

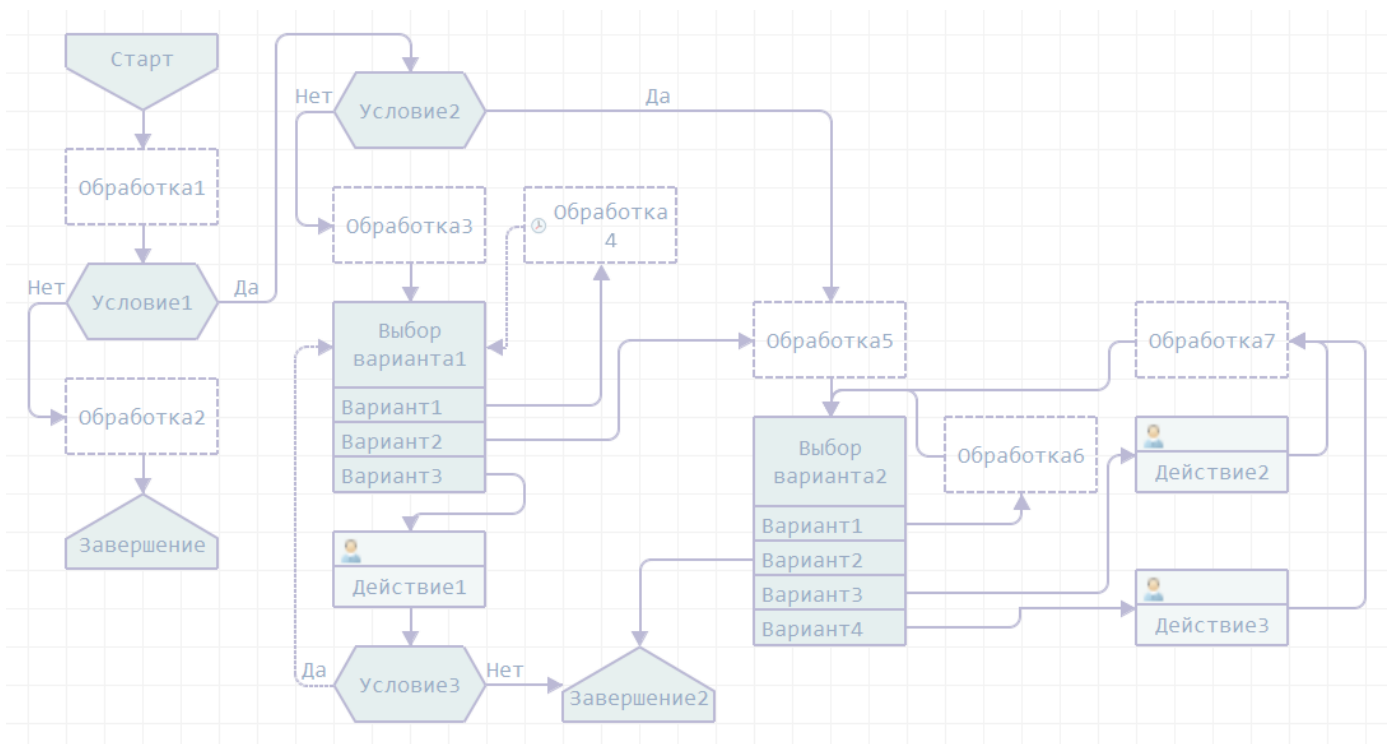


Примеры настройки сценариев Конструктор Бизнес Процессов®

- [Повтор шагов сценария](#)
- [Отправка контрагентам Telegram-оповещений](#)
- [Расшифровка результатов проверки условий](#)
- [Циклические условия](#)
- [Настройки работы с задачами в Telegram](#)
- [Выполнение программного кода в чат-боте](#)
- [Автоматическая остановка и автовозобновление задач](#)
- [Групповые задачи и условия](#)
- [Условная обязательность реквизитов задач](#)
- [Таблицы произвольных данных](#)
- [Выполнение задач по электронной почте](#)
- [Отложенная обработка](#)
- [Оформление графической схемы](#)



03

Версия основной поставки 1.4.5.70
Версия расширения «Сигнал» 1.2.2.14



Введение

Программный продукт **Конструктор Бизнес Процессов**^{®1} предназначен для настройки бизнес-процессов в любых конфигурациях 1С, работающих на технологической платформе «1С: Предприятие 8.3» в режиме управляемого приложения. Система разработана в среде «1С: Предприятие 8.3» в режиме управляемого приложения.

В настоящем документе приведены примеры настроек сценариев, приёмы, нюансы и тонкости, знание которых может оказаться полезным при настройке сценариев в системе  **Конструктор Бизнес Процессов**[®], а также при настройке автоматических оповещений с помощью расширения  «Сигнал».

Описание переменных, реквизитов, процедур и функций, которые встречаются в тексте, доступно на нашем сайте по ссылке [Описание алгоритмов, переменных и функций.pdf](#).

Дополнительные материалы (статьи, примеры, ответы на вопросы, видео) доступны на официальном сайте.

Официальный сайт:

www.1bpm.ru

Ответы на часто задаваемые вопросы:

www.bip.one/faq.html

Адрес электронной почты для вопросов и предложений:

mail@1bpm.ru

Авторы оставляют за собой право на опечатки, ошибки и технические изменения.

¹ Другие названия: «Новые Системы: Конструктор Бизнес-Процессов», «Новые Системы: Бизнес-Процессы», «Новые Системы: Мета-Бизнес-Процессы», «БИП: Конструктор Бизнес-Процессов».

Повтор шагов сценария (31 октября 2022)

В ряде случаев, по сценарию требуется возврат к ранее пройденным шагам.

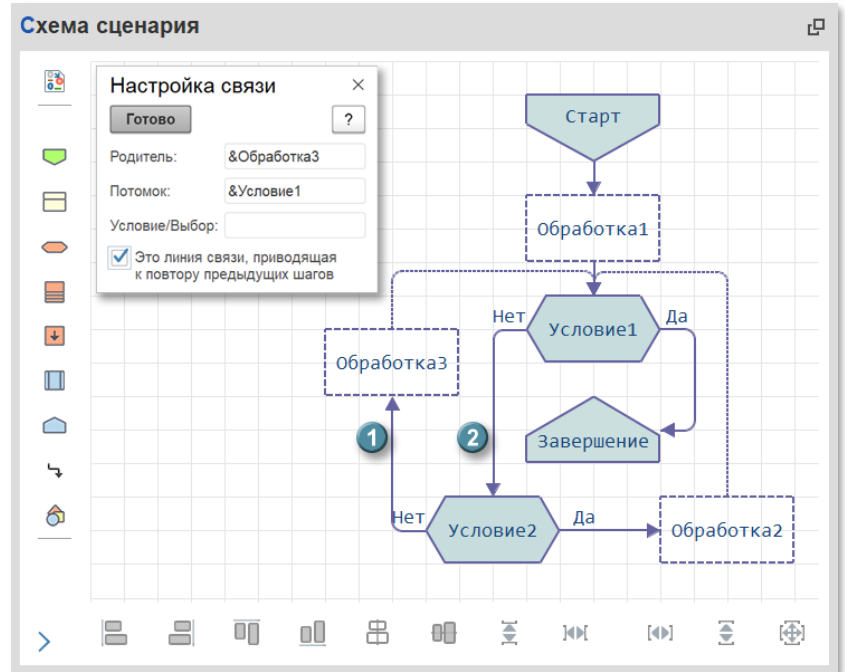
Для того, чтобы сценарий корректно обрабатывал подобные ситуации, следует соблюдать следующие условия:

1. Для линий связи, которые приводят к повтору ранее пройденных шагов, требуется устанавливать признак повтора в **настройках линии связи**.

2. Линиями повтора являются только те линии, которые непосредственно приводят к повтору, а не все линии в цепочке повторяемых шагов.

Линии повтора отображаются на схеме пунктиром

ПРИМЕР: линией повтора является только линия от **Обработка3** к **Условие1**. Линии и не являются линиями повтора, хотя, визуально и относятся к условному «блоку повторяющихся шагов».



Система умеет подсчитывать количество повторов шагов. Для этого предусмотрен реквизит шага сценария **КоличествоПовторов**.

Использование количества повторов возможно, например, в шагах вида **Условие**.

При использовании реквизита **КоличествоПовторов** следует учитывать:

1. Система рассчитана на учёт только **предыдущих** шагов. Если указать шаг, который ещё не наступил, система зафиксирует ошибку. Сценарий, при этом, может протекать некорректно.

ЗАМЕЧАНИЕ: С технической точки зрения неткаких-то особых ограничений, и система знает количество повторов любых шагов – и тех, что уже пройдены и тех, которые ещё не наступили. Но, соблюдая определенную логику построения сценариев, определение количества повторов шагов, которые ещё не наступили, не требуется. Поэтому, при настройке сценариев введено подобное ограничение.

2. Повтором **не является** первое прохождение шага. Т.е. если система прошла шаг сценария первый раз, а затем вернулась на него повторно, то значение реквизита **КоличествоПовторов** будет равно 1.

Отправка контрагентам Telegram-оповещений (31 октября 2022)

Система позволяет настроить отправку оповещений в Telegram не только пользователям системы, но и **Партнерам** и/или **Контрагентам**.

Для этого, при настройке оповещений по сценарию, можно использовать **алгоритм**, в котором следует указать **1 получателя** или **массив получателей**.

| Настройки оповещений | | | |
|-------------------------|--|--|--|
| Этап сценария | Вид события | Тип оповещения | |
| Подзадача/Условие/Выбор | Получатель | Сообщение | |
| Оповестить клиента | При выполнении шага | Telegram | |
| | Контрагент - получатель Telegram-оповещения... | <i>Сообщение для контрагента [_Процесс.Объект]</i> | |

☆ Контрагент - получатель Telegram-оповещения (Алгоритмы)

Записать и закрыть _Результат = <...> ? Проверить ?

Условие 21

```
_Результат = Новый Массив;  
_Результат.Добавить(_Процесс.Объект);  
//_Результат.Добавить(Справочники.Пользователи.НайтиПоНаименованию("Иванов"));  
//_Результат.Добавить(Справочники.Пользователи.НайтиПоНаименованию("Белкина"));  
//_Результат.Добавить(Справочники.Пользователи.НайтиПоНаименованию("Федоров"))
```

Имя для вызова: ?

Комментарий:

Алгоритм Telegram

ПРИМЕР: В данном примере получателем будет контрагент, который является основным объектом процесса (`_Процесс.Объект`).

Контрагент получит уведомление, если он авторизован в чат-боте Telegram, через который происходит отправка оповещений.

Подробнее об интеграции с Telegram см. [Интеграция с Telegram](#) и [Конструктор чат-ботов](#).

Сообщение для контрагента Савинков Олег Петрович 0:01

Расшифровка результатов проверки условий (31 октября 2022)

Если по сценарию в задаче указан **Алгоритм автоматического завершения**, то этот же алгоритм проверяет **возможность ручного завершения задачи**.

При невозможности выполнения задачи пользователю будет выведено соответствующее сообщение.

При этом не всегда этого сообщения бывает достаточно и требуется более детальная информация о причинах, по которым задача не может быть завершена.

Для этого в системе предусмотрен следующий порядок информирования:

1. По умолчанию, в текст сообщения выводится **Наименование алгоритма**, который проверяет возможность завершения задачи.
2. Если в алгоритме заполнен **Комментарий**, то в сообщении будет показан комментарий, а не наименование алгоритма.
3. Если в коде алгоритма было присвоено значение переменной **_РасшифровкаРезультата**, то значение этой переменной будет показано в окне сообщения.

ПРИМЕР: В настройках задачи указан алгоритм автоматического завершения задачи.

Ниже приводится **3 варианта** вывода сообщения о невозможности завершения задачи по причине того, что условие завершения не выполнено.

Автоматически завершать задачу по условию ?
Сегодня понедельник

☆ Сегодня понедельник (Алгоритмы)

Записать и закрыть | _Результат = <...> | Проверить

Условие: Сегодня понедельник | 94

_Результат = ДеньНедели(ТекущаяДата()) = 1

Имя для вызова: Имя алгоритма для вызова в программном коде... ?

Комментарий:

Алгоритм Telegram

Обратите внимание

В задаче не заполнено:
- Условие завершения не выполнено: Сегодня понедельник

Задача не может быть переведена в состояние "Выполнена".
Исправьте замечания и повторите попытку.

Спасибо, понятно

☆ Сегодня понедельник (Алгоритмы)

Записать и закрыть | _Результат = <...> | Проверить

Условие: Сегодня понедельник | 94

_Результат = ДеньНедели(ТекущаяДата()) = 1

Имя для вызова: Имя алгоритма для вызова в программном коде... ?

Комментарий: Можно завершить только в понедельник

Алгоритм Telegram

Обратите внимание

В задаче не заполнено:
- Условие завершения не выполнено: Можно завершить только в понедельник

Задача не может быть переведена в состояние "Выполнена".
Исправьте замечания и повторите попытку.

Спасибо, понятно

☆ Сегодня понедельник (Алгоритмы)

Записать и закрыть | _Результат = <...> | Проверить

Условие: Сегодня понедельник | 94

_Результат = ДеньНедели(ТекущаяДата()) = 1;
_РасшифровкаРезультата = "Завершите задачу в следующий понедельник"

Имя для вызова: Имя алгоритма для вызова в программном коде... ?

Комментарий:

Алгоритм Telegram

Обратите внимание

В задаче не заполнено:
- Условие завершения не выполнено: Завершите задачу в следующий понедельник

Задача не может быть переведена в состояние "Выполнена".
Исправьте замечания и повторите попытку.

Спасибо, понятно

Циклические условия (01 ноября 2022)

Если для реализации сценария требуется **циклический повтор** определенных шагов (например, пока не выполнено определенное условие), то следует учитывать следующее:


1. Для линий связи, приводящих к **повтору ранее пройденных шагов**, следует устанавливать **признак повтора**. См. пример выше.
2. Если циклическое повторение подразумевает автоматическую программную проверку данных, то в таких случаях следует устанавливать для шагов цикла **отложенный запуск**. Отложенный запуск даст определенный запас времени, чтобы изменения в системе успели произойти перед очередным циклом.
3. Если циклическое повторение **не** включает автоматическую программную проверку данных, то отложенный запуск шагов следует устанавливать только в соответствии с требуемой логикой.


ПРИМЕР:

В данном примере происходит автоматическая проверка условия **Условие1**. Если условие выполнено, сценарий переходит на шаг **Обработка1**, после чего процесс завершается.

Если условие **не** выполнено, процесс переходит на шаг **Обработка2**, после чего процесс возвращается на проверку условия.

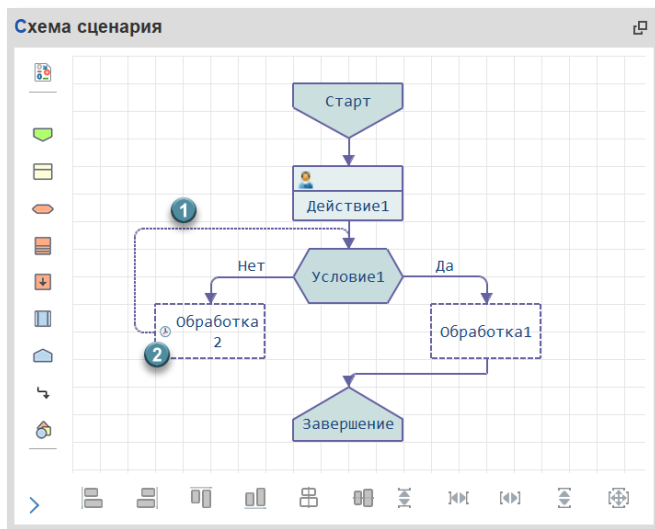
Это будет происходить до тех пор, пока условие не будет выполнено.

Для корректной работы такого сценария для линии связи **1**, установлен **признак повтора**. Линия, при этом, изменяется на пунктирную .

Для шага **Обработка2** установлен отложенный запуск **2**. На элементе схеме отображается пиктограмма отложенного запуска .

Если, проверка условия подразумевает проверку, например, *остатков товаров* или *оплаты от клиента*, то нет смысла делать отложенный запуск **через 1 минуту**, т.к. в этом случае условие будет проверяться **каждую минуту** (вряд ли, после очередной проверки остатков, они появятся через 1 минуту). В подобных случаях (в том числе, и принимая во внимание минимизацию малополезной нагрузки на сервер) оправдано ставить отложенный запуск, измеряемый **в часах**.

Дополнительно, в системе предусмотрена настройка **Отложенный запуск только при повторах**. При включенной настройке, если шаг сценария повторяется, то первый запуск этого шага будет выполняться без отсрочки.



Настройки работы с задачами в Telegram (01 ноября 2022)

Самая простая настройка оповещения о новой задаче выглядит, как показано справа.

| Настройки оповещений | | |
|-------------------------|------------------|--------------------|
| Этап сценария | Вид события | Тип оповещения |
| Подзадача/Условие/Выбор | Получатель | Сообщение |
| Выполнить действие | При запуске шага | Telegram |
| | Исполнитель | У вас новая задача |

Оповещение в Telegram будет выглядеть следующим образом:

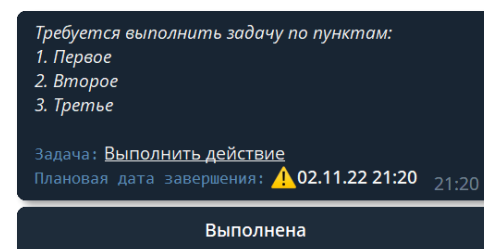
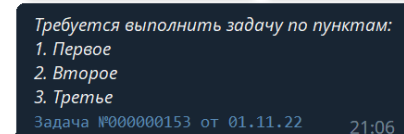
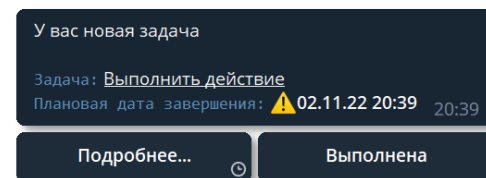
1. Текст сообщения,
2. Название задачи,
3. Плановая дата завершения (если указана),
4. Кнопки «Подробнее...» и «Выполнена».

При нажатии кнопки «Подробнее...» будет показано **описание задачи**, как оно задано в настройках сценария.

НАСТРОЙКА: Если при настройке оповещения не указывать текст сообщения, а оставить поле пустым (в поле **Сообщение** будет отображаться строка «[По умолчанию]»), то в оповещении сразу будет отображаться **подробное описание задачи**.

В этом случае, в дополнительных настройках основного чат-бота можно установить настройку **Без кнопки «Подробнее...»**.

Оповещение будет выглядеть нагляднее и проще.



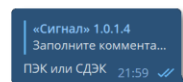
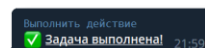
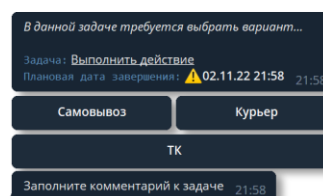
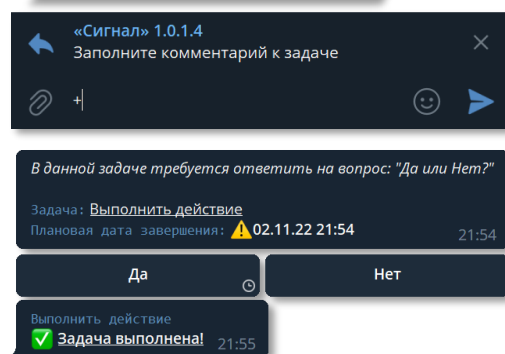
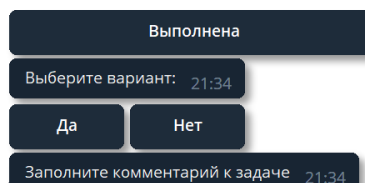
Система позволяет использовать Telegram для выполнения следующих задач:

1. не требующих заполнения обязательных реквизитов,
2. только с обязательным заполнением **комментария**,
3. только с обязательным **выбором варианта из списка**,
4. только с обязательной **проверкой условия** для последующего шага вида **Условие**,
5. только с обязательным **выбором варианта** для последующего шага вида **ВыборВарианта**,
6. с обязательным заполнением **комментария** и одним из вышеперечисленных обязательных реквизитов.

Для выполнения задачи требуется сначала нажать кнопку «**Выполнена**», а затем следовать инструкциям чат-бота.

Для обязательного ввода **Комментария** чат-бот предложит указать его в поле ввода сообщения и затем отправить его.

В случае **Проверки условия** будет предложено 2 кнопки «**Да**» и «**Нет**». После нажатия одной из кнопок потребуется ввести комментарий.



НАСТРОЙКА: В случае с проверкой **Условия**, **Выбором варианта** или **Выбором варианта из списка** система позволяет упростить процесс выполнения задач с помощью следующих настроек.

В дополнительных настройках сценария можно включить настройку **Автозаполнение комментариев задач**. При включенной настройке, в задачах, где проверяется условие и выбрано «**Да**», ручной ввод комментария не требуется.

В дополнительных настройках основного чат-бота можно установить настройку **Без кнопки «Выполнена»**. Если настройка включена, то для задач с проверкой условия или выбором вариантов, кнопка «Выполнена» в Telegram отображаться не будет. Вместо неё, будут показываться кнопки выбора вариантов.

Таким образом, процесс фиксации выполнения задач будет упрощен и будет минимизировано количество информации в окне сообщений.

Выполнение программного кода в чат-боте (01 ноября 2022)

Небольшая вспомогательная функция, которая позволяет отправлять чат-боту **программный код на встроенной языке 1С**. Чат-бот в ответ возвращает результат выполнения.

Для активации такой возможности в дополнительных настройках чат-бота следует включить настройку **Разрешить выполнять программный код**.


Но, в общем случае, рекомендуем использовать **встроенный в расширение «Сигнал» Конструктор чат-ботов** для возможности запуска **алгоритмов** по ключевым словам.

The screenshot shows a chat interface with the following messages:

- User: 1+1 22:08 ✓
- User: ТекущаяДата() 22:08 ✓
- User: 2 22:08
- User: 01.11.2022 22:08:40 22:08
- Bot: К сожалению, не могу понять, что вы имеете ввиду: 1/0 22:09
- User: 1/0 22:09 ✓
- User: 1/0 22:15 ✓
- Bot: Извините, это не в моей компетенции. Попробуйте поискать в Яндексе: 1/0 22:15
- Search Result: Яндекс 1/0 Найдётся всё 22:15
- User: Формат(ТекущаяДата(), "ДФ=DD") 22:15 ✓
- User: 1 ноября 2022 г. 22:15

Автоматическая остановка и автовозобновление задач (02 ноября 2022)

Иногда требуется остановить выполнение задачи по процессу.

Для этого в программе предусмотрен ручной режим остановки задачи с помощью кнопки .

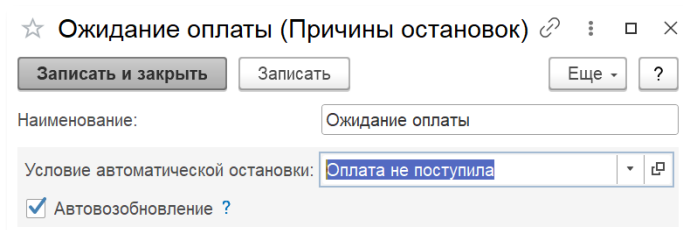
Но более интересным является вариант **автоматической** остановки задачи по условию.

Для этого в настройках справочника Причины остановок предусмотрен поле **Условие автоматической остановки**. В этом поле можно выбрать алгоритм, который программно определит требуется ли остановить задачу или нет. В данном случае алгоритм используется как условие.

В том случае, если **алгоритм** вернёт значение **_Результат = Истина**, система **автоматически остановит задачу**, в которой данная причина остановки указана как допустимая.

Более того, для этой причины можно установить признак **Автовозобновление**. Задача автоматически вернётся в работу, если предыдущее условие перестанет выполняться, т.е. вернёт значение **_Результат = Ложь**.

ПРИМЕР: В качестве примера можно привести ситуацию, когда система останавливает задачу пользователя, если клиент не произвёл оплату заказа. В этом случае система проверяет остатки взаиморасчётом с клиентом. Если оплата не поступала, то задача автоматически останавливается. Как только в системе появляется оплата и в причине остановки установлен признак **Автовозобновление**, система вернёт задачу пользователя в работу.



Групповые задачи и условия (03 ноября 2022)

Рассмотрим простой пример с «ручной» проверкой условия.

В зависимости от ответа пользователя в задаче – «Да» или «Нет», процесс пойдет по разным ветвям сценария.

Здесь разных трактовок поведения системы возникнуть не может.

Но, если задачу **Проверить условие вручную** сделать **групповой** и назначить её выполнение **группе пользователей**, то возникает вопрос о том, как система должна обрабатывать ответы всех пользователей.

Ведь в системе при запуске сценария появится не одна задача, а столько задач, сколько пользователей наделены указанной в настройках сценария ролью. В данном примере, это роль **Менеджеры**.

И только после того, как **все** пользователи выполняют свои задачи и выполняют ручную проверку условия, сценарий перейдет на следующий шаг.

И вот здесь возникают вопросы:


1. Какой шаг сценария будет следующим?
2. Выполнено ли условие, если хотя бы 1 пользователь ответил «Нет»?
3. Может быть, условие выполнено, если хоть один из пользователей ответил «Да»?
4. А что если после хотя бы одного ответа «Нет» дальнейшие проверки условий не имеют смысла, т.к. условие **уже** не выполнено?

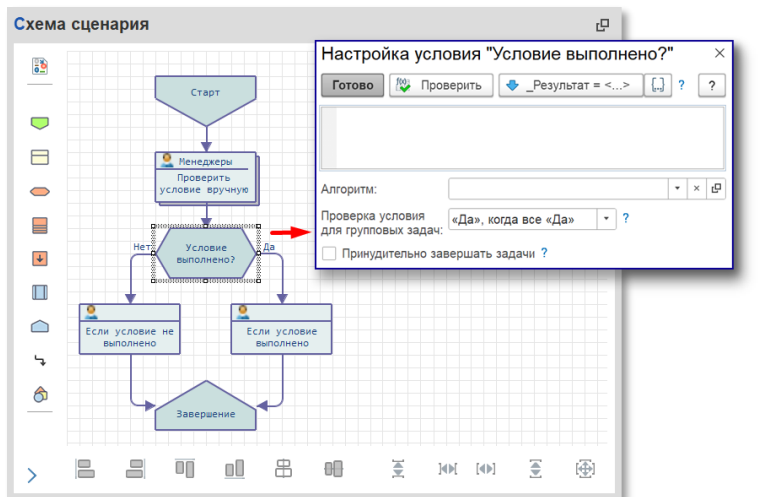
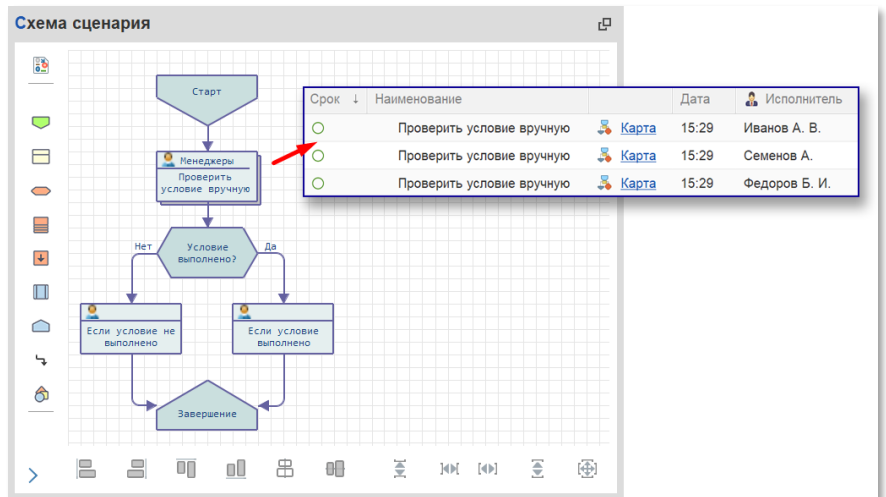
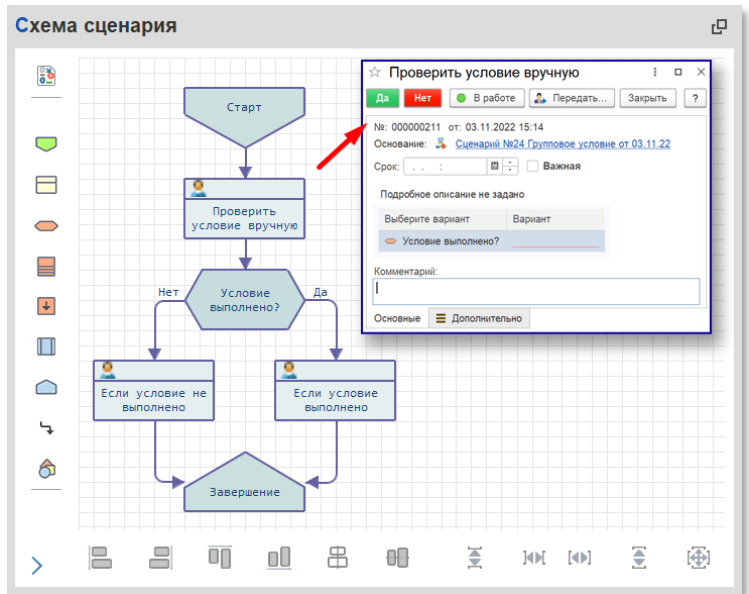
Для того, чтобы система могла разрешать подобные «коллизии», в настройках условий предусмотрены следующие настройки:

1. Проверка групповых условий:
 - «Да», когда все «Да»,
 - «Да», когда любое «Да».

Логика обработки **всех** условий зависит от данной настройки.

2. Принудительно завершать задачи
 - Если хотя бы в одной (уже выполненной) задаче выбрано «Нет», остальные, не взятые в работу задачи в группе, будут завершены принудительно.

Если выбран вариант «Да», когда все «Да» и включена настройка **Принудительно завершать задачи**, то при любой отрицательной проверке, все остальные задачи пользователей будут завершены автоматически и процесс по сценарию перейдет на следующий шаг. Автоматически завершённые задачи отмечаются в списке пиктограммой .



| Срок | Наименование | Дата | Исполнитель |
|------|-----------------------------|-------------|-------------|
| | ✓ Проверить условие вручную | Карта 15:47 | ПолныеПрава |
| | ✓ Проверить условие вручную | Карта 15:47 | |
| | ✓ Проверить условие вручную | Карта 15:47 | |
| | ○ Если условие не выполнено | Карта 15:47 | ПолныеПрава |

Таблицы произвольных данных (03 ноября 2022)

В предыдущем примере мы требовали от пользователя заполнения **условно обязательного реквизита**, необходимого в процессе выполнения сценария. Но, в ряде случаев, может потребоваться заполнение более «обширных» данных. Данных, которые можно представить в виде таблицы.

С одной стороны, заполнение большого количества данных при выполнении задач, логичнее производить в первичных объектах системы, с которыми связан процесс.

ПРИМЕР: Если процесс запущен, например, по заявке на возврат товаров от покупателя, от исполнителя задачи может потребоваться указать товары для возврата, их количество и причины возврата. Логично, что эти данные можно вносить непосредственно в первичном документе – источнике события возникновения процесса. А в задачах по процессу исполнитель будет просто отмечать факт выполнения необходимых действий.

Но не всегда есть возможность заполнять табличные данные в первичных документах. Например, при отсутствии подходящего первичного документа в рамках протекающего процесса. Или при отсутствии в первичном документе необходимых разрезов данных, которые требуется обрабатывать по сценарию.

Для подобных случаев в сценариях предусмотрен механизм **Таблиц данных**.

Для каждого сценария можно настроить **Таблицу данных** с произвольными колонками, а также настроить правила заполнения этих данных в задачах, которые будут ставиться исполнителям в процессе выполнения сценария.

При настройке заполняется состав колонок таблицы данных.

Здесь же настраивается возможность изменения состава строк таблицы и изменение их порядка. Для этого предназначены флаги **Изменять состав строк** и **Изменять порядок строк**.

После основной настройки таблицы данных, для каждого шага сценария вида **Действие** настраиваются доступы исполнителей к таблице в целом и к отдельным колонкам.

По умолчанию, все доступы к таблице данных в задачах отключены.

В дальнейшем, при работе с задачей, её выполнение возможно **только** если в **Таблице** будут заполнены все данные, в соответствии с доступами исполнителей к **Таблице данных**.

Таблица данных отображается на формах процессов и

задач на отдельной закладке Основные Таблица данных Дополнительно.

ПРОГРАММИРОВАНИЕ: Система позволяет заполнять **Таблицу данных** программно. Возможные варианты программного заполнения:

1. Автозаполнение при ручном создании нового процесса. Для этого используется **Алгоритм заполнения таблицы данных**, которые указывается на форме настройки таблицы данных.

Доступ к таблице данных: `_ПроцессОбъект.ТаблицаДанных` (тип `ТаблицаЗначений`).
Имена колонок указаны в таблице настройки в соответствующей колонке.

2. Автозаполнение при автоматическом создании нового процесса. Для этого используется **Алгоритм, выполняемый при запуске нового процесса**.

Доступ к таблице данных: `_ПроцессОбъект.ТаблицаДанных` (тип `ХранилищеЗначений`).
Имена колонок указаны в таблице настройки в соответствующей колонке.

3. В любых алгоритмах, в которых есть доступ к переменной `_Процесс`.

Настройка таблицы данных

| Заголовок | Ширина | Тип объекта | Видимость | Обязательный |
|-----------|--------|---|-------------------------------------|--------------------------------------|
| Подсказка | | Отбор | Только просмотр | <input type="checkbox"/> Имя колонки |
| Клиент | 20 | Справочник.Партнеры | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | | Клиент Равно "Да" | <input type="checkbox"/> | Колонка1 |
| Товар | | Справочник.Номенклатура | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | | ПометкаУдаления Равно... | <input type="checkbox"/> | Колонка2 |
| Цена | | Число | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> | Колонка3 |
| Аналог | | Булево | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> | Колонка4 |

Изменять состав строк Контроль заполнения таблицы данных в процессе
 Изменять порядок строк Алгоритм заполнения таблицы данных:


Для выполнения задачи требуется заполнить:

Изменять состав строк Изменять порядок строк


| Колонка | Видимость | Только просмотр | <input checked="" type="checkbox"/> |
|---------|-------------------------------------|-------------------------------------|-------------------------------------|
| Клиент | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Товар | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Цена | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Аналог | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |

Настройки Таблица данных

Выполнение задач по электронной почте (04 ноября 2022)

Расширения для автоматических оповещений  «Сигнал» позволяет **выполнять задачи по электронной почте**.

Возможно выполнение следующих видов задач:

1. Простые задачи, в которых достаточно просто нажать кнопку  **Выполнена**,
2. Задачи, в которых требуется дополнительно заполнить **Комментарий**,
3. Задачи **с выбором**. В этом случае при выполнении требуется выбрать результат выполнения из списка.

ПРИМЕР: Возьмем задачу, в которой требуется выбрать результат согласования.

Для того, чтобы пользователь мог её выполнить, настроим оповещения по сценарию.

При создании задачи, **исполнителю** будет отправлено письмо с темой «Согласование» и текстом письма «Согласовать проект договора».

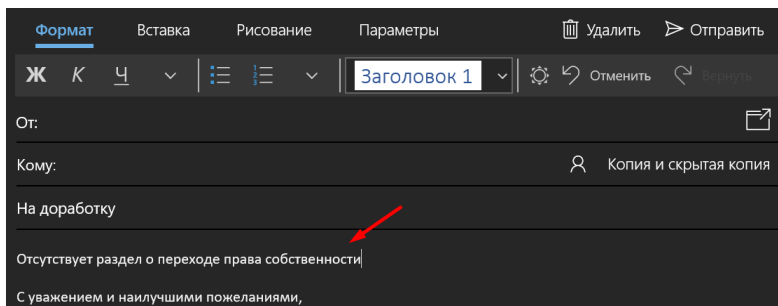
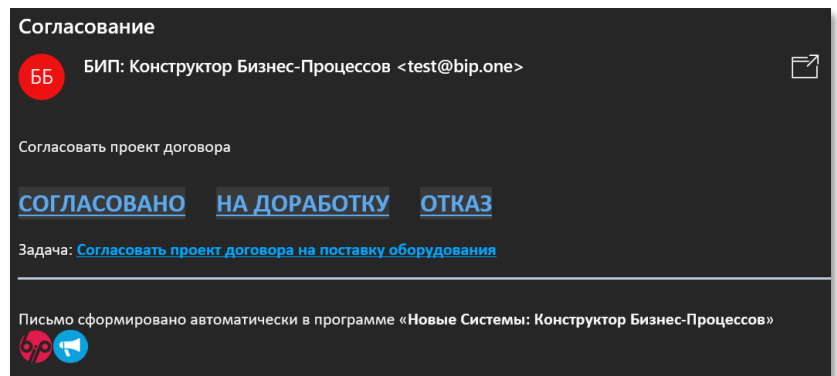
| Этап сценария | Вид события | Тип оповещения | Контекст/ Тема |
|-------------------------|------------------|-----------------------------|----------------------|
| Подзадача/Условие/Выбор | Получатель | Сообщение | |
| Согласовать проект ... | При запуске шага | Электронная почта | Тема: «Согласование» |
| | Исполнитель | Согласовать проект договора | |

В задаче настроен выбор результата из списка (в нашем примере это список результатов согласования), поэтому в письме, также, будут **кнопки для выполнения задачи**.

Отметим здесь, что **подвал письма** можно изменить в настройках программы.

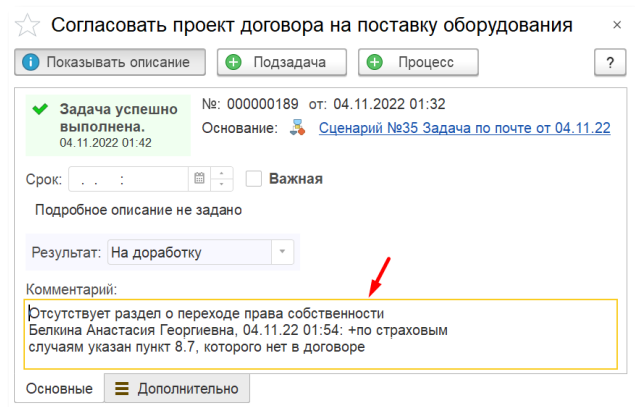
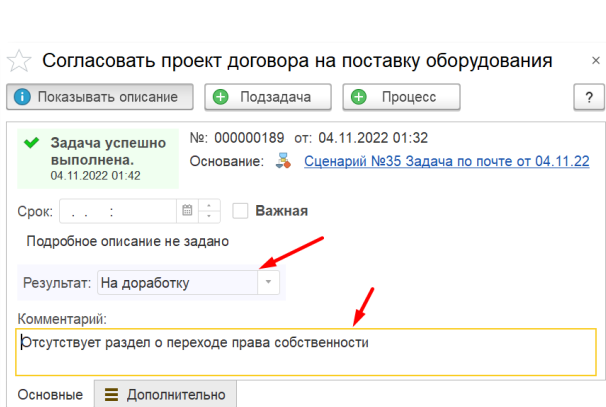
Для выполнения задачи исполнитель нажимает одну из кнопок.

Затем, в ответном письме оставляет комментарий и отправляет письмо.



Система обработает входящее письмо и отметит выполнение задачи.

Если после выполнения, исполнитель захочет дополнить свой комментарий, он может повторно ответить на письмо. В этом случае, его комментарий будет **добавлен** к основному комментарию задачи.

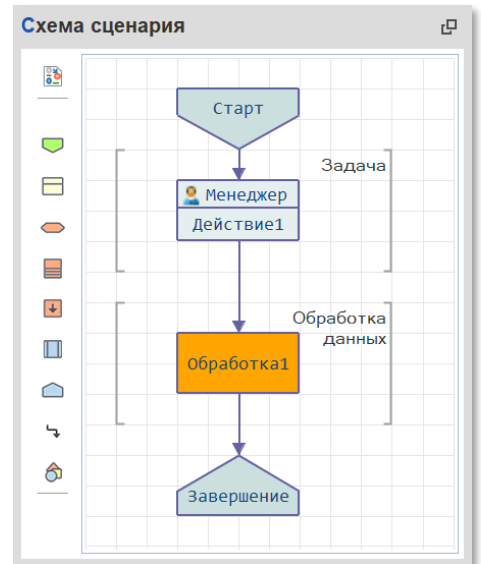


ПРИМЕР

Рассмотрим пример.

По сценарию, после выполнения **Задачи**, система должна выполнить шаг **Обработка**, в которой указан **программный код**, который вносит изменения в базу данных:

- перепроводит документы,
- выполняет обмен данными с другими информационными ресурсами,
- пакетно формирует новые документы
- и т.п.



В подобных случаях, когда может подразумеваться длительное выполнение процедуры обработки данных, для шагов вида **Обработка** рекомендуется устанавливать **Отложенный запуск** с минимальным задержкой **1 минута**.

Дело в том, что при завершении задачи, система выполняет проверку выполненной задачи (возможность её выполнения), а затем, **в той же самой транзакции** на сервере обрабатывает очередной шаг. Если очередной шаг – это проверка условия, или запуск следующей задачи, или запуск вложенного процесса, то такая логика работы системы не вносит каких-либо заметных «эффектов» при интерактивной работе с текущим шагом. Но, если после задачи, следует шаг, в котором будет происходить длительная процедура обработки данных информационной базы (множество циклов чтения/обработки/записи в базу данных), то на стороне клиента это может привести к «подвисанию» клиентской формы текущего шага.

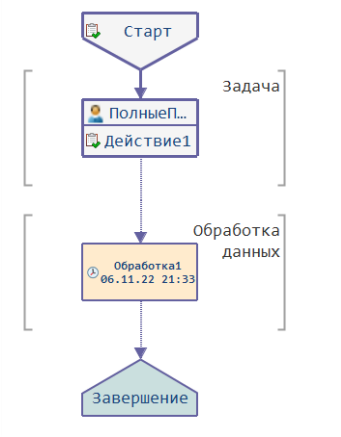
Настройка элементов схемы (Diagram Element Settings) показывает таблицу с параметрами запуска для каждого этапа:

| N | Этап | Описание | Запуск |
|----|------------|-------------|--------------|
| 01 | Старт | | Запуск... |
| 02 | Действие1 | Описание... | Запуск... |
| 03 | Обработка1 | | через 1 мин. |
| 04 | | | |

Открытое окно 'Настройка запуска' (Launch Settings) для элемента 'Обработка1':

- Отложенный запуск
- Запустить сразу
- Запуск по условию
- Запустить через: 1 Минута
- Запустить на дату:
- Дата запуска указана в предыдущей задаче
- Только при повторах ?

Текущая карта процесса (Current Process Map)



При установке минимального отложенного запуска, шаг

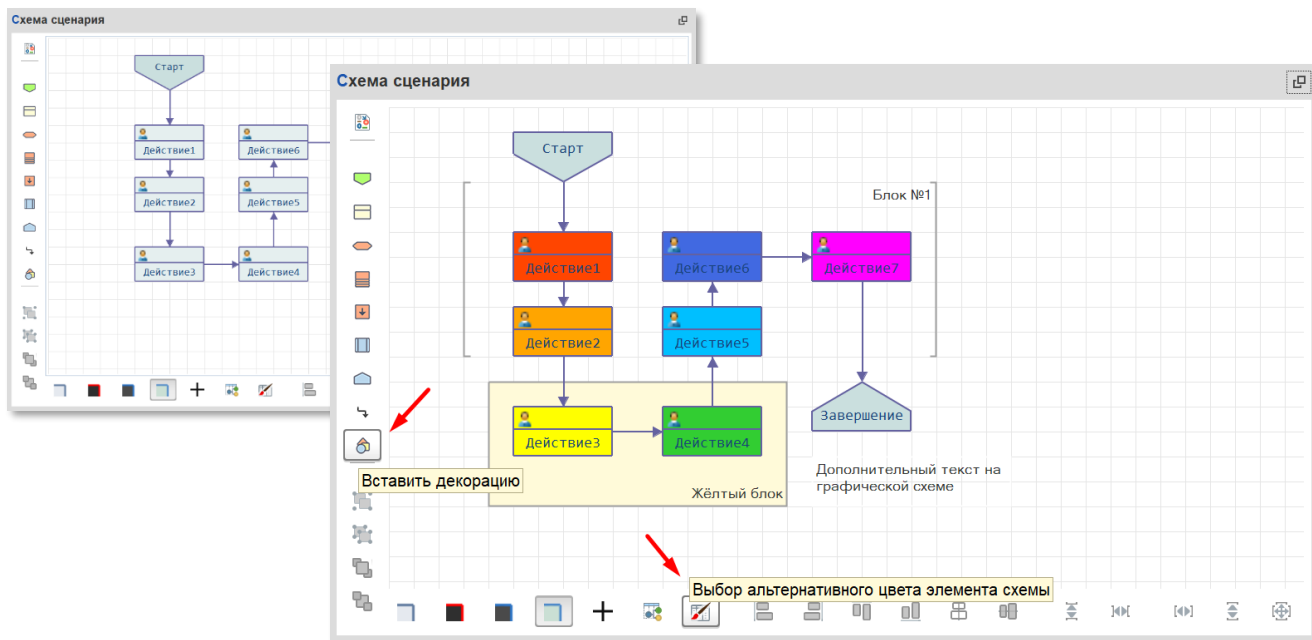
Обработка будет выполняться на сервере в **фоновом режиме**, а транзакция выполнения текущей задачи завершится **сразу** после проверки и записи задачи.

На карте процесса **отложенный шаг**, до того, как будет запущен, отображается с пиктограммой и **временем планируемого запуска**.

Оформление графической схемы (06 ноября 2022)

В ряде случаев, для улучшения восприятия графической схемы сценария, может оказаться полезным использование возможностей **оформления** схемы.

Для этого в системе предусмотрены **декорации** и **настройка цветов отдельных элементов схемы**.

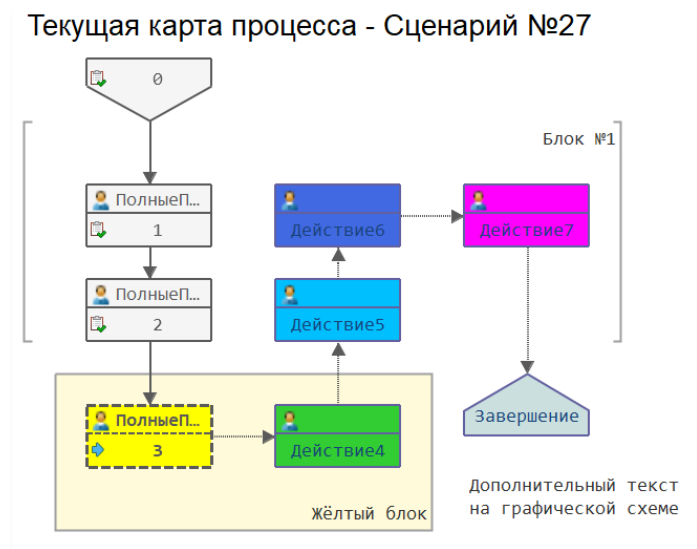


Для настройки добавленной на схему декорации следует в контекстном меню графической схемы выбрать пункт **Свойства**. В открывшемся окне можно задать различные настройки декорации.

Доступные виды декораций:

- блок,
- папка,
- файл,
- документ,
- декоративная стрелка (вверх, вниз, влево, вправо, вверх-вниз, влево-вправо),
- скобки вертикальные,
- скобки горизонтальные,
- эллипс.

Графическое оформление схемы будет отображаться не только при настройке схемы, но и на **динамической карте процесса**.



Свойства: Декорация

• Основные:

Имя: Декорация2
Наименование: Обработка данных

Гиперссылка:

• Характеристики:

Горизонтальное г: Прижать вправо
Вертикальное по: Прижать вверх
Подсказка:

• Оформление:

Цвет фона: Авто
Цвет рамки: стиль: Рамка
Цвет текста: стиль: Текст формы
Шрифт: Авто
Положение карти: Лево
Прозрачный фон:

Картинка:

Размер картинки: Автоматический размер
Фигура: Скобки вертикальные

• Расположение:

| | | | |
|--------|-----|--------|-----|
| Лево | 20 | Верх | 160 |
| Ширина | 180 | Высота | 80 |

Также, на **динамической карте процесса**, в ряде случаев, может оказаться полезным **отключение наименования выполненных шагов** и замена их на **порядковые номера** в последовательности шагов сценария.

Для этого следует нажать кнопку **Трекер**.