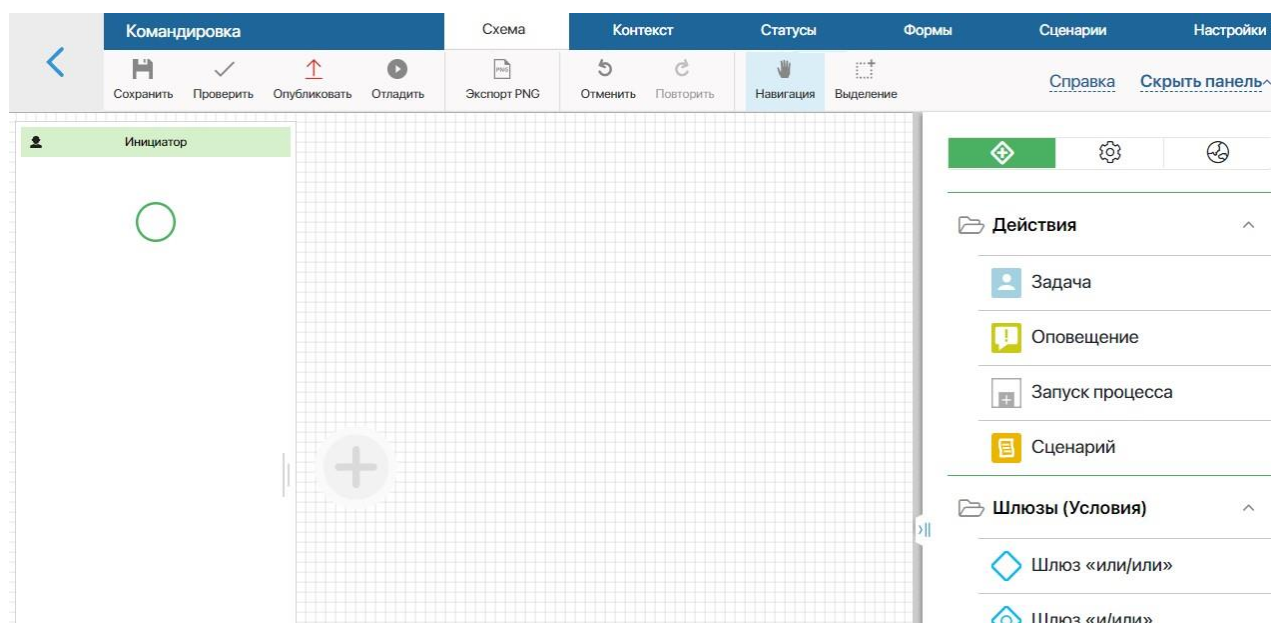
Для того, чтобы добавить процесс следует перейти в настройки приложения Командировка – Бизнес-процессы. В правом верхнем углу нажать **+Процесс**.



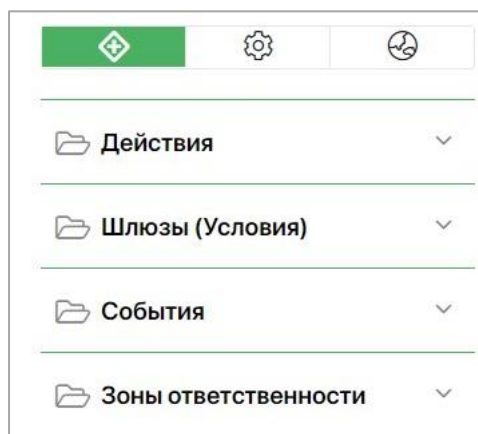
В открывшемся окне вводим название процесса и нажимаем кнопку Создать.

Итак, можно переходить к моделированию процесса.

После создания процесса мы видим на экране «пустой» процесс, где создана зона ответственности Инициатора и установлено лишь стартовое событие.



Следующий шаг – создание цепочки операций.



В правой части страницы расположена панель с графическими элементами для описания процесса.

Панель содержит в виде списка все доступные операции системы. Операции разбиты на несколько вкладок и групп по смыслу их работы.

### Настройка зон ответственности

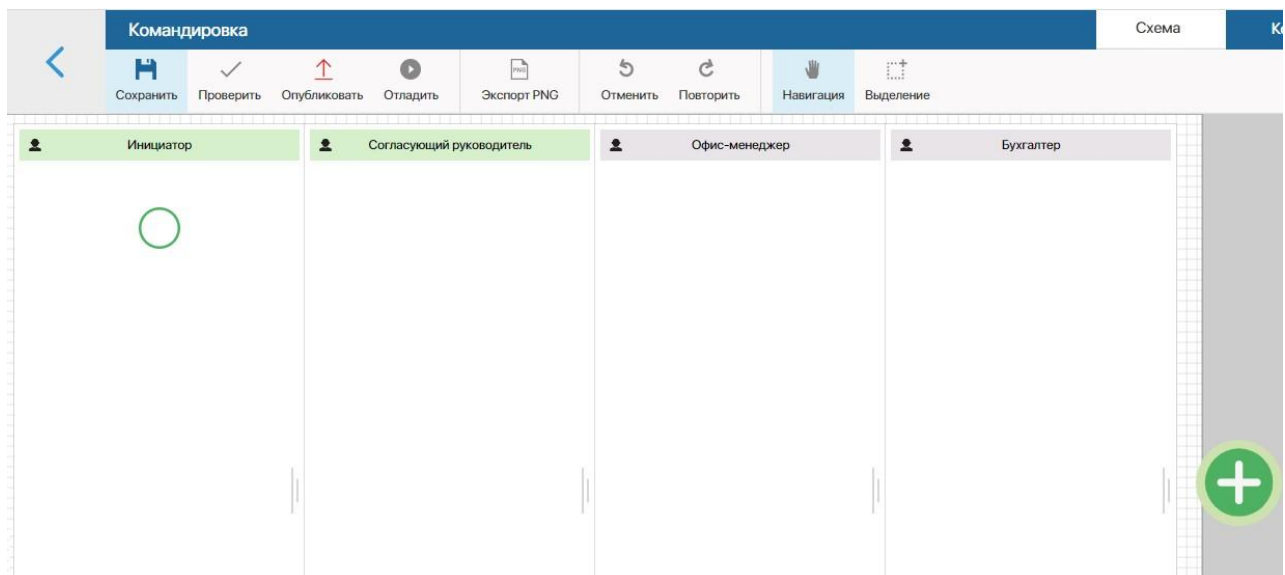
Для начала добавим необходимые Зоны ответственности.

**Зона ответственности** используется для обозначения участника бизнес-процесса. Это может быть определённый сотрудник организации, отдел, покупатель, поставщик или роль, например, инициатор или согласующий. Внутри зоны ответственности располагаются задачи и события, за выполнение которых отвечает данный участник процесса. Зона ответственности «Инициатор» ставится на карту процесса автоматически.

Чтобы определить участников бизнес-процесса, вы можете использовать один из следующих типов зон ответственности:

- **Динамическая** — используется в том случае, если должность участника определяется в ходе исполнения процесса. На схеме обозначается зеленым цветом;
- **Статическая** — выбирайте этот тип, если должность по оргструктуре или группа/роль ответственного известна до начала процесса. На схеме обозначается серым цветом.

Для того, чтобы добавить Зону ответственности следует нажать на значок «плюс» на поле для моделирования процесса.



Так, например, Зона ответственности Согласующий руководитель будет Динамическая.

The form is titled 'Согласующий руководитель' and has two tabs: 'Основное' (Main) and 'Права доступа' (Access Rights). The 'Основное' tab is active. It contains the following fields:

- Название\*** (Name): 'Согласующий руководитель'
- Тип** (Type): 'Статическая' (Static) and 'Динамическая' (Dynamic) buttons. 'Динамическая' is selected.
- Переменная** (Variable): 'Руководитель' (Manager) with a dropdown arrow. A button 'Изменить настройки переменной' (Change variable settings) is next to it.

At the bottom are two buttons: 'Сохранить' (Save) and 'Закрыть' (Close).

А Зона ответственности Бухгалтера – Статическая и привязана к должности в оргструктуре.

The form is titled 'Бухгалтер' and has two tabs: 'Основное' (Main) and 'Права доступа' (Access Rights). The 'Основное' tab is active. It contains the following fields:

- Название\*** (Name): 'Бухгалтер'
- Тип** (Type): 'Статическая' (Static) and 'Динамическая' (Dynamic) buttons. 'Статическая' is selected.
- Переменная** (Variable): 'Выберите переменную' (Select variable) with a dropdown arrow. A button 'Создать новую переменную' (Create new variable) is next to it.
- Группа пользователей** (User group): 'Группа или роль' (Group or role) and 'Элемент оргструктуры' (Organizational structure element) buttons. 'Элемент оргструктуры' is selected.

Below these fields is a tree view showing the organizational structure:

- Бухгалтер
  - Генеральный директор
    - Бухгалтерия
      - Главный бухгалтер
        - Бухгалтер
  - Офис-менеджер
  - Отдел внедрения

At the bottom are two buttons: 'Сохранить' (Save) and 'Закрыть' (Close).

Далее сформируем контекст процесса.

**Контекст процесса** — это все данные, которые создаются, изменяются и сохраняются во время выполнения конкретного процесса. Эти данные хранятся в переменных, содержание (значение) которых может изменяться в ходе процесса. Например, при каждом запуске смоделированного процесса могут быть разные сроки командировки.

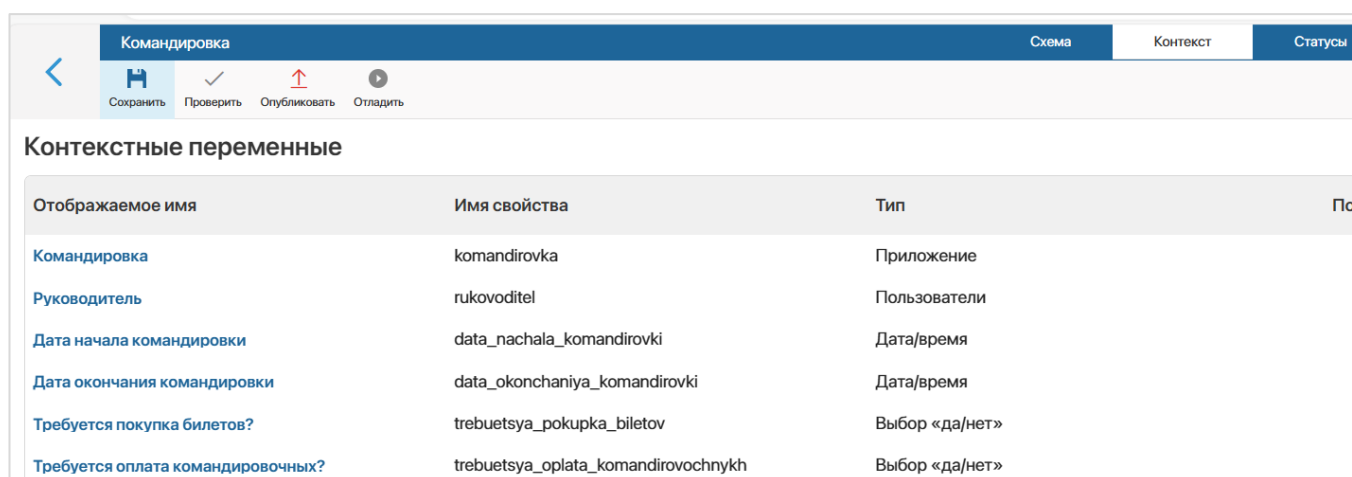
Список всех контекстных переменных находится в карточке процесса на вкладке **Контекст**. Там уже присутствует контекстная переменная **Командировка** типа **Приложение**, которая хранит всю информацию, которую мы ввели на форме создания командировки, и переменная **Руководитель**, которую мы создали, когда настраивали Зоны ответственности. Добавим в контекст процесса переменные:

Дата начала командировки – тип Дата/Время;

Дата окончания командировки – тип Дата/Время;

Требуется покупка билетов? – тип Выбор «да/нет»;

Требуется оплата командировочных? – тип Выбор «да/нет».



Отображаемое имя	Имя свойства	Тип	По
Командировка	komandirovka	Приложение	
Руководитель	rukovoditel	Пользователи	
Дата начала командировки	data_nachala_komandirovki	Дата/время	
Дата окончания командировки	data_okonchaniya_komandirovki	Дата/время	
Требуется покупка билетов?	trebuetsya_pokupka_biletov	Выбор «да/нет»	
Требуется оплата командировочных?	trebuetsya_oplata_komandirovochnykh	Выбор «да/нет»	

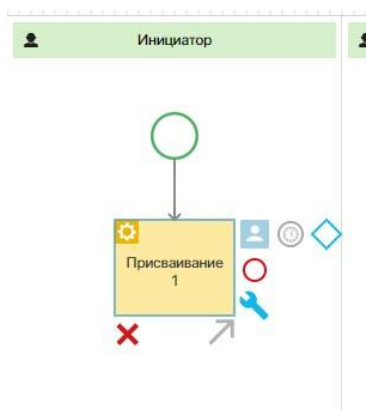
Сохраните внесенные изменения.

Далее переходим к моделированию схемы бизнес – процесса.

В зоне ответственности Инициатора уже присутствует стартовое событие (зеленый круг). Стартовое событие инициирует запуск процесса и указывает, в какой точке он берет начало. Процесс может содержать только одно стартовое событие.

Первым шагом добавим блок Присваивание – используется для того, чтобы автоматически передавать данные в бизнес-процесс и устанавливать соответствие между полями приложения и контекстными переменными процесса. Просто при помощи мыши перетащите значок операции Присваивание с панели операций на свободную область графической модели в зону ответственности инициатора. По

умолчанию операция будет названа автоматически сгенерированным именем, которое впоследствии можно изменить.



Двойной клик по блоку открывает окно его настроек. Назовем данный блок «Передача данных из контекста приложения» и установим соответствие как представлено на рисунке:

Передача данных из контекста приложения

Основное

Таблица соответствия

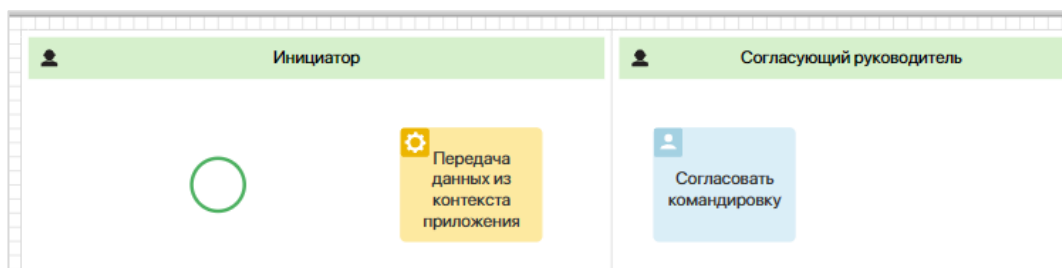
Поля контекста	Значения
Требуется покупка билетов?	<- Командировка . Требуется покупка билетов?
Дата начала командировки	<- Командировка . Дата начала
Требуется оплата командировочных?	<- Командировка . Требуется наличные?
Дата окончания командировки	<- Командировка . Дата окончания
Руководитель	<- Командировка . Согласующий руководитель

+ Добавить

Присваивание используем для того, чтобы в процессе могли использовать переменные приложения. Например, в условиях на шлюзе можно указывать только контекстные переменные процесса.

Подобным образом вы можете добавить сколько угодно необходимых операций на графическую модель процесса. При помощи мыши вы также можете свободно перемещать уже существующие операции по графической модели с места на место.

Следующим шагом будет задача Согласование командировки руководителем. Добавим блок «Задача» в зону ответственности согласующего руководителя и назовем ее «Согласовать командировку». Чтобы перейти к настройке операции, просто дважды щелкните по блоку операции на графической модели.



Для того, чтобы исполнитель задачи получал более детально описание задачи следует формировать название по шаблону. Информацию введенную в этом поле, исполнитель увидит в списке задач или на карточке задачи. В шаблон можно включить контекстные переменные, для этого нажмите на значок {+} в правом углу поля Шаблон.

Согласовать командировку

×

Основное

Форма

Время выполнения

Планирование в календаре

Переходы

Название\*

Согласовать командировку

Название задачи

☒ Формировать название по шаблону

Шаблон\*

Согласовать {\$business\_trip} {+}f{×}

Оповещение

☒ Оповещать исполнителя о постановке задачи

Переназначение

☒ Разрешить переназначать задачу на другого исполнителя

Сохранить

Закреть

Для некоторых задач время выполнения — критичный показатель. В нашем случае, заявка на командировку должна быть согласована до начала самой командировки. Поэтому на вкладке Время выполнения требуется настроить сроки выполнения задачи, а также определить, что произойдет, если в указанное время задача сделана не будет. В таком случае, если заявка не будет согласована в указанный срок инициатору процесса придет оповещение в ленту сообщений о том, что задача согласования просрочена.

**Согласовать командировку** ✕

Основное   **Форма**   Время выполнения   Планирование в календаре   Переходы

Время выполнения ☒ Ограничивать срок выполнения задачи

Тип ограничения ☐ Точное время ☒ Переменная

Срок выполнения  Изменить настройки переменной

Корректировать время ☐ Не корректировать ☐ Добавить ☒ Вычитать

Дни:  Часы:  Минуты:  ☒ С учётом рабочего календаря

Оповещение ☒ Отправить сообщение в ленту указанным пользователям

Добавить Очистить

Прервать ☐ По истечению срока перейти к следующему шагу

Сохранить Закрыть

## Смена статусов в бизнес-процессе

Для того, чтобы пользователи понимали на каком этапе в данный момент находится Командировка, необходимо добавить статусы. Статусы можно настроить двумя способами:

1. В настройках приложения Командировка – Статус.

Служебные записки

Командировка

Добавить

Настроить

☒ Добавить поле "Статус"

☐ Сохранять историю изменения

☐ Статус Приложения можно менять вручную

Статус	Код	Финальный	Скрывать
Создана	created	<input type="checkbox"/>	<input type="checkbox"/>
Согласована	agreed	<input type="checkbox"/>	<input type="checkbox"/>
Запрос денежных средств	money_request	<input type="checkbox"/>	<input type="checkbox"/>
Получение денежных средств	money_receiving	<input type="checkbox"/>	<input type="checkbox"/>
Денежные средства выданы	money_issued	<input type="checkbox"/>	<input type="checkbox"/>
Текущая	current	<input type="checkbox"/>	<input type="checkbox"/>
Завершена	completed	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Отказано	denied	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Добавить

Сохранить Отмена

2. В бизнес-процессе на вкладке Статусы.



Командировка

Схема

Контекст

Статусы

Сохранить

Проверить

Опубликовать

Отладить

### Статусы

☒ Добавить поле "Статус"
 ☐ Сохранять историю изменения
 ☐ Статус можно менять вручную

Статус	Код	Финальный	Скрывать
Создана	created	<input type="checkbox"/>	<input type="checkbox"/>
Согласована	agreed	<input type="checkbox"/>	<input type="checkbox"/>
Запрос денежных средств	money_request	<input type="checkbox"/>	<input type="checkbox"/>
Получение денежных средств	money_receiving	<input type="checkbox"/>	<input type="checkbox"/>
Денежные средства выданы	money_issued	<input type="checkbox"/>	<input type="checkbox"/>
Текущая	current	<input type="checkbox"/>	<input type="checkbox"/>
Завершена	completed	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Отказано	denied	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Добавить

Сохранить

Стоит отметить, что в системе ELMA365 статусы есть только у приложения, бизнес- процесс не имеет своих статусов. Поэтому в процессе отображаются статусы приложения, с которым связан процесс.

После того, как будут добавлены статусы, то в графической модели бизнес-процесса также следует добавить смену статуса приложения с помощью блока Управление статусом.

В окне настройки указываем переменную, у которой будет меняться статус, сам статус, который будет установлен при прохождении данного этапа.

Сменить статус 1

×

Основное

Название\*

Командировка -> Согласована

Переменная\*

Командировка

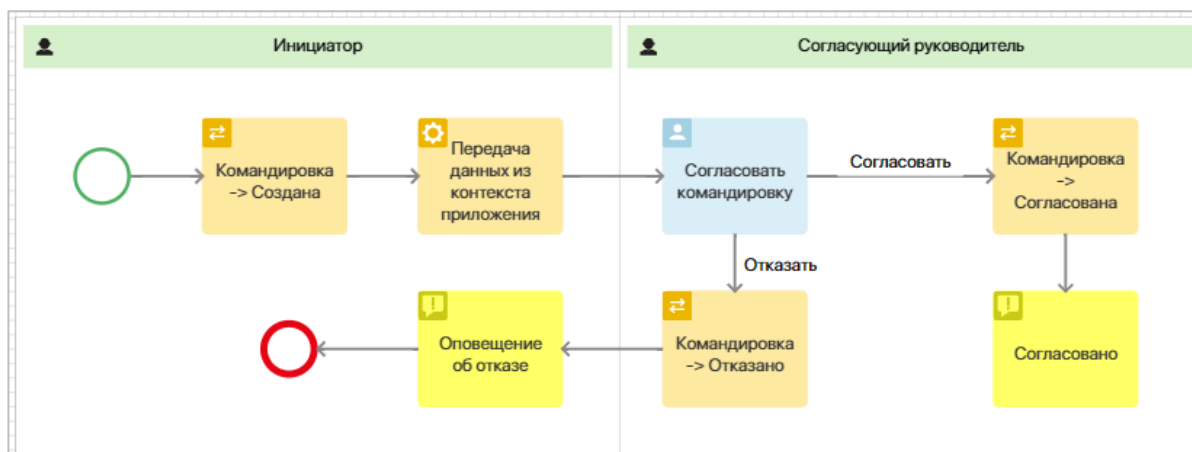
Измeнить настройки переменной

Установить статус\*

Согласована

Сохранить

Закрyть



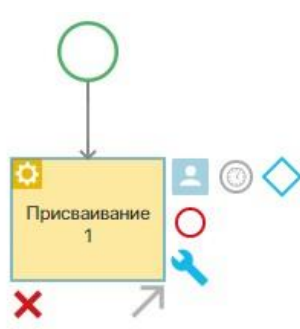
После того, как операции процесса размещены на графической модели, следует соединить их переходами – эти линии связи позволят указать направление хода выполнения бизнес-процесса.

Для соединения двух операций следует подвести указатель мыши к краю первой операции, около указателя мыши появится небольшой красный квадрат, обозначающий точку выхода из операции. Теперь нажмите и удерживайте левую кнопку мыши.

Переместите указатель мыши к краю второй операции, около указателя мыши появится крестик, обозначающий точку входа в операцию. Отпустите левую кнопку мыши. Линия перехода будет прочерчена. О направлении перехода позволяет судить стрелка на ее конце.

Таким образом, мы соединяем элементы на графической модели между собой. Некоторые операции могут иметь более одного входа и более одного выхода. Все элементы на графической модели, которые можно соединить между собой, соединяются схожим образом. На графической модели не должно остаться «висячих» операций, которые не имеют достаточного количества входов и выходов.

Существует еще один способ добавления операций на графическую модель процесса, позволяющий автоматически устанавливать связи между добавляемыми операциями: щелкните мышью по операции на графической модели процесса, от которой требуется создать следующую операцию.



Щелчок мышью по операции позволяет выбрать ее. За пределами объекта можно увидеть несколько небольших значков, которые представляют собой миниатюры операций, таких как Пользовательская задача, Шлюз, Промежуточное событие, Конечное событие.

Если при помощи мыши перетащить такой значок на нужное место на графической модели процесса, автоматически создастся новая операция, уже соединенную с родительской. Подобный механизм вы уже могли видеть при моделировании организационной структуры.

В случае Отказа в командировке руководителю необходимо заполнить комментарий с причиной отказа, для этого лучше всего добавить контекстную переменную с комментарием на форму перехода (не задачи!). Для этого перейдем на вкладку «Переходы» в настройках задачи, кликнем на переход Отказаться, в открывшейся форме выбрать цвет кнопки и вид подтверждения – форма. После этого станет доступна вкладка «Настройка формы». На этой вкладке выносим ту переменную, которую необходимо будет заполнить на переходе. В нашем случае это поле «Комментарий руководителя», который мы выбираем в качестве вложенного поля контекстной переменной «Командировка»

Отказаться

Основное Кнопка перехода Настройка формы

Цвет кнопки перехода: Красный

Проверки при переходе: ☒ Проверять заполнение обязательных полей задачи ☐ Проверять наличие незакрытых подзадач

Вид перехода: ☐ Без подтверждения ☐ Простое подтверждение ☒ Форма

Размер окна подтверждения: Средний

Пояснение к форме: Укажите причину отказа

Настроить форму

Сохранить Заккрыть

Отказаться

Основное Кнопка перехода Настройка формы

+ Добавить → Перенести

Контекст	Название на форме	Свойство
Инициатор	Командировка	Командировка
Название	Комментарий руководителя	Комментарий руководителя
Комментарий		
Требуется покупка билетов?		
бухгалтер		
Командировка		

Сохранить Заккрыть

Для того, чтобы командированный сотрудник узнал о том, что в командировке ему отказано, необходимо настроить оповещение. Блок оповещения настраиваем следующим образом: указываем тему, текст сообщения (эти данные увидит пользователь), а также автора и получателя. Автором будет Согласующий руководитель, а получателем Командированный сотрудник – переменная процесса Инициатор. Для того, чтобы оповещение сохранялось в ленте сообщений Командировки, требуется ассоциировать её с контекстной переменной «Командировка».

Оповещение об отказе

Основное

Получатели

Название\*

Оповещение об отказе

Тема сообщения\*

В командировке {\$business\_trip} - отказано. (+)f(x)

Текст сообщения

{\$business\_trip.comment} (+)f(x)

Автор

Текущий пользователь

● Контекстная переменная

Система

Руководитель

Сообщение будет отправлено от имени пользователя, заданного в контекстной переменной

Ассоциированный объект

● Текущий процесс

○ Контекстная переменная

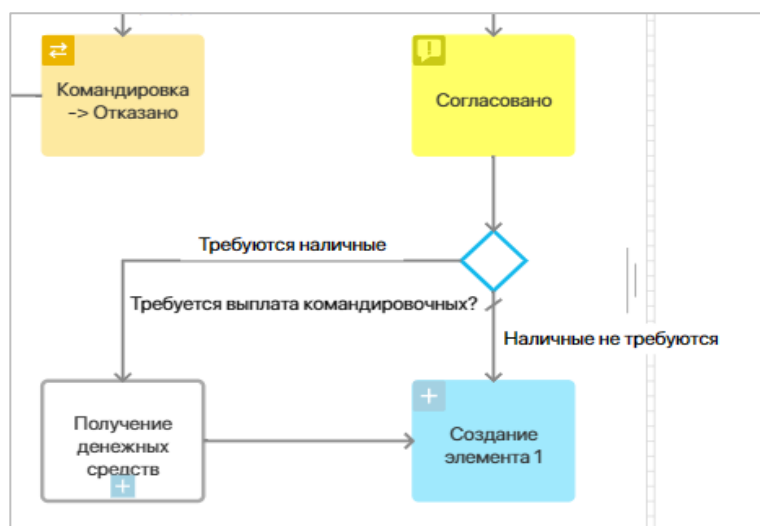
Сохранить

Заккрыть

Подобным образом настраивается оповещение о согласовании командировки.

После согласования Руководителем процесс может пойти по двум направлениям: на выдачу наличных или на создание отсутствия и покупку билетов.

Для этого необходимо использовать исключающий шлюз «или/или», в котором будут следующие выходы: Требуются наличные, Наличные не требуются. Кроме названия следует настроить условия перехода, за выполнение условий будет отвечать контекстная переменная, которую заполнял командированный сотрудник при оформлении заявки на командировку. Один из переходов должен быть отмечен как переход по умолчанию – его условие можно не заполнять, он сработает только в том случае, если не сработают условия остальных переходов шлюза.



Требуется выплата командировочных?

Основное

Служебная переменная

Переходы

Название

По умолчанию

Требуется наличные

Наличные не требуются

Условия в настройках переходов проверяются в порядке, приведенном выше. Для того, чтобы изменить последовательность проверки условий, переместите их, используя метод Drag-and-Drop.

Сохранить

Заккрыть

Требуется наличные

Основное

Условия перехода

Связка

Операнд 1

Операция

Операнд 2

Требуется оплата командировочных?

=

Да

Нет

+ Условие

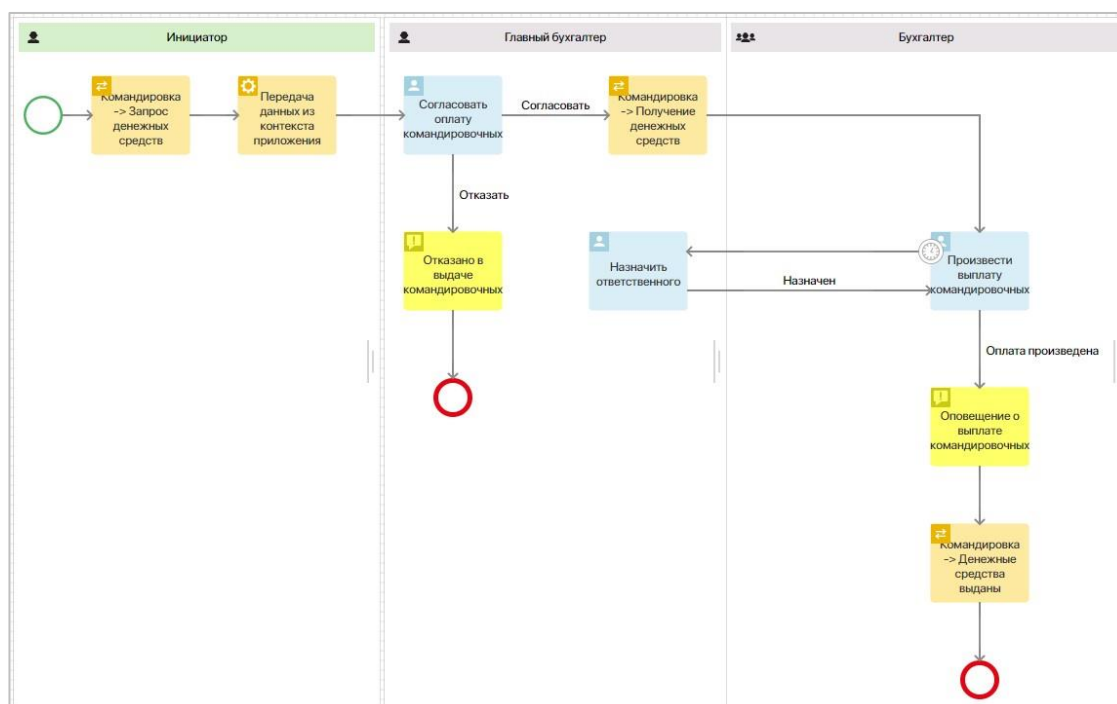
Сохранить

Заккрыть

В случае если требуется получить денежные средства, необходимо запустить процесс Получение денежных средств, для этого используется блок Запуск процесса.

Блок **Запуск процесса** используется для декомпозиции и повышения наглядности схем. Он позволяет вызывать отдельный подпроцесс в рамках более крупного родительского бизнес-процесса.

На первом этапе создадим процесс Получения денежных средств. Ниже представлена карта процесса.



Произвести выплату наличных должен бухгалтер. Т.к. на должности бухгалтера может быть несколько сотрудников, не имеет значения кто именно будет заниматься выплатой наличных, главное, чтобы задача была выполнена в срок, для этого необходимо настроить задачу следующим образом:

- Устанавливаем множественное выполнение – Кто первый, это значит, что задача придет каждому бухгалтеру, и тот, кто менее загружен в данный момент, сможет сразу приступить к выполнению. Как только система получит подтверждение, что работа начата (для этого нужно нажать на кнопку в карточке задачи), она отменит задачу для других сотрудников.

The screenshot shows the 'Основное' (Main) tab of a task configuration window titled 'Произвести выплату наличных'. The window has a close button (X) in the top right corner. Below the title bar are five tabs: 'Основное', 'Форма', 'Время выполнения', 'Планирование в календаре', and 'Переходы'. The 'Основное' tab is active. The form contains the following fields and options:

- Название\***: Text input field containing 'Произвести выплату наличных'.
- Название задачи**: Checkmark ☒ 'Формировать название по шаблону'.
- Шаблон\***: Text input field containing 'Произвести выплату наличных {business\_trip}' with a '{+}f(\*)' icon on the right.
- Оповещение**: Checkmark ☒ 'Оповещать исполнителя о постановке задачи'.
- Переназначение**: Checkmark ☒ 'Разрешить переназначать задачу на другого исполнителя'.
- Множественное исполнение**: Three radio buttons: 'Кто первый' (selected), 'Параллельное', and 'Последовательное'.

At the bottom are two buttons: 'Сохранить' (Save) and 'Закрыть' (Close).

- Ограничить задачу по времени выполнения и указать, что задачу необходимо прервать и перейти на следующий шаг – на задачу Главного бухгалтера, для назначения конкретного исполнителя задачи

The screenshot shows the 'Время выполнения' (Execution Time) tab of the same task configuration window. The 'Время выполнения' tab is active. The form contains the following fields and options:

- Время выполнения**: Checkmark ☒ 'Ограничивать срок выполнения задачи'.
- Тип ограничения**: Two radio buttons: 'Точное время' and 'Переменная' (selected).
- Срок выполнения**: A dropdown menu showing 'Дата начала командировки' and a button 'Изменить настройки переменной'.
- Корректировать время**: Three radio buttons: 'Не корректировать', 'Добавить', and 'Вычитать' (selected).
- Дни**: Input field with '0'.
- Часы**: Input field with '10'.
- Минуты**: Input field with '0'.
- С учётом рабочего календаря**: Checkmark ☐.
- Оповещение**: Checkmark ☐ 'Отправить сообщение в ленту указанным пользователям'.
- Прервать**: Checkmark ☒ 'По истечению срока перейти к следующему шагу'.
- Next Step**: A dropdown menu showing '-> Назначить ответственного'.

At the bottom are two buttons: 'Сохранить' (Save) and 'Закрыть' (Close).

После того, как смоделировали процесс Получение денежных средств, нужно его опубликовать. Далее необходимо привязать этот процесс в блоке процесса Командировки и указать входные/выходные атрибуты в подпроцессе.

Получение денежных средств

Основное

Входные и выходные атрибуты

Название

Получение денежных средств

Процесс

Получение денежных средств

Асинхронный запуск

☐ Продолжить выполнение текущего процесса, не дожидаясь окончания дочернего

Связать с полем

Командировка

Сохранить

Заккрыть

Получение денежных средств

Основное

Входные и выходные атрибуты

Входные переменные

Подпроцесс

Родительский

Командировка

<- Командировка

Инициатор

<- Инициатор

+ Входная переменная

Выходные переменные

Подпроцесс

Родительский

+ Выходная переменная

Сохранить

Заккрыть

Следующим этапом будет создание записи в Графике отсуствий. Для этого необходимо в разделе Служебные записки создать приложение-событие График отсуствий и добавить необходимые свойства, как показано на рисунке.

График отсуствий / Настройки формы

Сохранить

Контекст

Создание

Просмотр

Редактирование

Плитка

+ Добавить

Отображаемое имя	Имя свойства	Тип	Поиск и сортировка	Полнотекстовый поиск
Название	__name	Строка	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Дата начала	__startAt	Дата/время	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Дата окончания	__endAt	Дата/время	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Участники	__participants	Пользователи	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Командировка	business_trip	Приложение	<input type="checkbox"/>	<input type="checkbox"/>

После того, как приложение создано, мы можем создавать элементы приложения в процессе с помощью блока Создание элемента. У данного блока также есть две вкладки, на первой с помощью контекстной переменной указываем элемент какого приложения будем создавать – График отсуствий. Способ создания Автоматически – без участия пользователей.



**Создание отсутствия** [X]

Основное | Значения полей

Название\*

Переменная\*

Переменная с типом "Приложение". В нее будет записана ссылка на созданный элемент. Тип создаваемого элемента Приложения определяется в соответствии с настройками данной переменной.

Способ создания

☐ Вручную  
На текущего исполнителя будет поставлена задача с формой создания элемента Приложения. По умолчанию значения полей будут заполнены в соответствии с заданными настройками на вкладке "Значения полей".

☒ Автоматически  
Элемент будет создан автоматически. Поля будут заполнены в соответствии с заданными настройками на вкладке "Значения полей".

На вкладке Значения полей указываем те поля приложения, которые будут заполнены в ходе бизнес-процесса.

**Создание отсутствия** [X]

Основное | Значения полей

Поле Приложения	Переменная процесса
Название	<- Командировка . Название
Дата начала	<- Командировка . Дата начала командировки
Дата окончания	<- Командировка . Дата окончания командировки
Участники	<- Командировка . Командируемый сотрудник
Командировка	<- Командировка

После того, как будет создана запись в календаре об отсутствии сотрудника, необходимо поставить шлюз, который будет направлять процесс по одной из веток в зависимости от того, нужна ли покупка билетов. Если нужно купить билеты – ставится задача Офис-менеджеру о покупке билетов, если билеты не требуются, то процесс идет дальше.

В том случае, если билеты были куплены, то отправляем их командируемому сотруднику по электронной почте. Для этого используем блок Оповещение на email. Настраиваем блок следующим образом, в графе Получатель указываем электронную почту Инициатора (командируемого сотрудника), прописываем тему и текст письма – это то, что увидит пользователь, в графе Файлы – указываем переменную, в которой хранятся билеты.

Данную переменную заполняет Офис-менеджер на предыдущей задаче.



Оповещение на email

Основное

Обработка ошибок

Название\*

Оповещение на email

Получатель\*

{\$\_createdBy.email}

(+)(-)(v)

Имя отправителя

(+)(-)(v)

Адрес для ответа

admin@example.com, info@example.com

(+)(-)(v)

Тема письма\*

Билеты для {\$\_business\_trip}

(+)(-)(v)

Текст письма\*

Во вложении билеты.

(+)(-)(v)

Файлы

Билеты

+ Добавить

Сохранить

Закрыть

После того, как билеты отправлены устанавливаем таймер на дату окончания командировки.

Ожидание окончания командировки

Основное

Название\*

Ожидание окончания командировки

Тип ограничения

☐ Точное время
☒ Переменная

Срок выполнения

Дата окончания командировки

Изменить настройки переменных

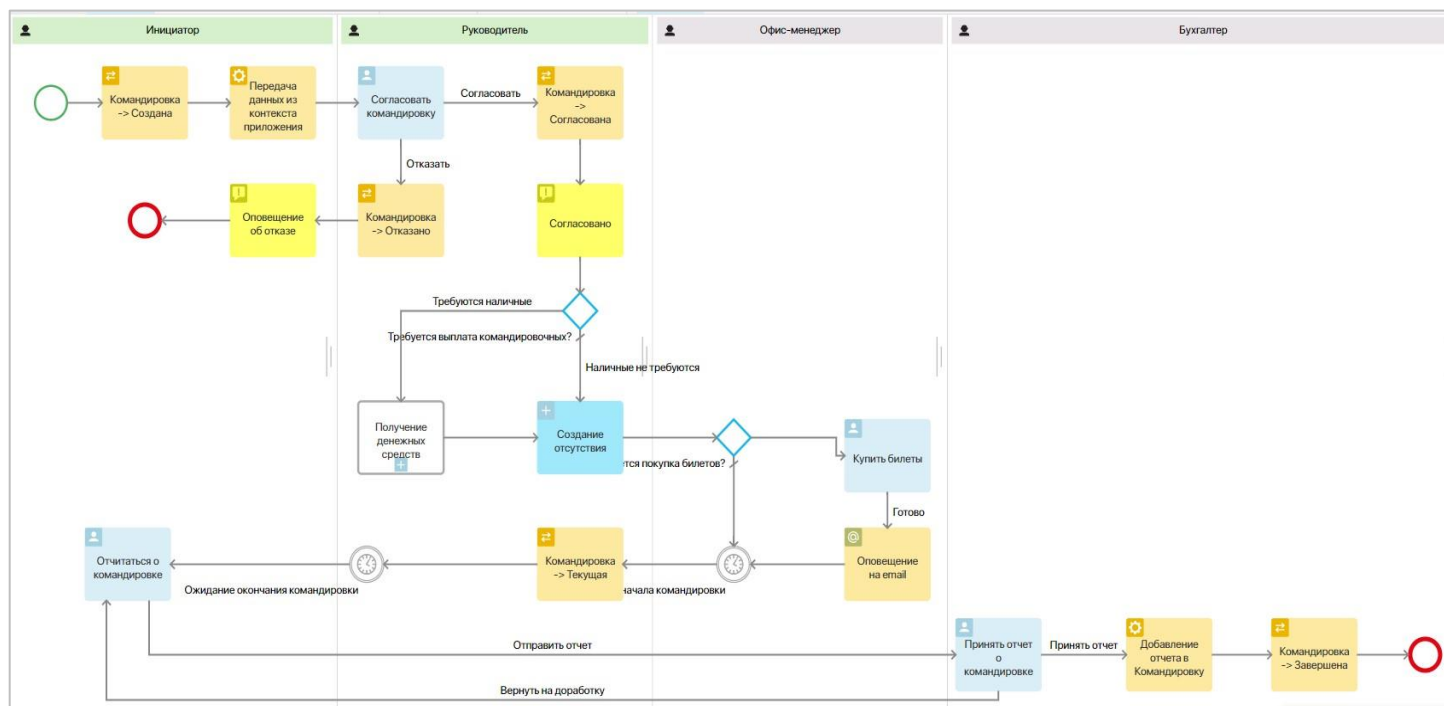
Корректировать время

☒ Не корректировать
☐ Добавить
☐ Вычесть

Сохранить

Закрыть

Настроенный процесс должен выглядеть следующим образом:



После окончания командировки Командируемому сотруднику должна поступить задача, в которой он прикладывает отчет по командировке (файл) и отправляет его на согласование бухгалтеру. Бухгалтер может согласовать или отправить на доработку. В случае отправки на доработку также необходимо указать комментарий, аналогично задаче Согласования командировки. Для этого необходимо завести поле «Комментарий проверяющего» и вынести на форму задачи «Составить отчёт по командировке» с условием скрытия, если поле не заполнено.

Если отчет принят, то его следует прикрепить к приложению Командировка с помощью блока Изменение элемента. На вкладке Основное указываем какую переменную мы будем изменять – приложение Командировка.

The screenshot shows a dialog box titled 'Добавление отчета в Командировку' with a close button (X) in the top right corner. It has two tabs: 'Основное' (selected) and 'Значения полей'. In the 'Основное' tab, there are three main sections: 'Название\*' with a text input field containing 'Добавление отчета в Командировку'; 'Переменная\*' with a dropdown menu showing 'Командировка' and a button 'Изменить настройки переменной'; and 'Способ создания' with two radio buttons: 'Вручную' (unselected) and 'Автоматически' (selected). Below the 'Автоматически' option is a blue box containing text: 'Элемент будет изменен автоматически. Поля будут заполнены в соответствии с заданными настройками на вкладке "Значения полей"'. At the bottom are two buttons: 'Сохранить' and 'Закрыть'.

На вкладке Значение полей в переменную приложения Отчет записываем контекстную переменную процесса Отчет, которая была заполнена в рамках процесса на задаче «Составить отчет по командировке».

The screenshot shows the same dialog box, but with the 'Значения полей' tab selected. It displays a table with two columns: 'Поле Приложения' and 'Переменная процесса'. The first row has 'Отчет' in the first column and '<- Отчет' in the second column, with a red 'X' icon to the right of the second cell. Below the table is a button '+ Добавить'. At the bottom are two buttons: 'Сохранить' and 'Закрыть'.

Так как в системе один бизнес-процесс может быть одновременно запущен несколько раз, каждый его экземпляр должен иметь уникальное имя, позволяющее отличить один экземпляр от другого. Старайтесь давать экземплярам процесса простые и понятные названия, чтобы избежать возможной путаницы.

Для формирования названия по шаблону необходимо перейти на вкладку Настройки.

## Проверка бизнес-процесса и его публикация

После моделирования процесса система предоставляет возможность проверить графическую модель процесса на корректность и наличие потенциальных ошибок.

Для проверки графической модели процесса используйте кнопку Проверить панели инструментов. В случае, если графическая модель составлена верно в соответствии со стандартом BPMN, система выдаст соответствующее уведомление.

В случае, если графическая модель содержит ошибки, система отметит цветной рамкой элемент, где предположительно содержится ошибка, и выведет список ошибок на панели Ошибки и предупреждения.

Желтой рамкой обводятся элементы, вызвавшие у системы подозрения – это обычное предупреждение, говорящее пользователю, что с объектом не все в порядке, но это не повлияет на ход процесса. Красной рамкой обводятся элементы, содержащие ошибки – запустить такой процесс не удастся.

После того, как моделирование процесса завершено, и процесс отложен, можно переходить к публикации процесса. Сохраненный процесс нельзя будет использовать, пока он не будет опубликован.

Публикация необходима в двух случаях:

1. Для нового процесса. Пока процесс не опубликован, с ним не смогут работать обычные пользователи.
2. При модификации существующего процесса. Публикация процесса делает доступными изменения, которые были сделаны с момента последней публикации.

## Запуск бизнес-процесса

Возможность запуска бизнес-процесса имеет пользователь, который имеет на это право. Кроме того, администратор системы может ограничить возможность запуска процесса лишь определенным кругом пользователей.

Запущенный процесс выполняется в соответствии с его графической моделью. Система будет автоматически назначать пользователям задачи, выполнять сценарии, принимать решения в шлюзах, посылать пользователям уведомления и совершать другие действия, предусмотренные при моделировании бизнес-процесса.