Инструкция по созданию процесса BPMN

СОДЕРЖАНИЕ

1.Базовое взаимодействие 3
1.1 Создание тестового пользователя 3
1.2 Создание БП 3
1.3 Форма задачи 9
1.3.1 Создание формы 9
1.3.2 Сохранение комментария к задаче в историю 12
1.4 Добавление условий в БП 14
1.5 Публикация БП 15
1.6 Запуск БП 17
2. Взаимодействие с документом 20
2.1 Прикрепление документа к процессу 20
2.2 Взаимодействие с документом в процессе 22
2.2.1 Пример с условиями (flow condition) 23
2.2.2 Пример изменения метаданных документа 25
2.3 Отображение полей документа на форме задачи 26
3. Взаимодействие с кейсом 29
3.1 Предопределенные переменные 29
3.2 Сервисы 29
3.2.1 Case Completeness Service 29
3.2.2 Case Status Service 29
3.2.3 Flowable Recipients Service 29
3.3 Listeners 31
3.3.1 Execution listeners 31
3.3.1.1 Flowable document set listener 31
3.3.1.2 Flowable Check Completeness Levels ExecutionListener 31
3.3.2 Tasks listeners 32
3.3.2.1 Flowable Check Mandatory Variable On Outcome Task Listener 32
3.3.2.2. Check Lists Task Listener 34
4. Согласование задач через email 35
5. Локализация 38
5.1 Outcomes 38
5.2 Поля 40
5.3 Названия задач 41

1.Базовое взаимодействие

1.1 Создание тестового пользователя

1. Открыть сервер http://bpm.citeck.ru/share. Логин/пароль: admin/ tiny room drive.
2. Нажать в главном меню «Пользователи».
3. Нажать «Новый пользователь» - открыта страница создания пользователя.

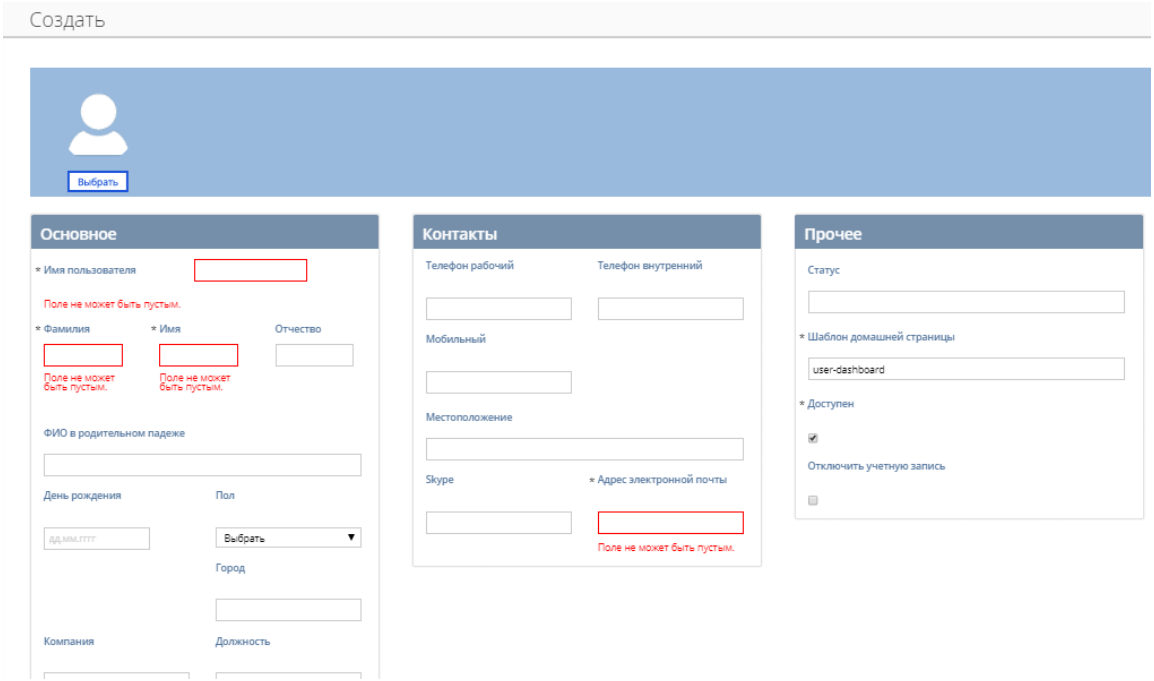


Рисунок 1 - Страница создания пользователя.

4. Заполнить обязательные атрибуты (запомнить логин и пароль).
5. Нажать «Создать» - пользователь с указанными данными создан на сервере.

1.2 Создание БП

1. Нажать в главном меню «Редактор бизнес-процессов» – открыта страница логина в моделер на рисунке 2.

Рисунок 2 - Открыта страница логина в моделер

2. Ввести логин/пароль: admin/ tiny room drive – открыт моделер на рисунке 3.

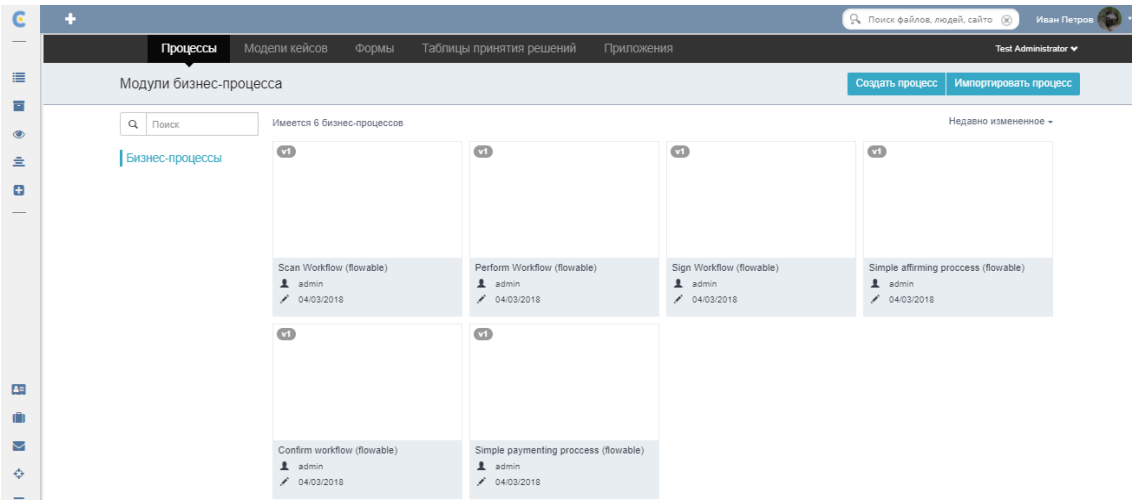


Рисунок 3 - Открыт моделер.

3. Открыть вкладку Процессы.
4. Нажать «Создать процесс».
5. Заполнить поля "Наименование модели" и "Ключ бизнес-процесса" (ключ должен быть написал латинскими буквами и не должен содержать пробелов).
6. Нажать "Создать новую модель" - откроется страница моделирования процесса на рисунке 4. В левой панели отображаются активности BPMN 2.0, которые можно использовать при моделировании, основная канва - для создания модели, снизу есть панель с параметрами объекта модели (например, исполнитель задачи, её наименование, события и так далее).

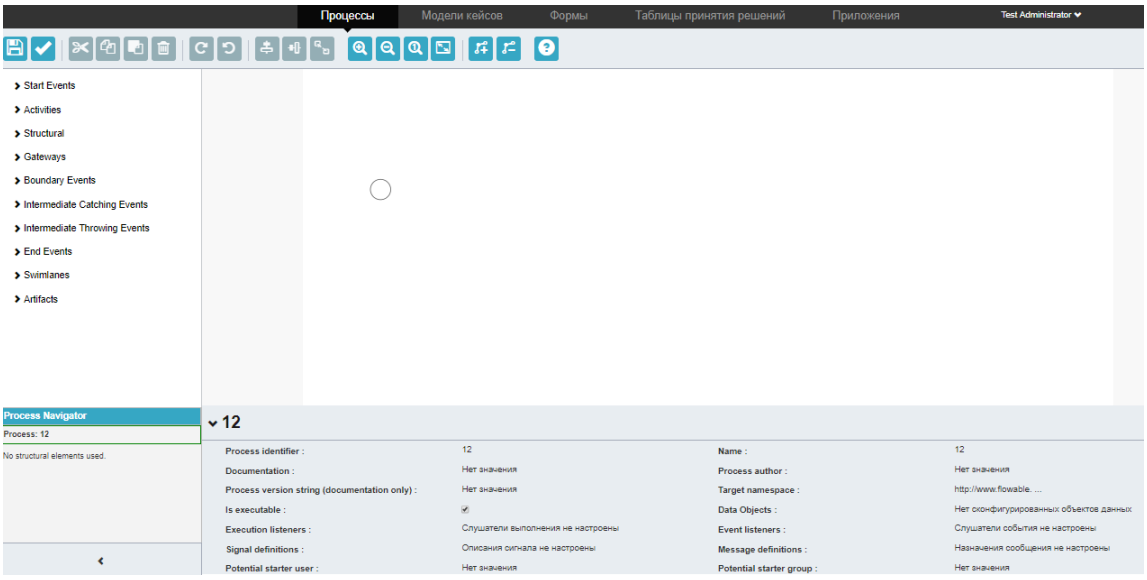
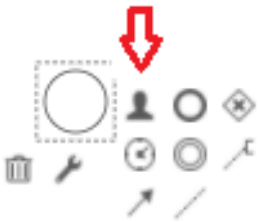


Рисунок 4 - Страница моделирования процесса.

7. Кликнуть на событие старта на экране - откроется всплывающее меню.

8. Выбрать пользовательскую задачу – появится пользовательская задача, связанная с событием старта.



9. Нажать на пользовательскую задачу.

10. В кардлете параметров заполнить Name = Согласование (рисунок 5).

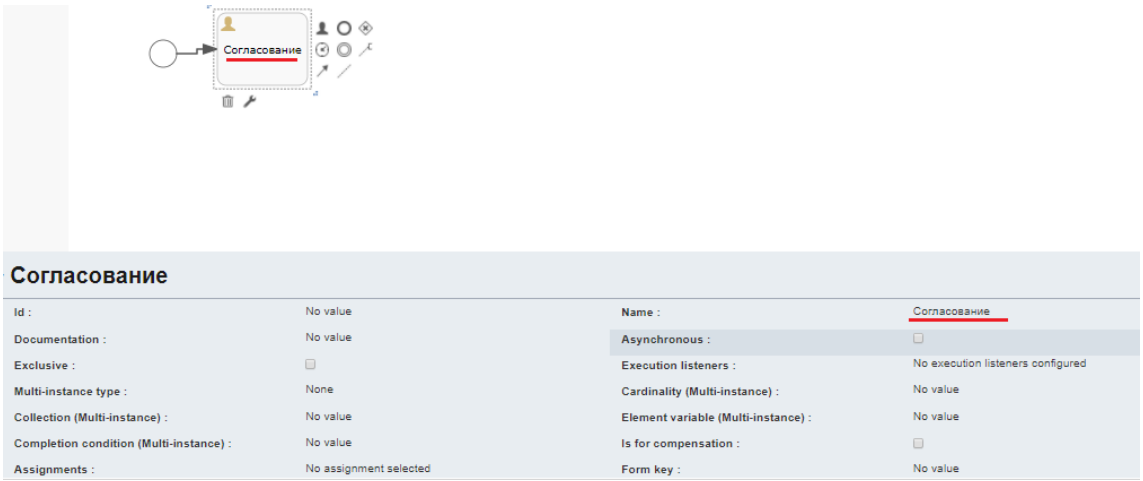


Рисунок 5 – Заполнение параметра Name

11. Выбрать во всплывающем меню Exclusive gateway – будет создано условие, связанное с пользовательской задачей на рисунке 6.

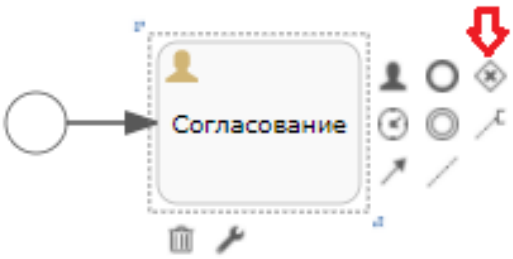


Рисунок 6 – Выбор условия.

12. Выбрать во всплывающем меню событие завершения - будет создано событие завершения(рисунок 7), связанное с разветвлением. Дать название связи – Согласовано.

Рисунок 7 – Событие завершения.

13. Выбрать во всплывающем меню пользовательскую задачу - будет создана новая пользовательская задача. Дать название задаче – Доработка, дать название связи – На доработку (при необходимости объекты можно перемещать по канве). Рисунок 8.

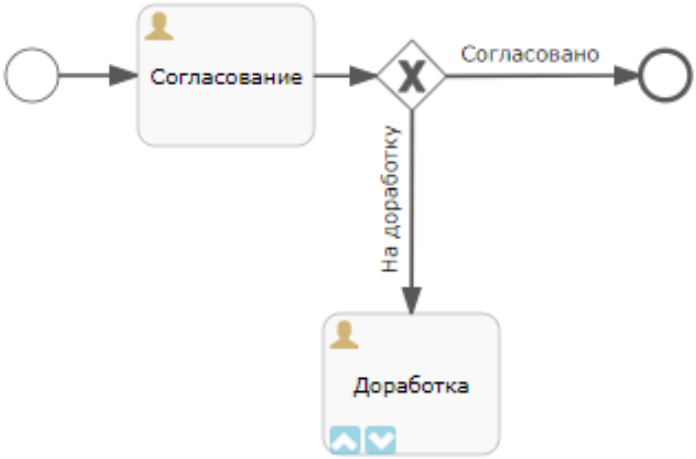


Рисунок 8 – Схема пользовательской задачи и созданные связи.

14. Нажать на задачу Доработка.
15. Зажать стрелочку и протянуть её к задаче Согласование – будет создана связь между двумя задачами(рисунок 9). Дать название связи – Выполнено.

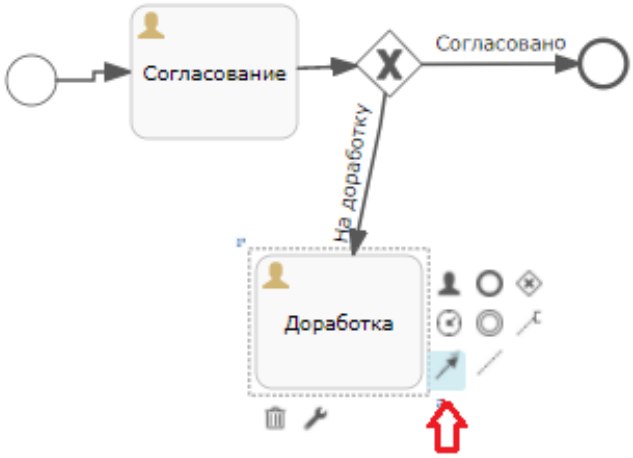


Рисунок 9 – Создание связи между двумя задачами.

16. Нажать на задачу Согласование.
17. Заполнить значение Assignments = test user (для выбора тестового пользователя необходимо нажать на атрибут Assignments, в открывшемся окне выбрать вкладку Установленные значения, в атрибуты «Назначенное лицо» и «Кандидаты» ввести логин тестового пользователя(рисунок 10), созданного ранее).

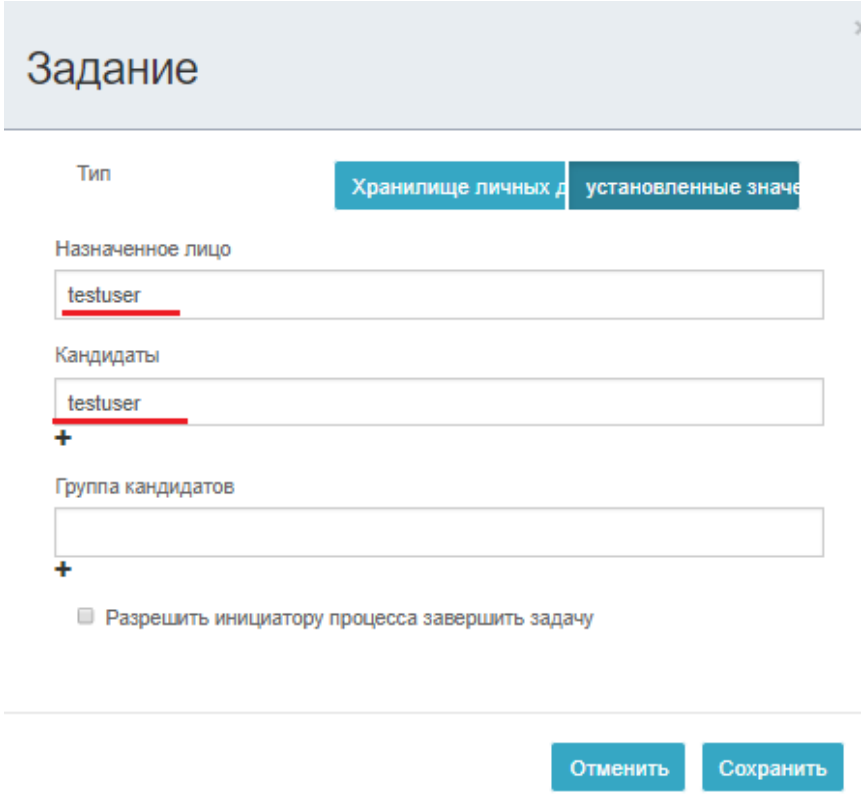


Рисунок 10 – Заполнение атрибутов «Назначенное лицо» и «Кандидаты».

18. Нажать на задачу Доработка.
19. Заполнить значение Assignments = Инициатор процесса.

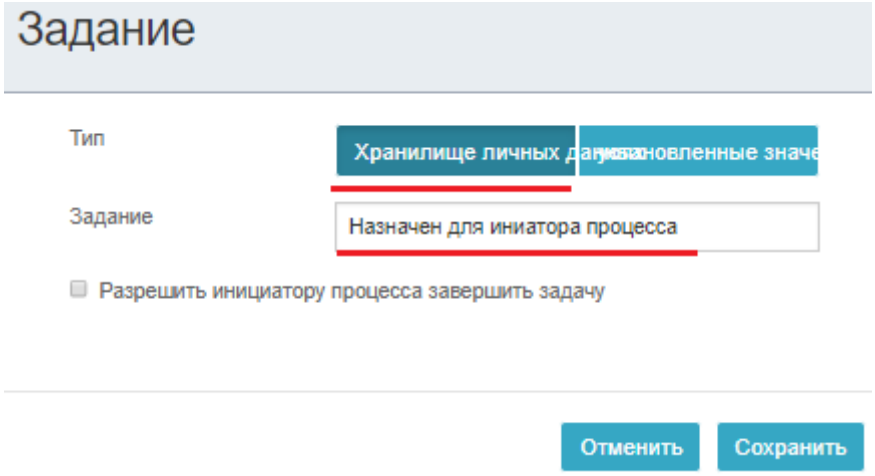


Рисунок 11 – Заполнение значения Assignments = Инициатор процесса.

20. Нажать «Сохранить» в левом верхнем углу.

21. Выбрать «Сохранить».

1.3 Форма задачи

1.3.1 Создание формы

1. Нажать на задачу Согласование.
2. Нажать на атрибут «Form reference» - откроется окно выбора форм, можно выбрать готовую или создать новую.
3. Нажать «Новая форма» - откроется форма создания новой формы.
4. Заполнить поля "Наименование формы" = «Форма согласования» и "Ключ формы" = approveform (значение ключа не должно содержать пробелов и написано латинскими буквами).
5. Нажать «Создать новую форму» - откроется страница создания формы (рисунок 12). В левой панели расположены атрибуты, которые можно добавить на форму, основное поле содержит две вкладки Дизайн и Результаты, на первой вкладке можно размещать атрибуты формы, на второй - возможные вердикты.

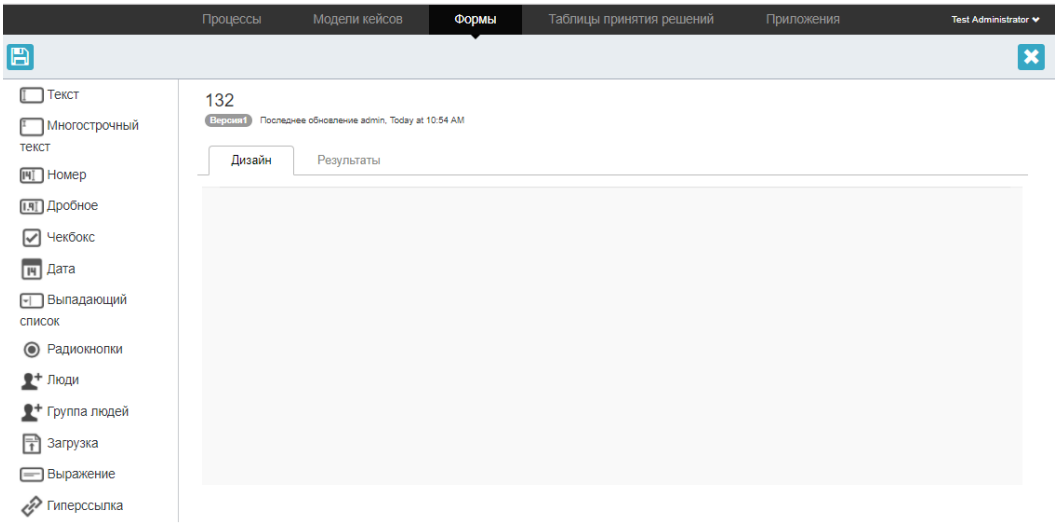
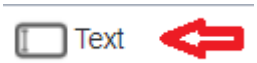


Рисунок 12 – Страница создания формы.

6. На вкладку Дизайн добавить перетаскиванием из левой части несколько атрибутов, дать атрибутам названия и указать другие характеристики. Например, текстовое поле с названием «Введите комментарий» (рисунок 13 и рисунок 14).



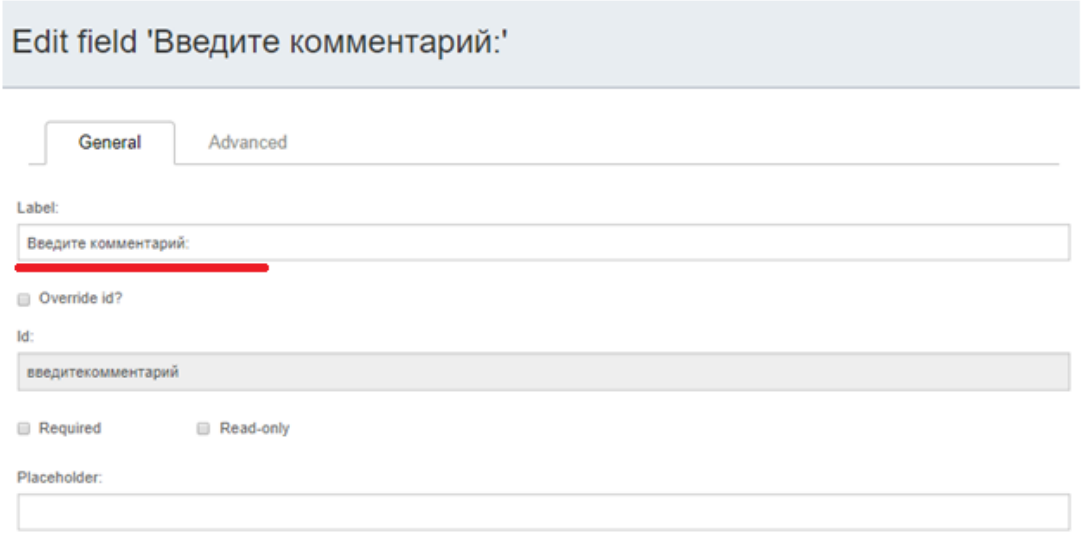


Рисунок 13 – Текстовое поле «Введите комментарий»

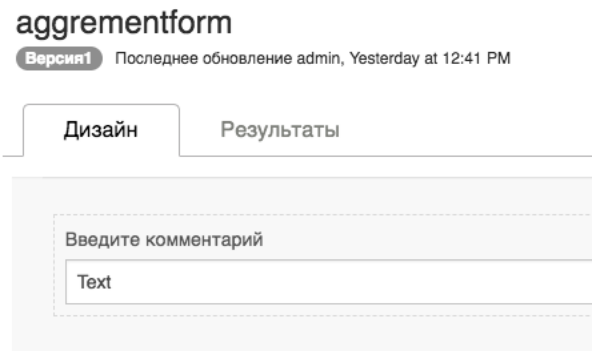


Рисунок 14 – Вкладка «Дизайн».

7. Открыть вкладку Результаты.
8. Выбрать чекбокс «Использовать результаты пользователей для этой формы» - откроется список Возможные результаты с полем для ввода значения первого результата (рисунок 15).
9. Ввести в поле результата – Approve.
10. Нажать «Добавить результат» - добавлено новое поле для значения еще одного результата.

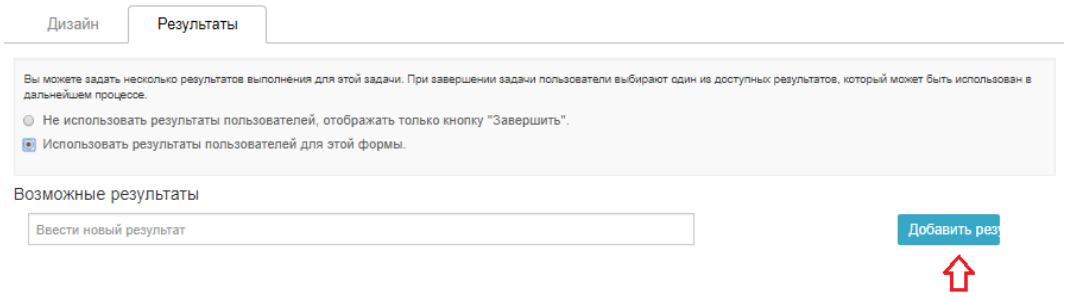


Рисунок 15 – Возможные результаты с полем для ввода значения первого результата.

11. Ввести во второе поле – Reject (рисунок 16).

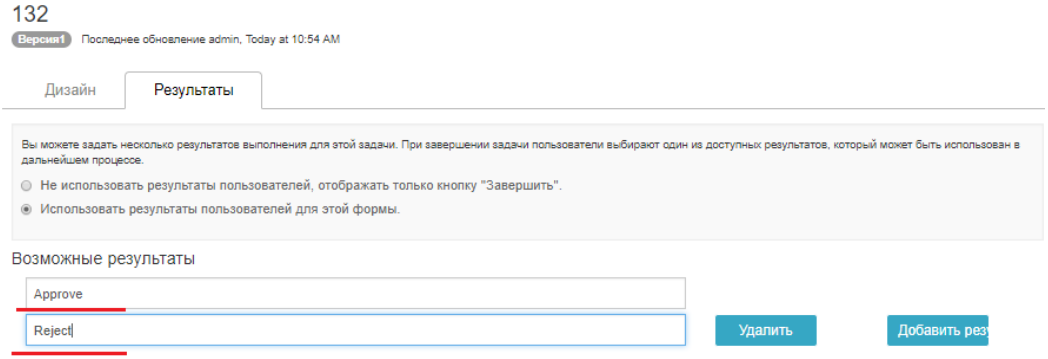


Рисунок 16 – Поле Reject.

12. Нажать «Сохранить» в левом верхнем углу.

13. Выбрать «Сохранить и закрыть редактор» - открыта страница процесса, в задаче Согласование заполнено значение «Form reference» = approveform
14. Нажать на задачу Доработка.
15. Нажать на атрибут «Form reference» - откроется окно выбора форм.
16. Нажать «Новая форма» - откроется форма создания новой формы.
17. Заполнить поля "Наименование формы" = «Форма доработки» и "Ключ формы" = reworkform (значение ключа не должно содержать пробелов)
18. Нажать «Создать» - откроется страница создания формы.
19. Добавить на вкладку Дизайн несколько атрибутов (например, текстовое – Изменения, и дата – Дата доработки). Рисунок 16.

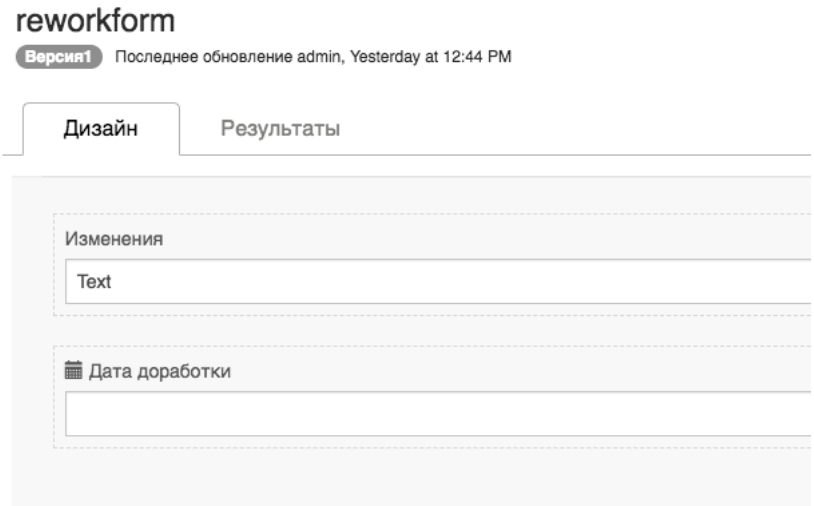


Рисунок 16 – Вкладка Дизайн с добавленными атрибутами.

20. Открыть вкладку Результаты.
21. Выбрать чекбокс «Использовать результаты пользователей для этой формы» - откроется список Возможные результаты с полем для ввода значения первого результата.
22. Ввести в поле результата – Done.
23. Нажать «Сохранить» в левом верхнем углу.

24. Выбрать «Сохранить и закрыть редактор» - открыта страница процесса, в задаче Доработка заполнено значение «Form reference» = reworkform.

1.3.2 Сохранение комментария к задаче в историю

В пункте 1.3.1 в форме задачи Согласования было добавлено поле комментария, результат комментария можно сохранить в историю событий. Для этого необходимо:

1. Перезаписать id у поля комментарий. Формат id – id формы_comment. Таким образом, id поля комментарий для формы *Согласования* будет равен approveform_comment (рисунок 17).

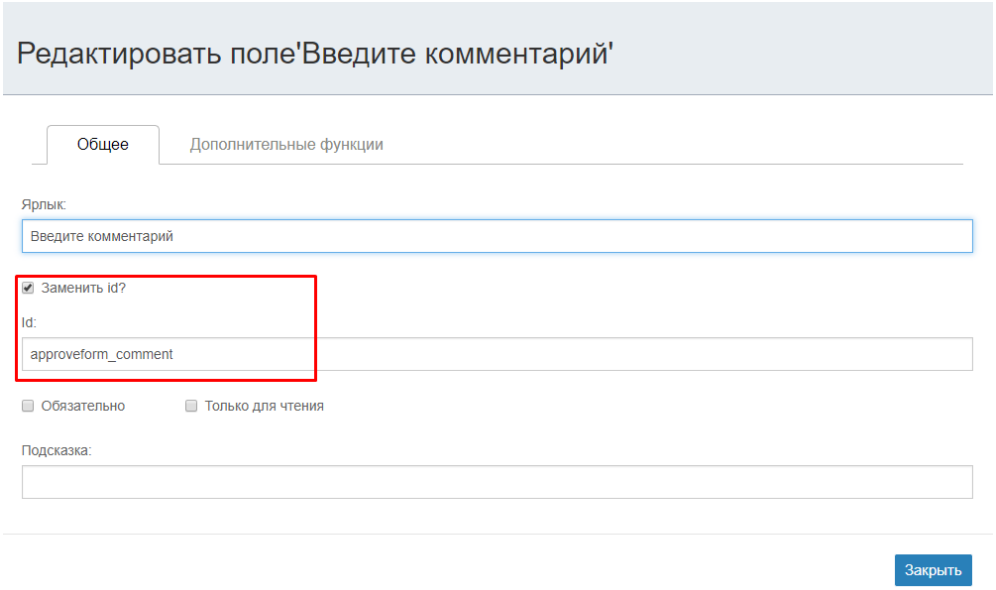


Рисунок 17 – Комментарий approveform_comment для формы «Согласования».

2. В системном журнале добавить объект соотношения процесса и поля комментария.
   a. Открыть меню, выбрать *системные журналы* в разделе *Инструменты администратора.*
   b. Выбрать системный журнал *Процессы flowable и кастомные поля комментариев.*

c. Нажать создать *Процессы flowable и кастомные поля* (рисунок 18).

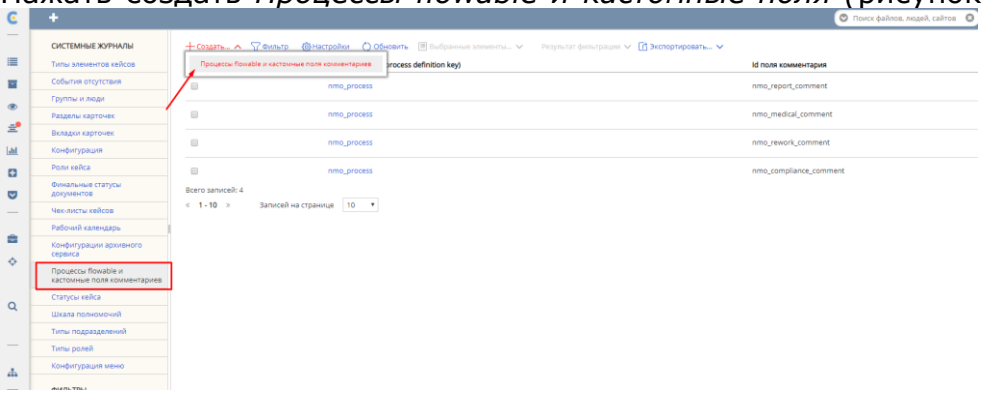


Рисунок 18 – Процессы flowable и кастомные поля.

d. На форме создания ввести id процесса (Ключ бизнес-процесса, указанный вами ранее) и id поля комментария (рисунок 19).

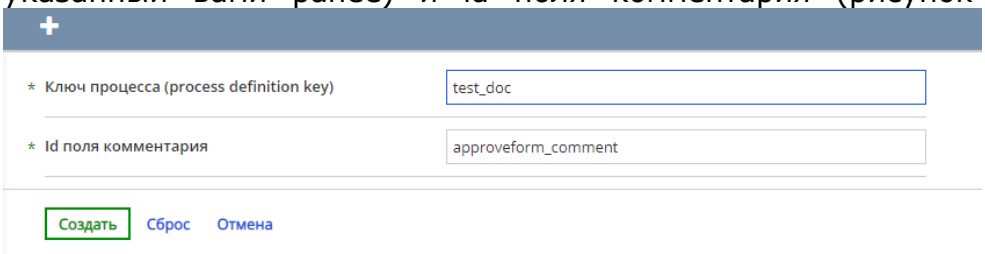


Рисунок 19 – Ввод id процесса и id поля комментария.

e. Нажать создать.

Как результат, при выполнении задачи согласования, заполненный комментарий будет сохраняться в истории событий.

Заполняем комментарий (рисунок 20):

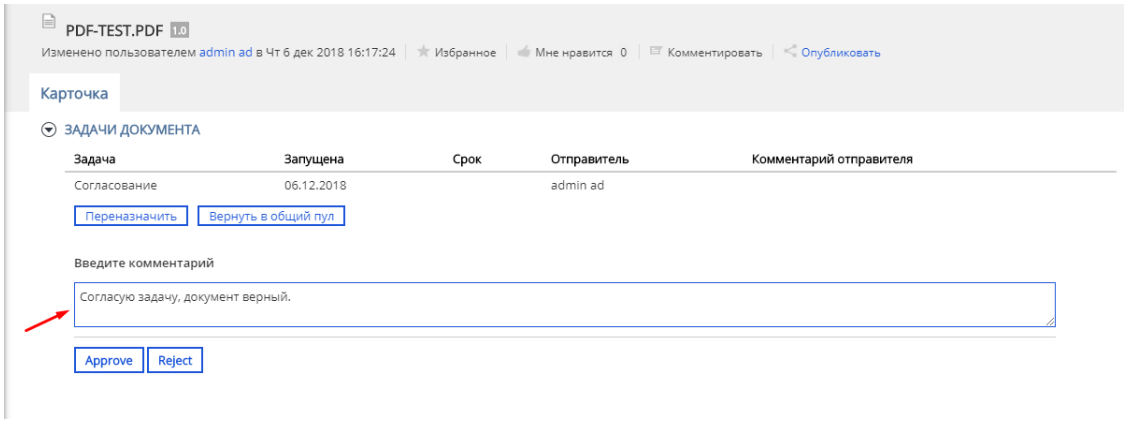


Рисунок 20 – Заполнение комментария.

Результат сохранился в историю событий (рисунок 21):

06.12.2018 16:36:39	Задача завершена	1.0	admin ad	admin	Задача бизнес-процесса	Согласую задачу, документ верный.

Рисунок 21 – История событий

1.4 Добавление условий в БП

1. Нажать на связь Согласовано (рисунок 22).

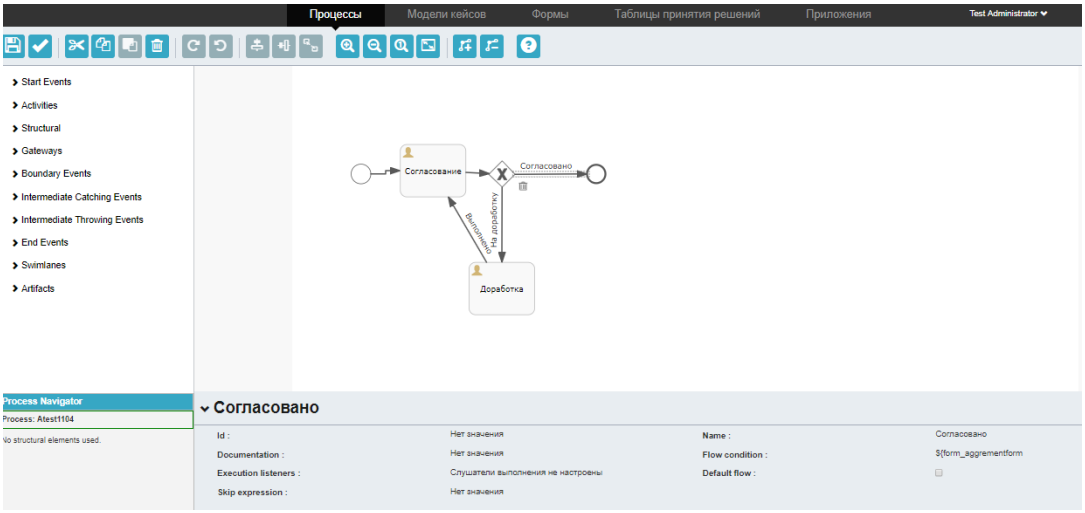


Рисунок 22 – Выделена связь «Согласовано».

2. Нажать на атрибут «Flow condition» - откроется окно создания условия
3. Ввести в поле «Выражение условий выполнения» значение ${form_approveform_outcome == "Approve"}(рисунок 23) (где approveform – ключ формы, "Approve" – один из вердиктов формы, который мы задали ранее).

Рисунок 23 – Выражение условий выполнения.

4. Нажать «Сохранить» - форма создания условия закрыта, атрибут «Flow condition» содержит значение из предыдущего пункта.
5. Нажать на связь «На доработку".
6. Нажать на атрибут «Flow condition» - откроется окно создания условия
7. Ввести в поле «Выражение условий выполнения» значение ${form_approveform_outcome == "Reject"}.
8. Нажать «Сохранить» - форма создания условия закрыта, атрибут «Flow condition» содержит значение из предыдущего пункта.
9. Нажать на связь Выполнено.
10. Нажать на атрибут «Flow condition» - откроется окно создания условия
11. Ввести в поле «Выражение условий выполнения» значение ${form_reworkform_outcome == "Done"}.
12. Нажать «Сохранить» - форма создания условия закрыта, атрибут «Flow condition» содержит значение из предыдущего пункта.

13. Нажать «Сохранить» в левом верхнем углу.

14. Выбрать «Сохранить и закрыть редактор» - форма создания процесса закрыта.

1.5 Публикация БП

1. Открыть вкладку Приложения в моделере (рисунок 24).

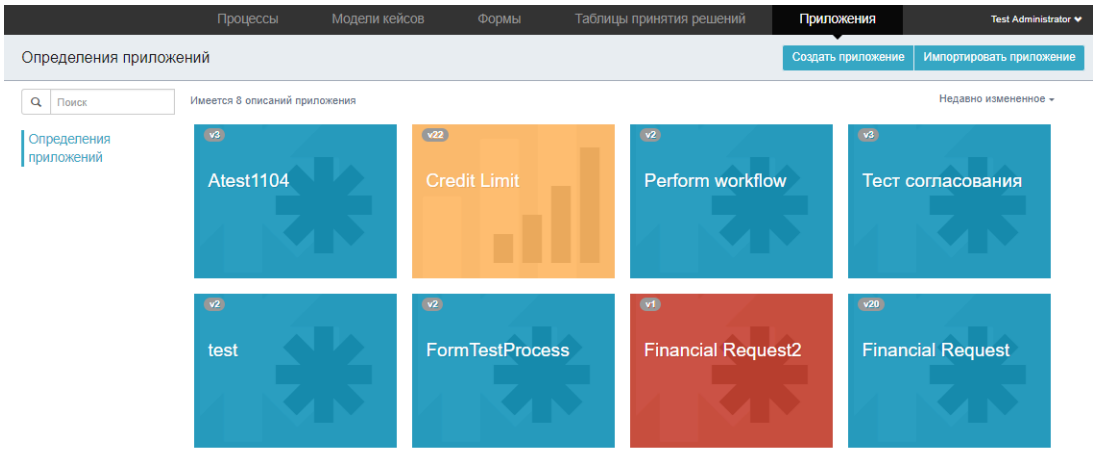


Рисунок 24 – Открытая вкладка «Приложения».

2. Нажать «Создать приложение».
3. Заполнить поля "Наименование приложения" и "Ключ приложения" (значение ключа не должно содержать пробел).
4. Нажать «Создать новое определение приложения» – форма создания приложения закрыта, открыта страница созданного приложения на рисунке 25.

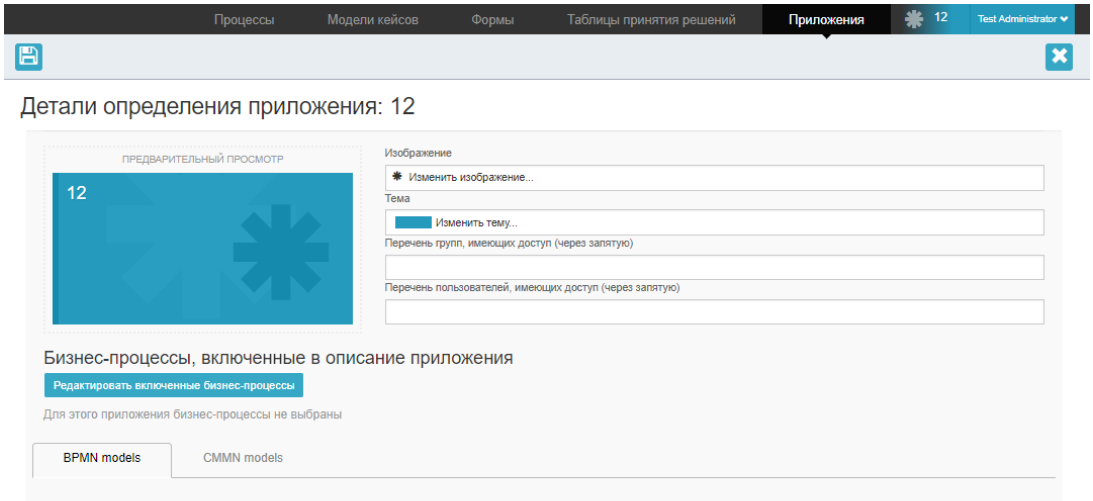


Рисунок 25 – Страница созданного приложения

5. Нажать «Редактировать включенные бизнес-процессы»(рисунок 26).

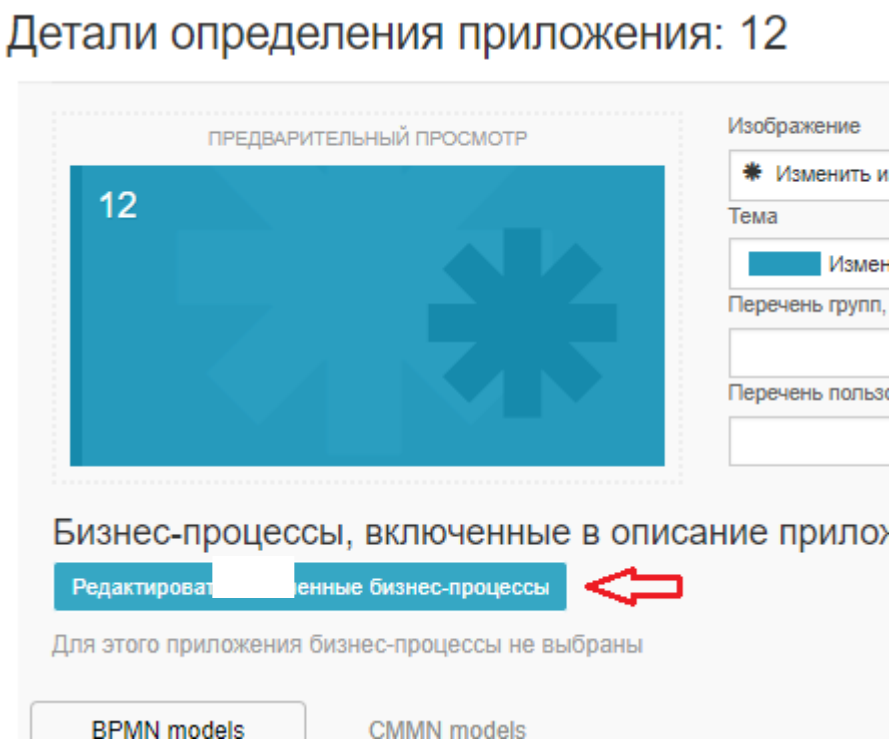


Рисунок 26 – Расположение кнопки редактировать включенные бизнес-процессы.

6. Нажать на процесс, созданный на предыдущих шагах – процесс будет помечен в правом верхнем углу как выбранный.
7. Нажать «Закрыть» - схема выбранного процесса отображается внизу страницы на вкладке BPMN models (рисунок 27).

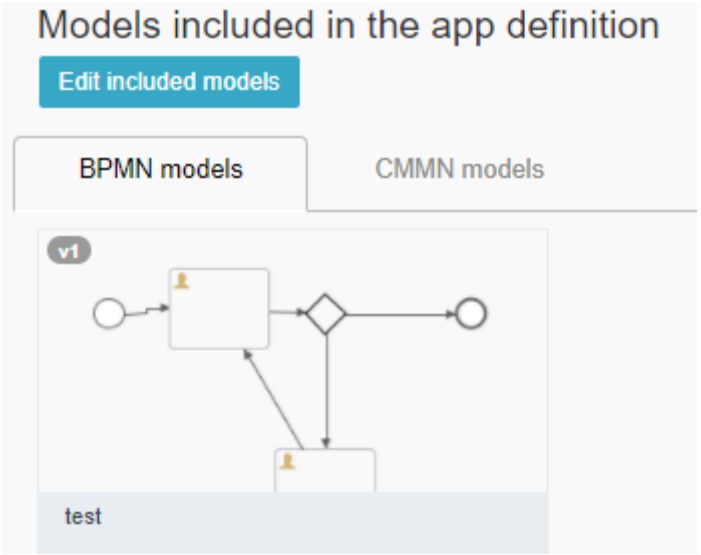


Рисунок 27 – Вкладка BPMN models

8. Нажать «Сохранить» в левом верхнем углу.

9. Выбрать «Сохранить и закрыть редактор» - открыта вкладка Приложение, созданное приложение присутствует на вкладке.
10. Нажать на созданное приложение – открыта страница приложения.
11. Нажать «Опубликовать» в правом верхнем углу на рисунке 28.

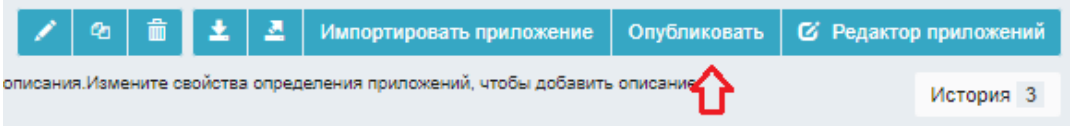


Рисунок 28 – Расположение кнопки «Опубликовать»

12. Подтвердить публикацию процесса – созданный процесс будет добавлен на сервер bpm.citeck.ru.

1.6 Запуск БП

1. Нажать в главном меню «Текущие задачи».
2. Нажать «Запустить бизнес-процесс» - откроется форма запуска бизнес-процесса с выпадающим списком бизнес-процессов (рисунок 29).

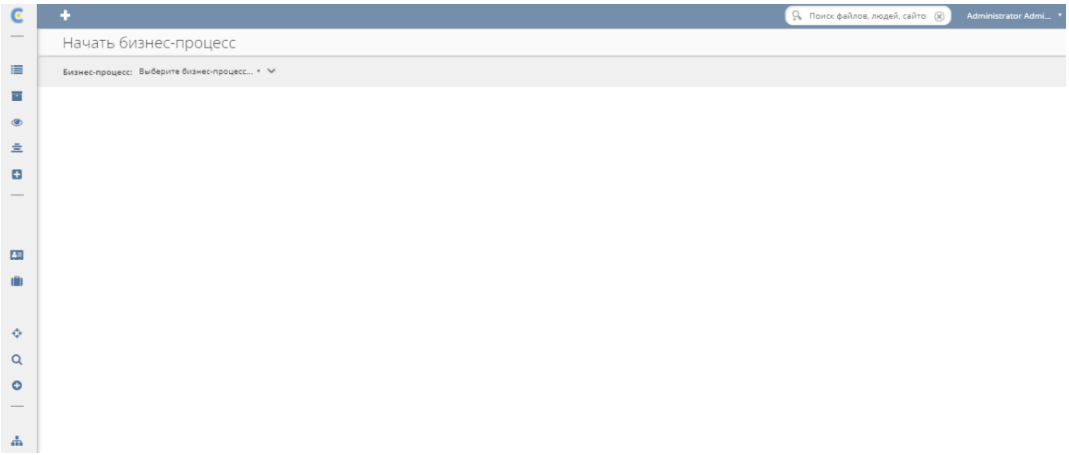


Рисунок 29 – Форма запуска бизнес-процесса с выпадающим списком.

3. Выбрать в выпадающем списке созданный в моделере процесс – открыта форма запуска БП на рисунке 30.

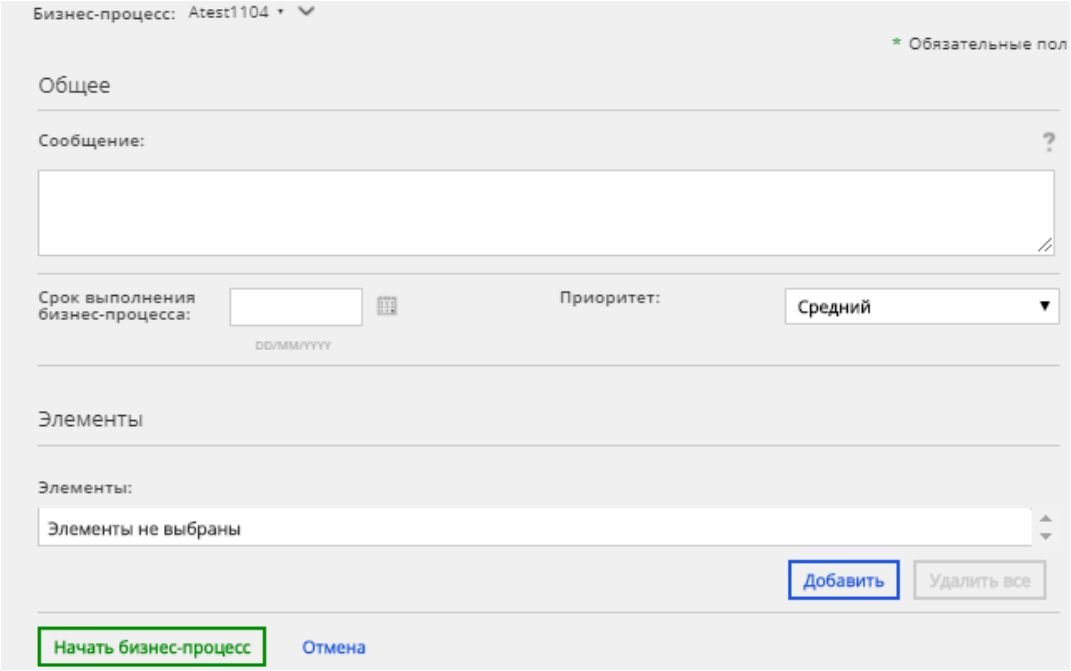


Рисунок 30 – Форма запуска бизнес-процесса.

4. Нажать «Начать бизнес процесс» - на тестового пользователя назначена задача Согласования.
5. Зайти на сервер http://bpm.citeck.ru/share под тестовым пользователем.
6. Открыть Текущие задачи – в списке представлена задача Согласования на рисунке 31.

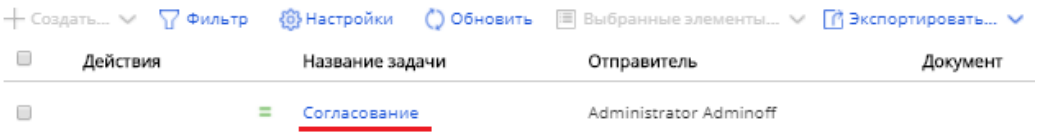


Рисунок 31 – Задача «Согласования».

7. Открыть назначенную задачу Согласования – представлена форма с текстовым атрибутом «Введите комментарий» и вердиктами «Approve» и «Reject» на рисунке 32.

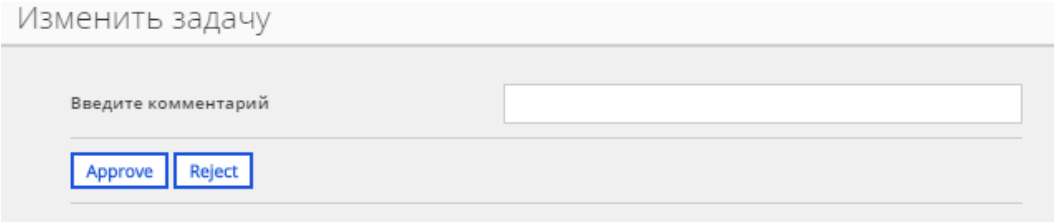


Рисунок 32 – Форма с текстовым атрибутом «Введите комментарий».

8. Нажать «Reject» - на Инициатора назначена задача Доработки.
9. Зайти на сервер http://bpm.citeck.ru/share под admin.

10. Открыть Текущие задачи – в списке представлена задача Доработки (рисунок 33).

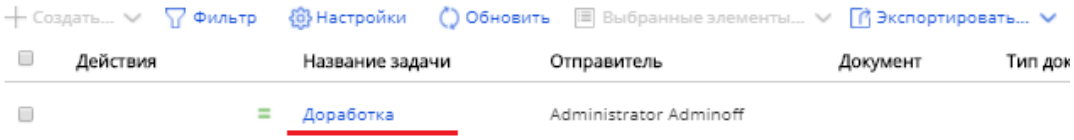


Рисунок 33 – Задача «Доработка».

11. Открыть назначенную задачу Доработки – представлена форма с вердиктом «Done» (рисунок 34).

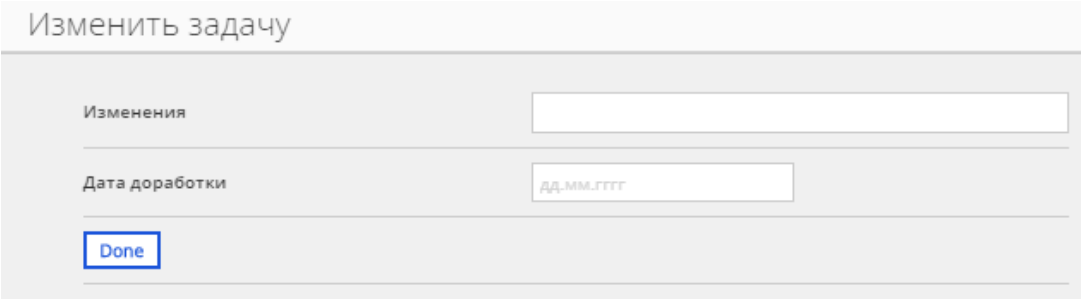


Рисунок 34 – Изменение задачи «Доработка».

12. Нажать «Done» - на тестового пользователя назначена задача согласования.
13. Зайти на сервер под тестовым пользователем.
14. Открыть Текущие задачи.
15. Открыть назначенную задачу согласования и завершить её с вердиктом «Approve» - процесс будет завершен, новых задач нет.

2. Взаимодействие с документом

Из страницы запуска бизнес процесса можно указать элемент, по которому запускается процесс. Таким элементом может выступать любой документ, например, как определенная сущность в системе – договор или счет, так и любой загруженный пользователем документ. В рамках процесса мы можем взаимодействовать с документом, изменять его метаданные, настраивать условие переходов в зависимости от метаданных, отсылать уведомления по документу и т.д.

2.1 Прикрепление документа к процессу

На форме запуска БП, в разделе Элементы можно выбрать документ, по которому запускается БП. Для этого:

1. Кликнуть на кнопку добавить (рисунок 35).

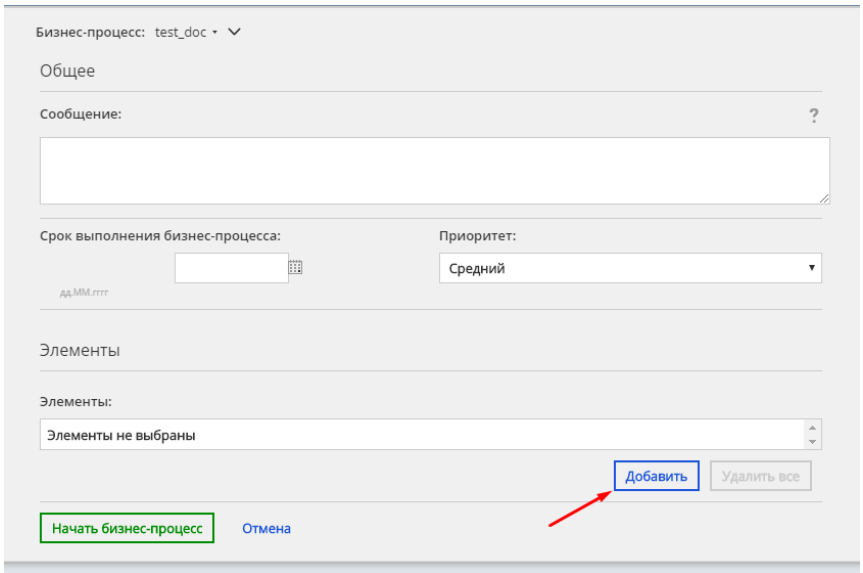


Рисунок 35 – Расположение кнопки «Добавить».

2. Выбрать необходимый объект и нажать ОК (рисунок 36).

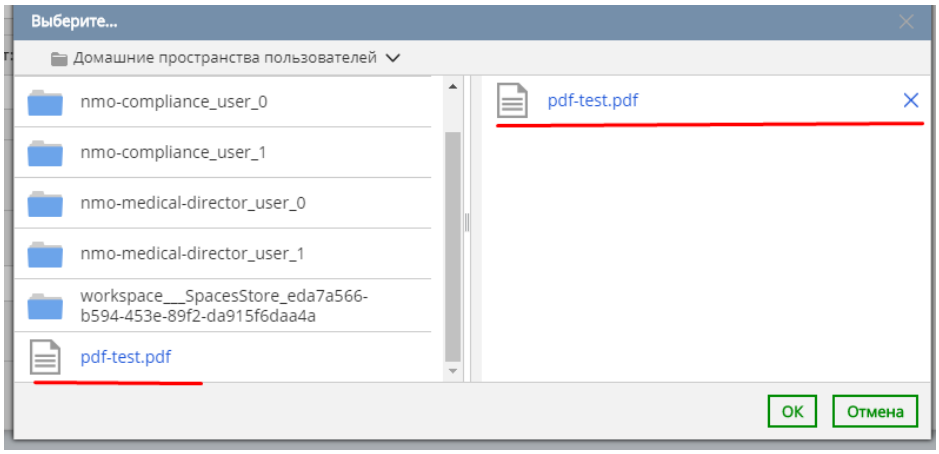


Рисунок 36 – Результат после нажатия кнопки ОК.

3. В элементах появится выбранный объект, по которому запустится БП (рисунок 37).

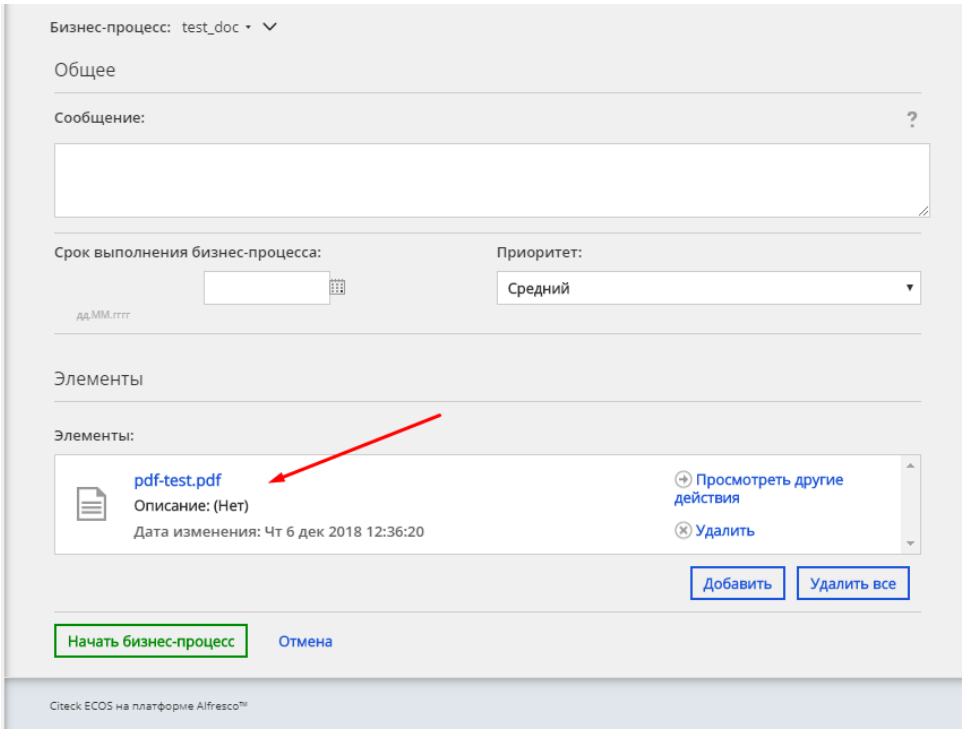


Рисунок 37 – Выбор элемента.

Теперь если перейти на карточки документа *pdf-test.pdf*, то на ней можно отслеживать текущие задачи, процессы и метаданные (рисунок 38).

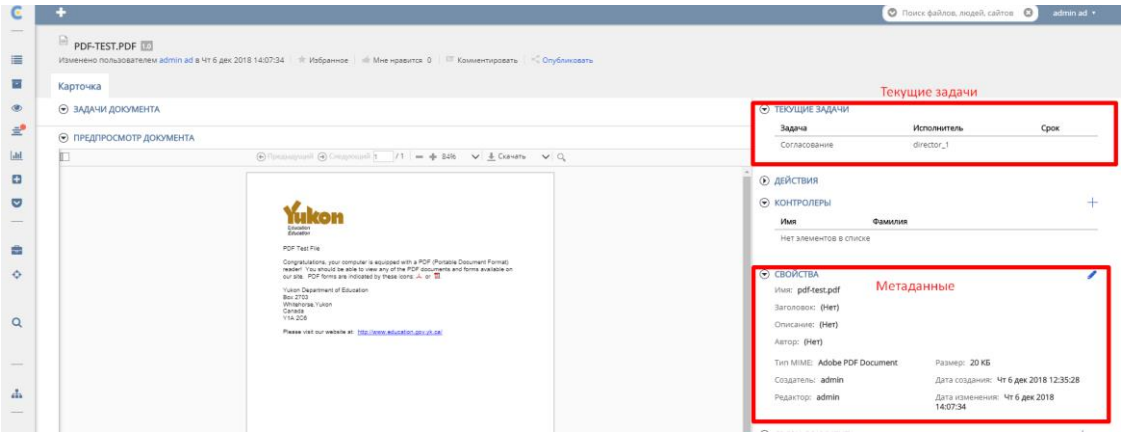


Рисунок 38 – Карточка документа

2.2 Взаимодействие с документом в процессе

Для взаимодействия с документом внутри процесса, необходимо на форме редактирования БП к событию старта назначить *listener*, который будет заполнять переменную *document*, выбранным документом. Для этого необходимо:

1. Перейти к редактированию процесса и выбрать событие старта (рисунок 39).

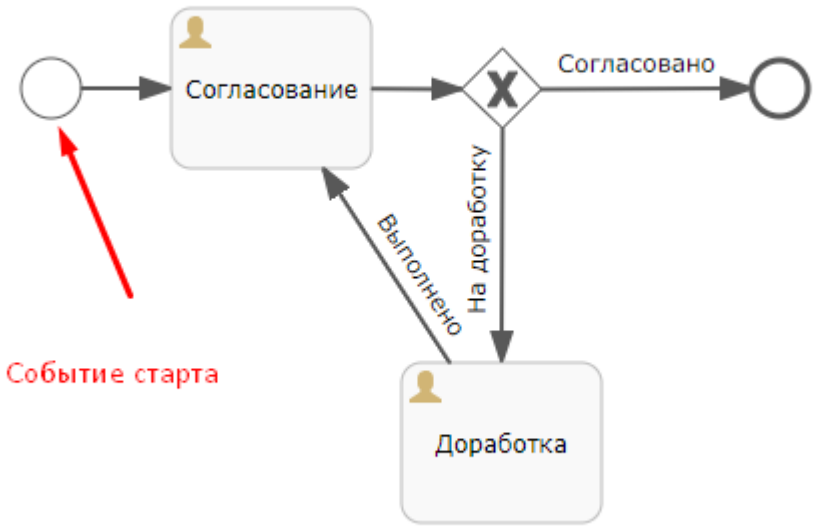


Рисунок 39 – Событие старта.

2. У события старта в свойствах элемента кликнуть Execution listeners

3. Добавить listener на событие start и в классе указать *ru.citeck.ecos.flowable.example.FlowableDocumentSetListener* (рисунок 40).
4. Сохранить.

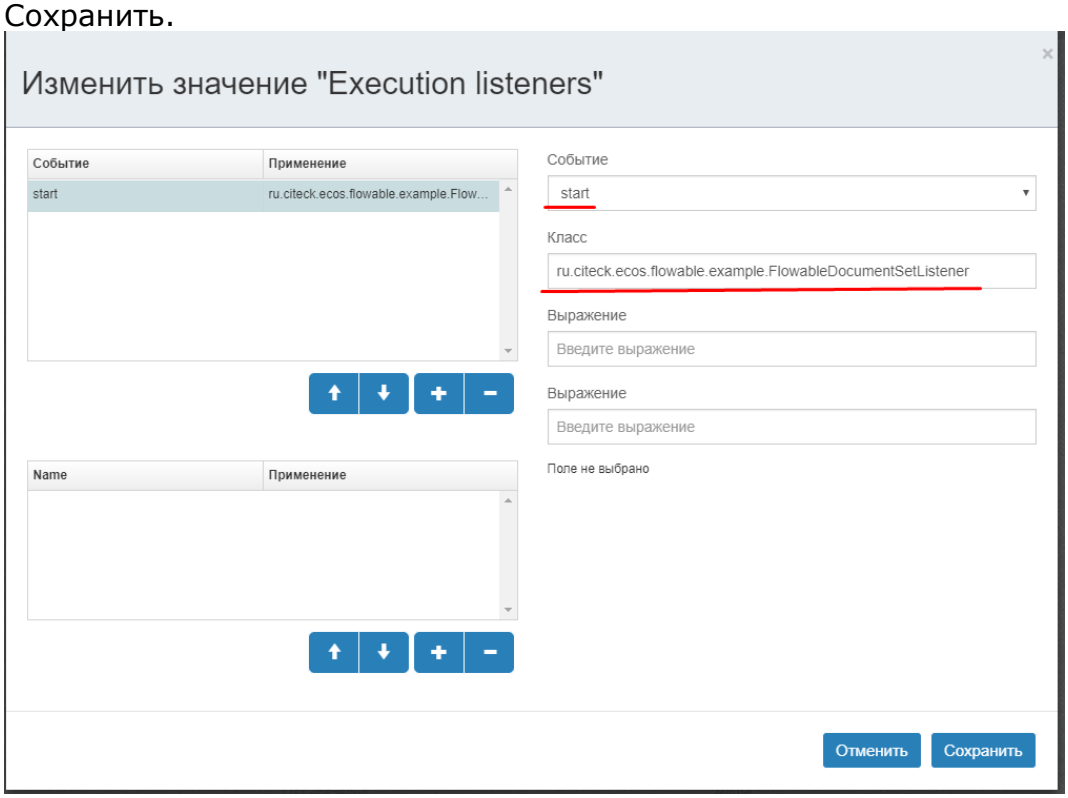


Рисунок 40 – Заполнение класса и события.

Теперь мы можем взаимодействовать с документом внутри процесса. Рассмотрим пару примеров.

2.2.1 Пример с условиями (flow condition)

В первом примере, в зависимости от формата прикрепленного документа (*pdf или docx*), будем назначать разные задачи после задачи согласования.

1. Создадим 2 задачи, обработка docx и обработка pdf, задачи будут назначаться в зависимости от указанного conditions (рисунок 41).

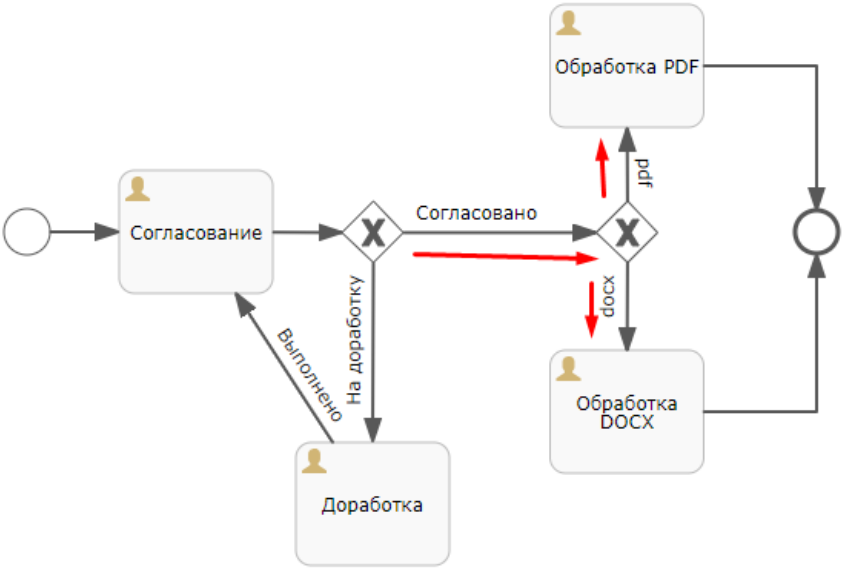


Рисунок 41 – Создано две задачи по обработке docx и pdf.

2. В условии перехода (flow condition) к задаче PDF укажем *${document.properties["cm:name"].contains(".pdf")}*
3. В условии перехода (flow condition) к задаче DOCX укажем *${document.properties["cm:name"].contains(".docx")}*

Указанное условие обращается к прикрепленному документу, берет его имя и проверяет наличие того, или иного расширения, и в зависимости от этого, назначаются разные задачи обработки документа (рисунок 42 и 43).

Рисунок 42 – Схема условий перехода.

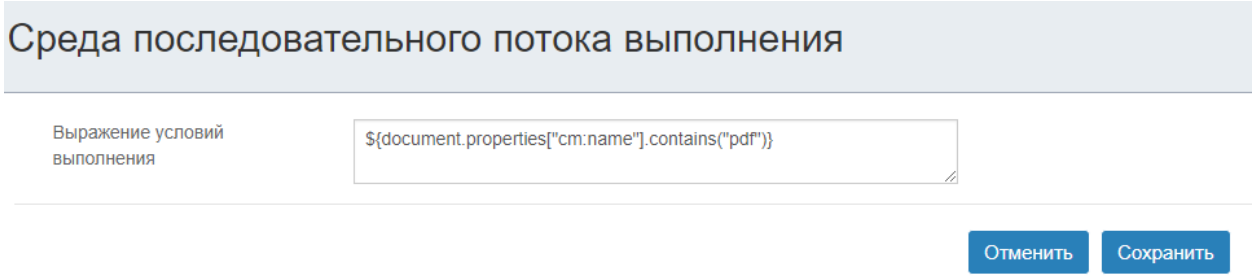


Рисунок 43 – Выражение условий выполнения.

2.2.2 Пример изменения метаданных документа

Во втором примере рассмотрим изменение метаданных документа через скрипт. В конце процесса заполним описание к документу (поле – *cm:description*) (рисунок 44).

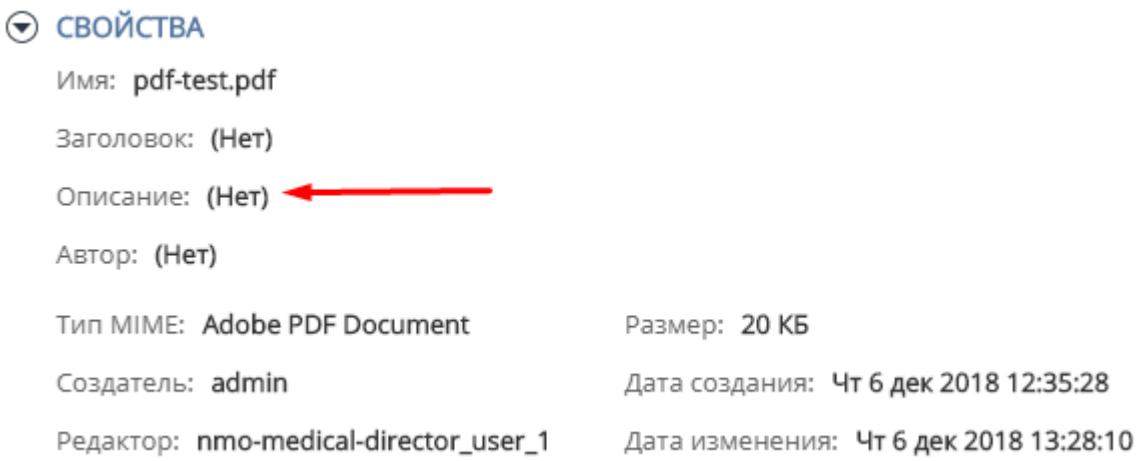


Рисунок 44 – Параметр «Описание».

1. Перед завершением процесса добавим script task (рисунок 45) с именем «Заполнение описания».

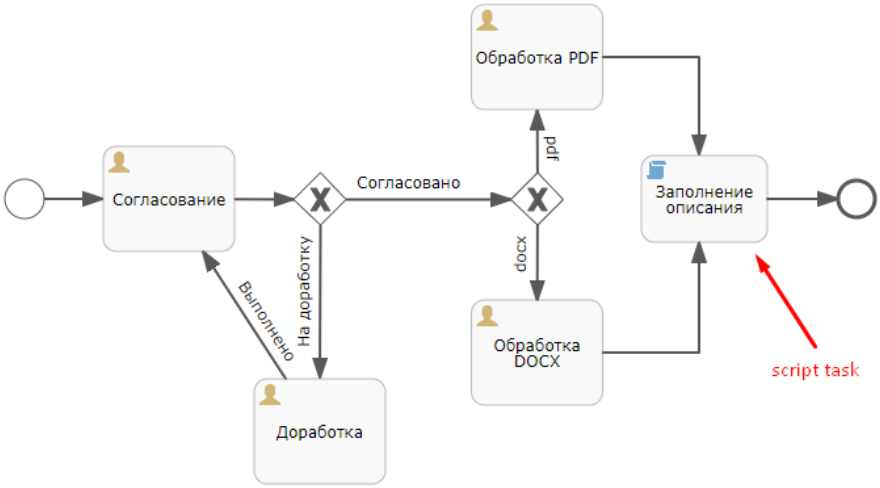


Рисунок 45 – script task.

2. В объекте script task необходимо заполнить поле *Script format*, значением *javascript*, поле *script* следующим скриптом:

```
var today = new Date().toLocaleDateString(undefined, {
   day: '2-digit',
   month: '2-digit',
   year: 'numeric'
});
var description = "Документ был полностью согласован датой: " + today;

document.properties["cm:description"] = description;
document.save();
```

Рисунок 46 – Заполнение описания.

В данном скрипте происходит формирование сообщения, состоящее из текста и текущей даты, в последствии сообщение сохраняется в поле *Описание (cm:description)* документа (рисунок 47). После выполнения задачи *Обработка PDF* или задачи *Обработка DOCX* вызывается созданный нами скрипт, после выполнения которого можно отследить изменения на карточке документа:

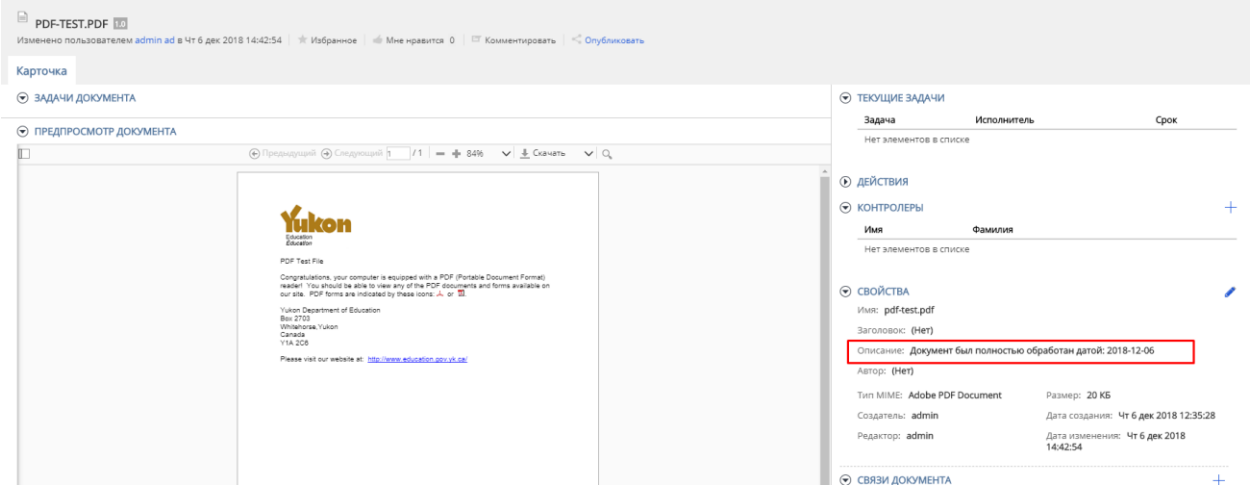


Рисунок 47 – Заполнение описания.

2.3 Отображение полей документа на форме задачи

На форму задачи есть возможность добавить поля из документа, это удобно, если необходимо в рамках задачи изменить какое-либо свойства документа.

Например, добавим на форму согласования обязательное заполнение поля *Заголовок* (cm:title). Для этого необходимо:

1. Открыть форму задачи согласования (рисунок 48).
2. Добавить текстовое поле на форму *Заголовок.*

Рисунок 48 – Форма согласования.

3. Заполнить id по определенному шаблону - _ECM_NSPREFIX_FIELD. Вначале идентификатора добавляется _ECM_. Затем идет префиксная запись QName свойства или ассоциации с заменой двоеточия (:) на подчеркивание (_). QName чувствителен к регистру. В данном случае id будет равен _ECM_cm_title (рисунок 49).

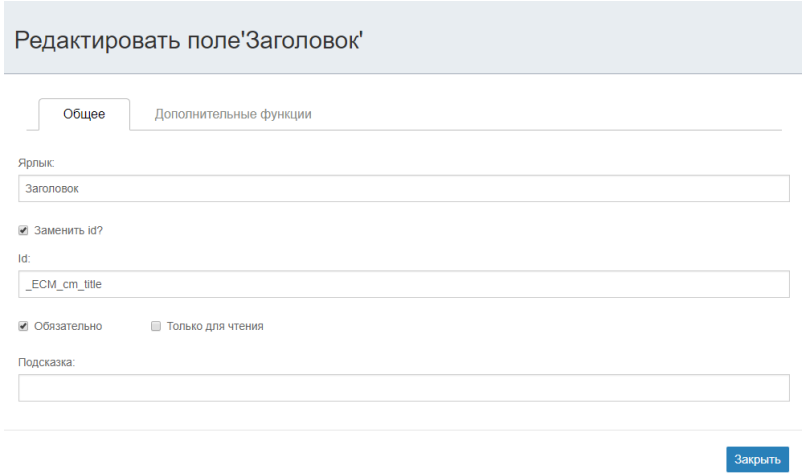
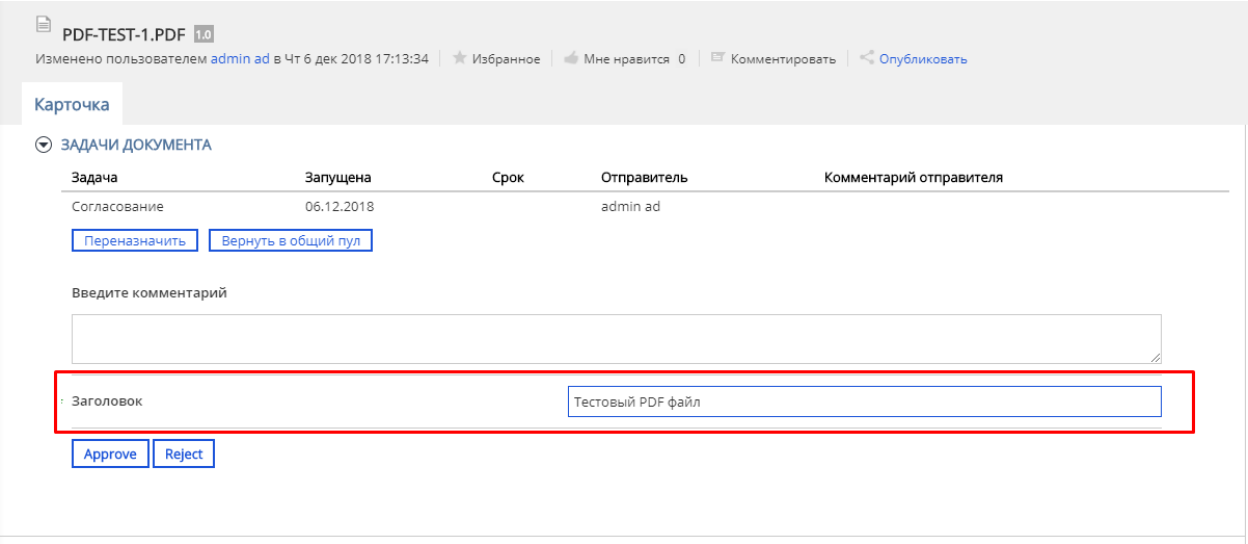


Рисунок 49 – Редактирование поле «Заголовок».

Теперь на форме задачи согласования будет отображатсья поле *Заголовок*, и при завершении задачи значение из данного поля сохранится в поле документа – cm:title (рисунок 50).



Рисуно - Поле «Заголовок» в карточке.

Выполняем задачу - значение сохранилось (рисунок 51):

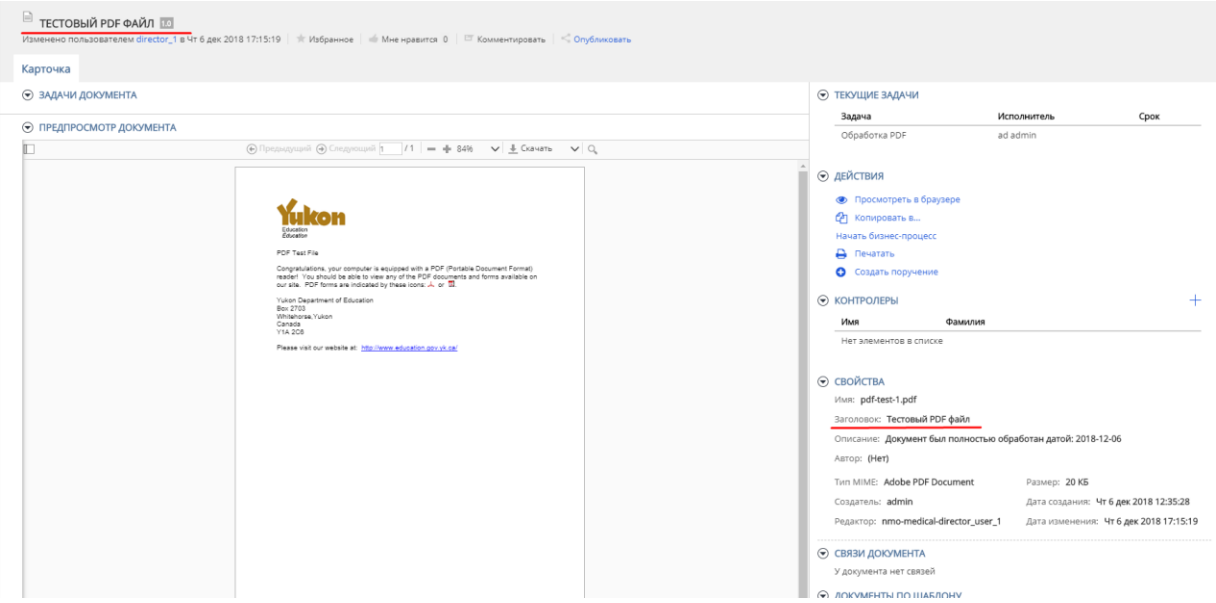


Рисунок 51 – Заголовок сохранен.

Приложение с процессом и формами, реализованное в данной главе - test_doc_app.zip

3. Взаимодействие с кейсом

Наиболее часто BPMN процесс используется вместе с кейс-менеджментом, это позволяет использовать преимущества двух технологий. Поэтому была реализована интеграция с кейс-менеджментом.

3.1 Предопределенные переменные

1. *case_status* - cm:name кейс статуса.
2. *case_status_before* – cm:name предыдущего кейс статуса.

Переменные кейс статусов можно использовать как flow condition:

```
${case_status_before == "ssg-approval"}
```

3.2 Сервисы

3.2.1 Case Completeness Service

JS имплементация сервиса (caseCompletenessServiceJS), доступна по ключу - *completeness*.
Пример использования - проверка выполнения чек листов в flow condition:

```
${completeness.isLevelCompleted(document, "workspace://SpacesStore/cl-hr-itn") &&
completeness.isLevelCompleted(document, "workspace://SpacesStore/cl-hr-bank-details")}
```

3.2.2 Case Status Service

JS имплементация сервиса (caseStatusServiceJS), доступна по ключу - *caseStatusService*.
Пример использования - установка кейс статуса через expression:

```
${caseStatusService.setStatus(document, "ssg-signed")}
```

3.2.3 Flowable Recipients Service

JS имплементация сервиса (flowableRecipientsServiceJS) доступна по ключу - *flwRecipients*.
Данный сервис позволяет получить данные по реципиентам из кейс роли.

Методы:

getRoleEmails(NodeRef document, String caseRoleName) - получение e-mail адресов из кейс роли. Применяется для заполнения поля To в mail task (рисунок 52).

${flwRecipients.getRoleEmails(document, 'ssg_initiator')}

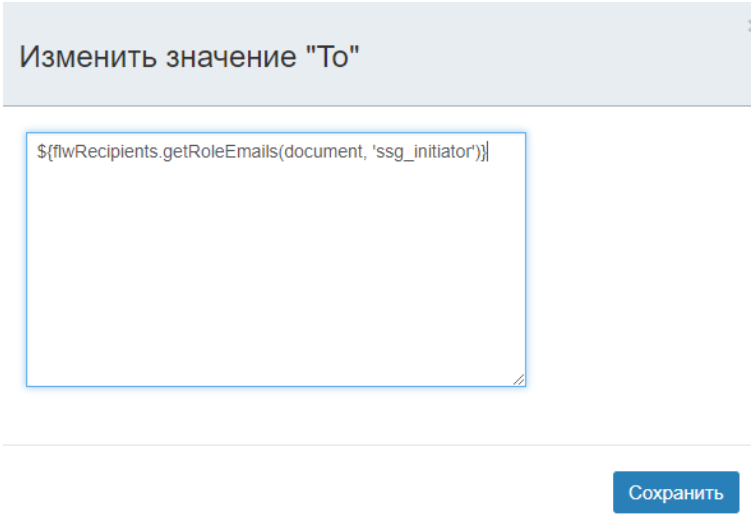


Рисунок 52 – Изменение значения.

getRoleGroups(NodeRef document, String caseRoleName) - получение групп из кейс роли. Применяется для заполнения candidate groups в user task assigments.

${flwRecipients.getRoleGroups(document, 'ssg_initiator')}

getRoleUsers(NodeRef document, String caseRoleName) - получение пользователей из кейс роли. Применяется для заполнения candidate users в user task assigments.

${flwRecipients.getRoleUsers(document, 'ssg_initiator')}

Результат заполнения на рисунке 53.

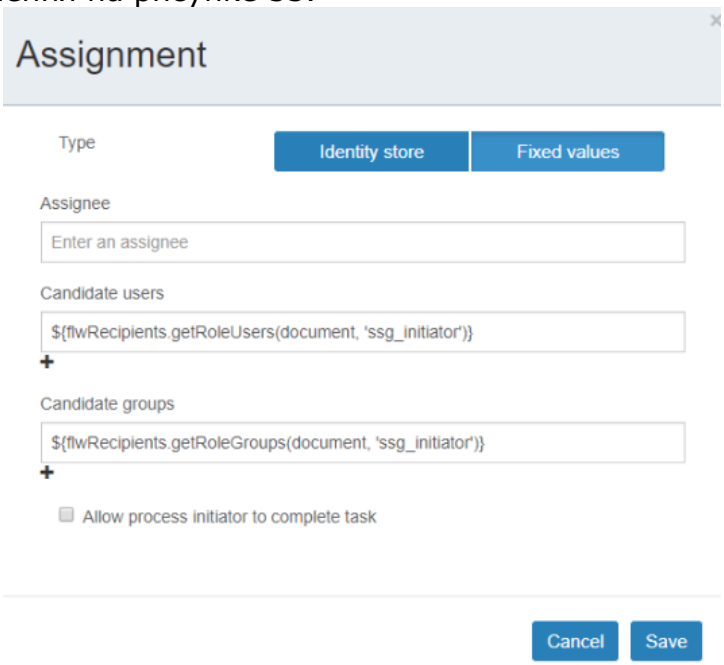


Рисунок 53 – Результат заполнения.

3.3 Listeners

3.3.1 Execution listeners

3.3.1.1 Flowable document set listener

Устанавливает переменную document в process execution, представляет собой ScriptNode документа, по которому идет процесс. Позволяет обращаться к свойствам, ассоциациям документа в expressions. См. 2.2 Взаимодействие с документом в процессе.

3.3.1.2 Flowable Check Completeness Levels ExecutionListener

Проверяет, выполнены ли заданные чек-листы, если нет, то выводит сообщения об ошибке, с перечислением невыполненных чек-листов.

Параметры:

- completenessLevels - перечисление nodeRef чек-листов, через ";"

Пример использования:

Кейс - необходимо проверить, что заполнены чек листы workspace://SpacesStore/cl-hr-uni-form-t-8, workspace://SpacesStore/cl-hr-uni-form-t-61, если чек листы не заполнены, то блокировать движение по процессу и вывести сообщение с информацией о незаполненных чек-листах. Для этого назначим execution listener:

ru.citeck.ecos.flowable.listeners.FlowableCheckCompletenessLevelsExecutionListener

на необходимый event, со следующими параметрами:

- completenessLevels = workspace://SpacesStore/cl-hr-uni-form-t-8;workspace://SpacesStore/cl-hr-uni-form-t-61 (nodeRefs чек-листов)

Результат заполнения на рисунке 54.

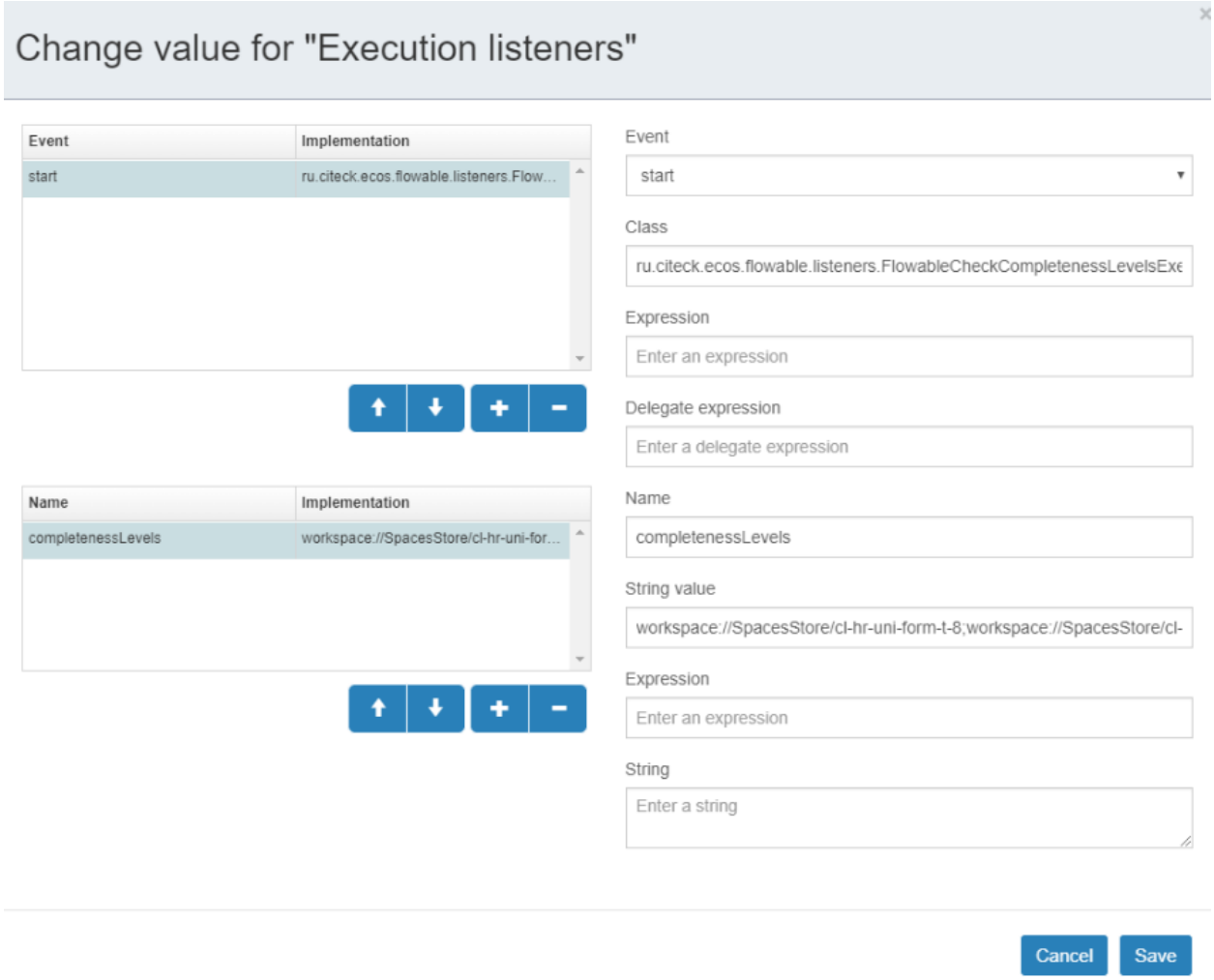


Рисунок 54 – Результат заполнения.

3.3.2 Tasks listeners

3.3.2.1 Flowable Check Mandatory Variable On Outcome Task Listener

Проверяет, что определенная переменная заполнена, при определенных *task outcomes*, если переменная не заполнена, то блокирует движение по процессу и выводит указанное сообщение.

Параметры:

- *variableId* - проверяемая переменная
- *outcomeId - task outcome id*
- *outcomes - task outcomes,* перечисленные через *";"*
- *message* - сообщение об ошибке, которое будет показано пользователю. Можно передать i18 *message key.*

Пример использования:

Кейс - Необходимо проверить, заполнен ли комментарий, при выполнении задачи с результатами *На доработку, Отменить*, если комментарий не заполнен, то блокировать движение по процессу и вывести пользователю сообщение о необходимости заполнения комментария.

Для этого, на необходимую задачу назначим *listener on complete*, класса *ru.citeck.ecos.flowable.listeners.FlowableCheckMandatoryVariableOnOutcomeTaskListener*, со следующими параметрами:

- *variableId = cd_signer_sign_comment* (id переменной комментария, которую нужно проверить на заполненность)
- *outcomeId = form_cd_signer_sign_outcome* (id переменной результата выполнения задачи)
- *outcomes = На доработку;Отменить* (результаты задачи, при которых необходимо осуществлять проверку)
- *message = wfcf_confirmworkflow.message_comment_is_empty* (*message key* сообщения об ошибки, в данном случае будет содержать локализованное сообщение: "Необходимо ввести комментарий")

Результат заполнения на рисунке 55.

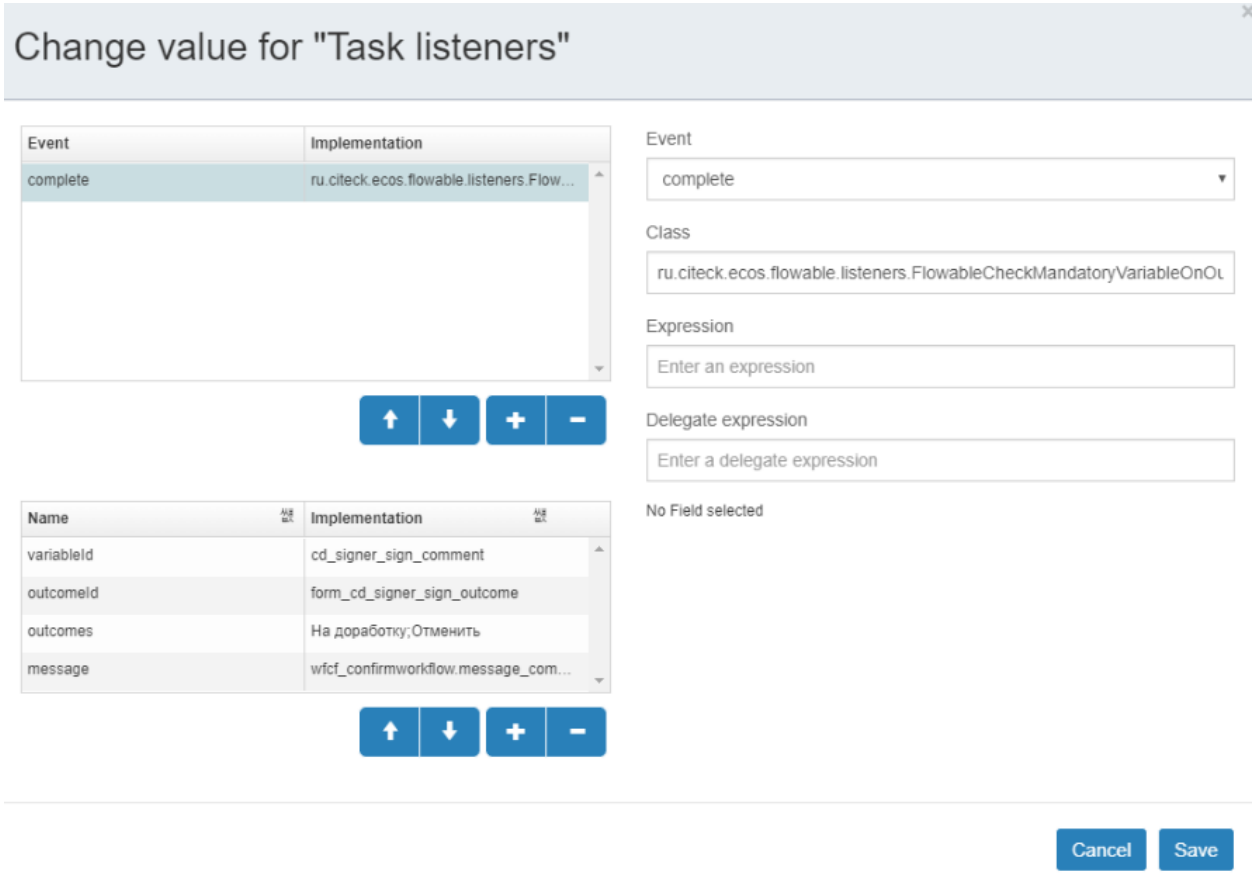


Рисунок 55 – Результат заполнения.

3.3.2.2. Check Lists Task Listener

Если в кейсе не запущен бизнес-процесс *flowable* тогда во вкладке "Документы" все документы отмечены красным (как и раньше).

Если в кейсе есть активные бизнес-процес(ы) *flowable*, то проверяются текущие активные задачи на наличие *listener'a* на событии "*complete*" *ru.citeck.ecos.flowable.listeners.CheckListsTaskListener.*

Если таких *listener'ов* не найдено, то все незагруженные документы отображаются серым цветом. Если найдены *CheckListsTaskListener,* то проверяется свойство "*lists*", в котором перечислены чек-листы. Все документы из указанных чек-листов отображаются красным цветом на вкладке "Документы".

Если у *Listener'a* задано свойство *outcomesToCheck* и корректный *outcomeField* (поле с результатом выполнения задачи), то при невыполненных чек-листах задача не завершится.

Если у *Listener'a* свойство *outcomesToCheck* не задано, то чек листы проверяются при любом *outcome.*

Если у *Listener'a* установлено свойство *checkEnabled* в *false,* то проверки чек-листов при завершении задачи не происходит, но подсветка документов красным во вкладке "Документы" по-прежнему будет работать.

Пример конфигурации на рисунке 56:

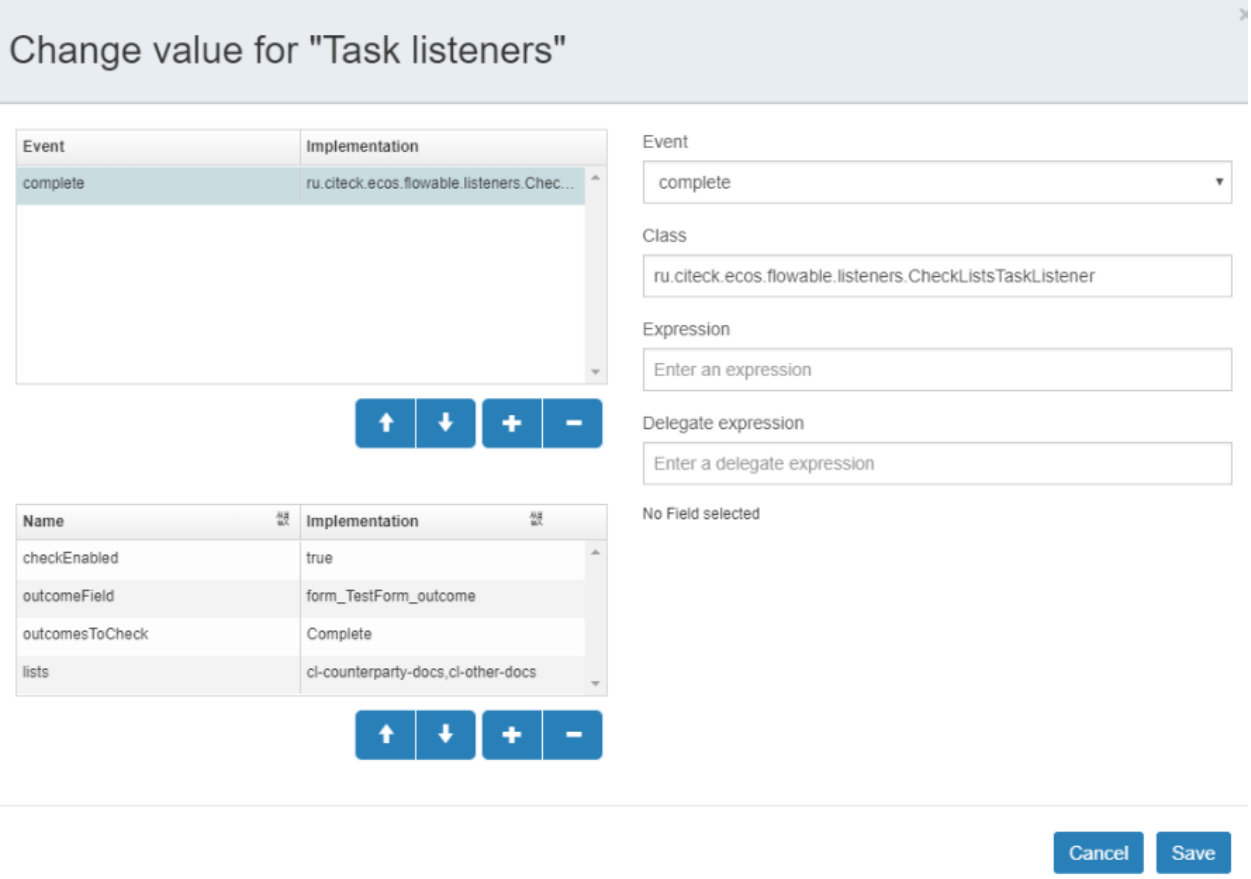


Рисунок 56 – Пример конфигурации.

4. Согласование задач через email

Предполагается, что выполнение задач через емейл уже сконфигурировано в контексте сервера (джоба, сервис, провайдер, и тд) согласно статье - [Выполнение задач по E-mail.]

Для того, чтобы в BPMN отправить сообщение с вердиктами задачи, необходимо:

1. Создать в процессе Mail task, заполнить его необходимыми данными для отправки (при этом получателей рекомендуется указать тех же, что и в кандидатах задачи), присвоить уникальный ***id***.
2. В html шаблоне вставить код формирования кнопок выполнения задач.
3. В задаче, для которой необходимо включить возможность выполнения через емейл, добавить task listener на событие *create* с

классом *ru.citeck.ecos.taskmailapproval.flowable.listeners.ExecuteApprovalServiceMailTaskListener*, добавить переменную *emailId* с *id* Mail task, заполненным в пункте 1.

Пример:

Кейс

Необходимо добавить возможность выполнения через емейл для задачи, которая назначается на роль из кейса "*nmo-compliance*", и имеет 3 результата: *Согласовано, Отказ, Отправить* на доработку инициатору.

Решение

1. Создаем mail task, с id *compliance_email_1*.

2. В получателях указываем роль nmo_comliance - *${flwRecipients.getRoleEmails(document, 'nmo-compliance')}*

3. В html шаблоне добавляем вердикты:

```
<div style="width: 100%; font-family:'GE Inspira',sans-serif;">
    <fieldset>
        <p style="font-weight:bold">Вердикты:</p>
        <div>
            <p><a
href="mailto:${taskMailData.compliance_email_1.email_to}?Subject=CA-
${taskMailData.compliance_email_1.task_id}-Согласовано-
${taskMailData.compliance_email_1.task_token}&body=Вердикт: Согласовано
%0D%0A %0D%0A Поле комментарий обязательно для заполнения %0D%0A %0D%0A
$[comment] %0D%0A %0D%0A
${taskMailData.compliance_email_1.default_task_comment} %0D%0A %0D%0A
[comment]"
                target="_top">Согласовано</a></p>
            <p><a
href="mailto:${taskMailData.compliance_email_1.email_to}?Subject=CA-
${taskMailData.compliance_email_1.task_id}-Отказ-
${taskMailData.compliance_email_1.task_token}&body=Вердикт: Отказ %0D%0A
%0D%0A Поле комментарий обязательно для заполнения %0D%0A %0D%0A
$[comment] %0D%0A %0D%0A
${taskMailData.compliance_email_1.default_task_comment} %0D%0A %0D%0A
[comment]"
                target="_top">Отказ</a></p>
            <p><a
href="mailto:${taskMailData.compliance_email_1.email_to}?Subject=CA-
${taskMailData.compliance_email_1.task_id}-Отправить на доработку
инициатору-${taskMailData.compliance_email_1.task_token}&body=Вердикт:
Отправить на доработку инициатору %0D%0A %0D%0A Поле комментарий
обязательно для заполнения %0D%0A %0D%0A $[comment] %0D%0A %0D%0A
${taskMailData.compliance_email_1.default_task_comment} %0D%0A %0D%0A
[comment]"
                target="_top">Отправить на доработку инициатору</a></p>
        </div>
    </fieldset>
</div>
```

Внутри *div* необходимо сформировать ссылки, каждая ссылка соответствует одному вердикту в задаче. Ссылки формируются по правилу:

href="mailto:${taskMailData.ID_мейл_таска.email_to}?Subject=CA-${taskMailData.ID_мейл_таска.task_id}-ID_вердикта_по_задаче-${taskMailData.ID_мейл_таска.task_token}&body=Вердикт: имя_вердикта_по_задаче%0D%0A %0D%0A Поле комментарий обязательно для заполнения %0D%0A %0D%0A $[comment] %0D%0A %0D%0A ${taskMailData.ID_мейл_таска.default_task_comment} %0D%0A %0D%0A [comment]"

4. В задаче, добавляем task listener, с классом *ru.citeck.ecos.taskmailapproval.flowable.listeners.ExecuteApprovalService MailTaskListener* и в переменой *emailId* указываем *compliance_email_1*. Рисунок 57.

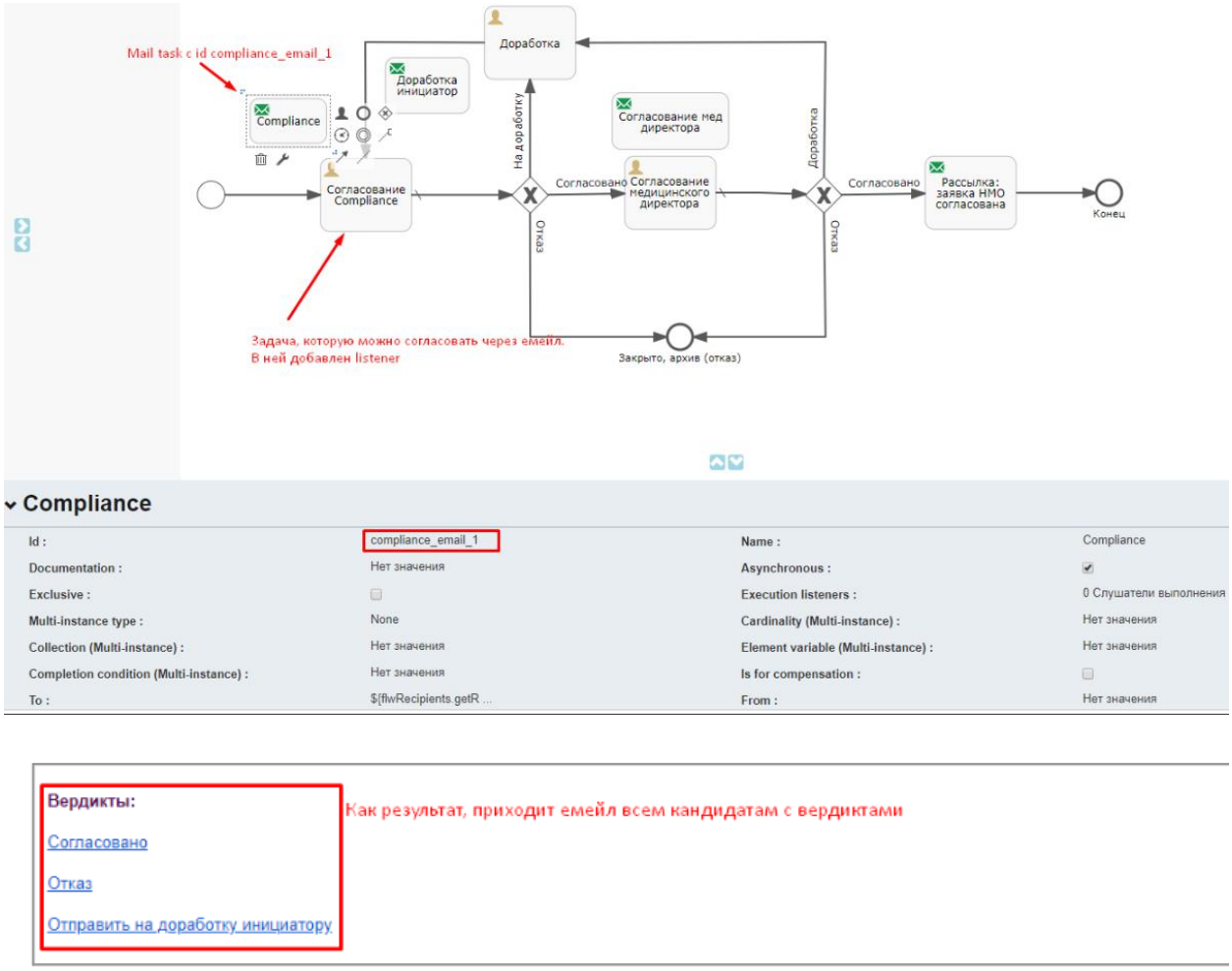


Рисунок 57 – Результат выполнения.

5. Локализация

Ресурсные файлы с локализацией определяются на стороне *repo*.

5.1 Outcomes

При создании формы задается ключ формы (рисунок 58):

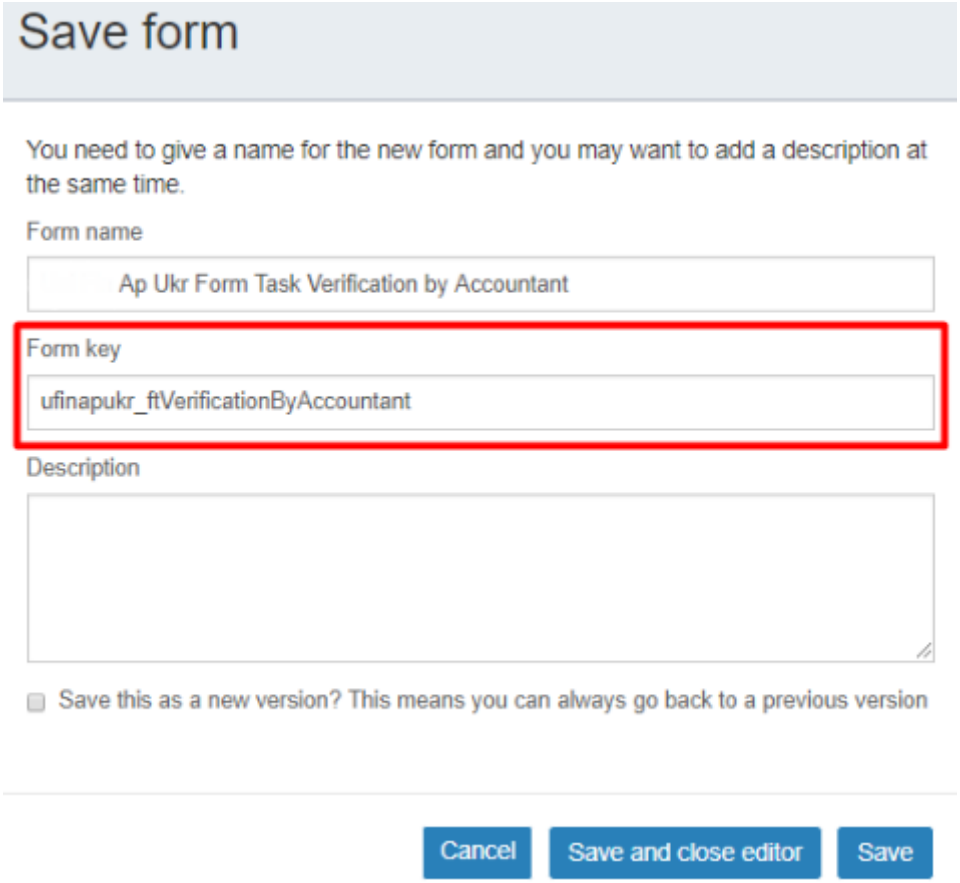


Рисунок 58 – Ключ формы

И возможные *outcomes* задач (рисунок 59):

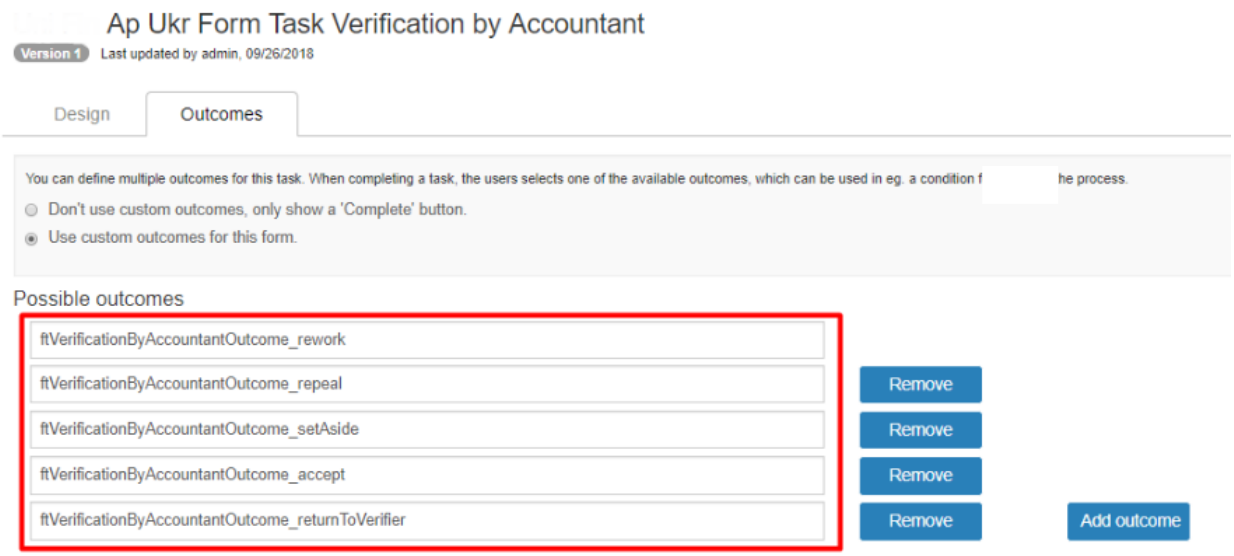


Рисунок 59 – Outcomes задачи.

Для локализации *outcome* строка в файле должна быть следующего формата:

flowable.form.button.formKey.id.label, где *formKey* - "Form key" формы (в случае если он не задан используется "Form name"), *id* - идентификатор *outcome*, задаваемый в "Possible outcomes".

Далее необходимо задать ресурсные файлы на стороне *repo* (рисунок 60):

```
flowable.form.button.ufinapukr_ftVerificationByAccountant.ftVerificationByAccountantOutcome_rework.label=На доработку
flowable.form.button.ufinapukr_ftVerificationByAccountant.ftVerificationByAccountantOutcome_repeal.label=Аннулировать
flowable.form.button.ufinapukr_ftVerificationByAccountant.ftVerificationByAccountantOutcome_setAside.label=Отложить
flowable.form.button.ufinapukr_ftVerificationByAccountant.ftVerificationByAccountantOutcome_accept.label=Согласовать
flowable.form.button.ufinapukr_ftVerificationByAccountant.ftVerificationByAccountantOutcome_returnToVerifier.label=Вернуть верификатору
```

Рисунок 60 – Ресурсные файлы.

Если для *outcome* не задана локализация, на кардлете задач документа отобразится его *id* (рисунок 61).

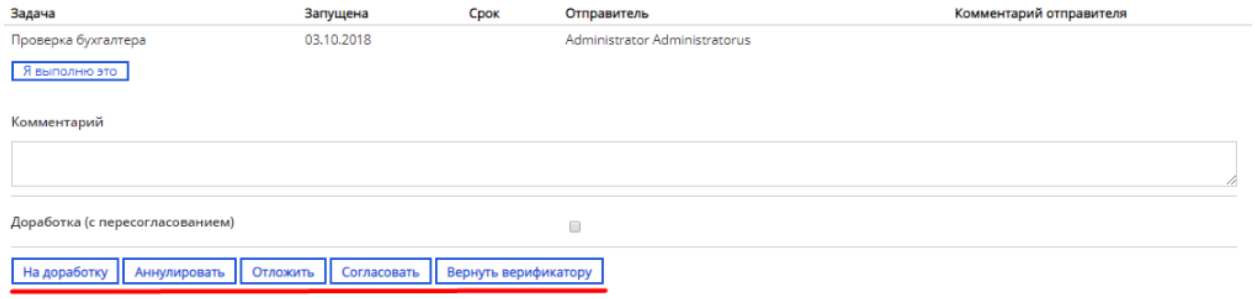


Рисунок 61 – Кардлет задач.

5.2 Поля

Так же есть возможность локализовать заголовки полей формы задачи (рисунок 62):

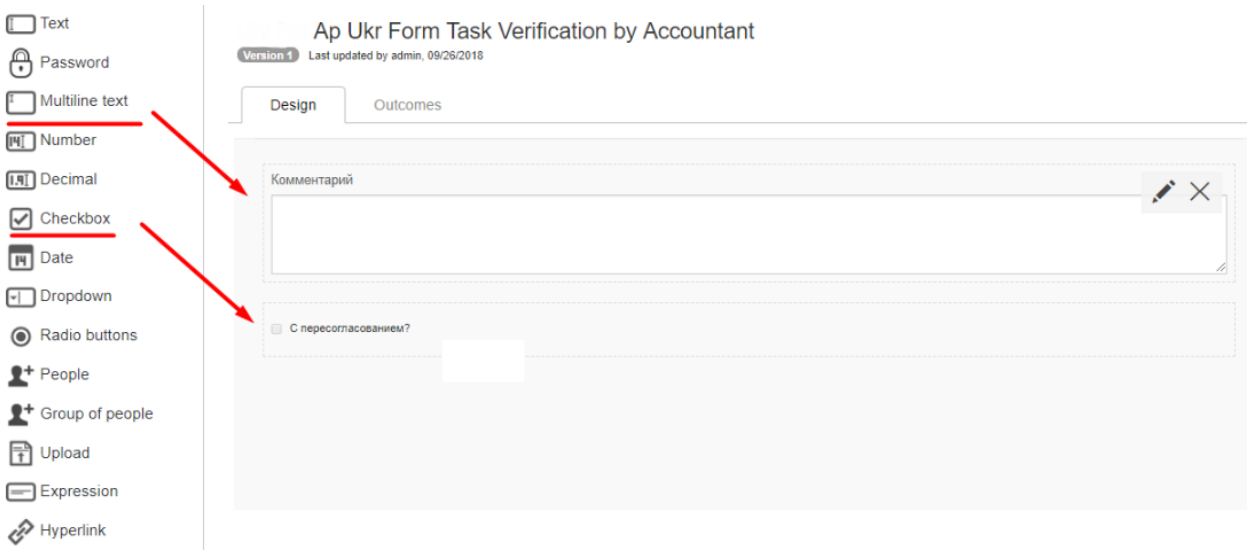


Рисунок 62 – Локализация заголовка.

Для этого строка в файле локализации должна быть следующего формата(рисунок 63 и 64):

flowable.form.field.id.label, - где *id* - "Id" поля(в случае если он не задан будет использоваться *name* поля(*Label*)).

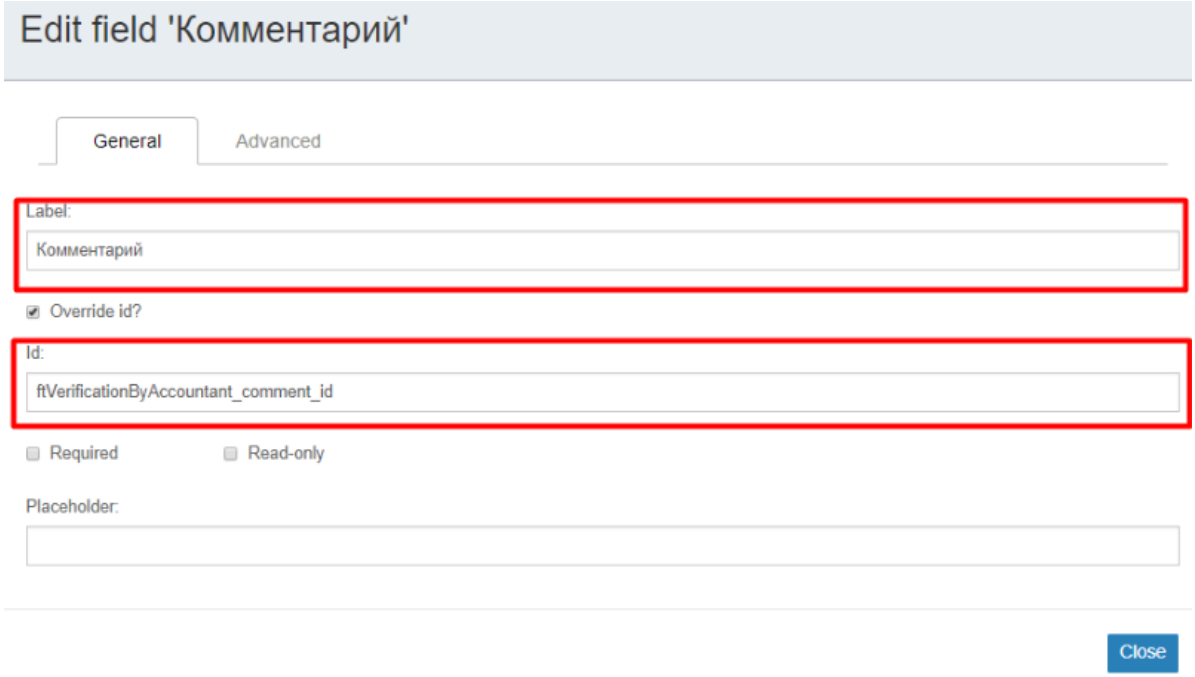


Рисунок 63 – Формат файла локализации.

```
flowable.form.field.ftVerificationByAccountant_checkbox_id.label=Доработка (с пересогласованием)
flowable.form.field.ftVerificationByAccountant_comment_id.label=Комментарий
```

Рисунок 64 – Формат файла локализации

Если для *field* не задана его локализация, на кардлете задач документа отобразится его *label*.

5.3 Названия задач

Для локализации названия задачи используется поле "*Id*" задачи, если локализация не задана или не задан "*Id*", используется поле "*Name*"(рисунок 65).

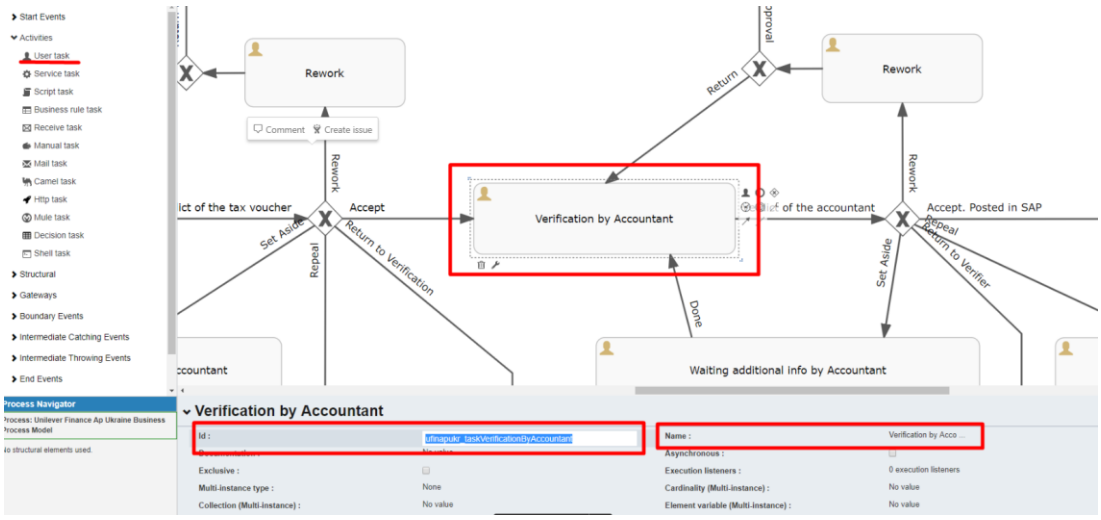


Рисунок 65– Локализации названия задачи.

Строка в файле локализации должна быть следующего формата (рисунок 66): *flowable.task.id.title*, - где *id* - "Id" задачи(в случае если он не задан, строка должна будет равна полю "*Name*" задачи).

```
flowable.task.ufinapukr_taskVerificationByAccountant.title=Проверка бухгалтер
```

Рисунок 66 – Формат строки.

citeck

УПРАВЛЯЙТЕ БИЗНЕС-ПРОЦЕССАМИ ЭФФЕКТИВНО

С КОМПАНИЕЙ CITECK

INFO@CITECK.RU
8 800 500-85-36