

Стандарты моделирования бизнес-процессов

Как придать решению законченный вид



ESMQ

Система управления бизнес-процессами и эффективностью

Оглавление

Глава 1. Проектирование решения.....	3
1.1 Концепция решения	3
1.2 Объектная модель.....	3
1.3 Интеграция с внешними системами.....	9
1.4 Техническое задание	10
Глава 2. Моделирование бизнес-процессов.....	12
2.1 Моделирование решения	12
2.2 Графическая модель процесса.....	15
2.3 Запуск бизнес-процессов	20
2.4 Настройка стартового события.....	22
2.5 Настройка пользовательских задач.....	24
2.6 Настройка переходов.....	29
2.7 Сценарии	31
2.8 Администрирование процессов	34
Глава 3. Контроль и мониторинг	37
3.1 Оперативный контроль.....	38
3.2 Заключительный контроль.....	40
Глава 4. Внутренний портал	44
4.1 Настройка внутреннего портала	44
4.2 Настройка интерфейсов	44
Глава 5. Документирование решения	48
Глава 6. Обучение пользователей.....	49
Глава 7. Чек-лист	50
Глава 8. Полезные ресурсы	51

Введение

Эта книга предназначена для тех пользователей, которые самостоятельно осваивают систему управления бизнес-процессами ELMA, а также для специалистов, которые планируют профессионально заниматься внедрением данной системы. Основная задача этой книги – познакомить с основными инструментами системы, которые позволят придать решению законченный вид. Соблюдение приведенных здесь требований и рекомендаций позволит улучшить качество вашего решения, а также позволит избежать некоторых ошибок.

Данная книга предполагает, что пользователь владеет базовыми навыками работы с системой ELMA, которые описаны в кратком руководстве по Платформе ELMA BPM.

Полный перечень кратких руководств по системе:

- [Краткое руководство по Платформе ELMA BPM;](#)
- [Краткое руководство по использованию внутреннего портала ELMA;](#)
- [Краткое руководство по приложению ELMA ECM+;](#)
- [Краткое руководство по приложению ELMA CRM+;](#)
- [Краткое руководство по приложению ELMA Проекты+;](#)
- [Краткое руководство по приложению ELMA KPI.](#)

Подробное описание функций и настроек содержит справка по системе. Справка доступна на web-ресурсе <http://www.elma-bpm.ru/kb/help/>.

Решение многих технических вопросов описаны в базе знаний ELMA по адресу <http://www.elma-bpm.ru/kb/>. База знаний постоянно пополняется специалистами компании.

Эта книга не является справочным руководством, а скорее набором рекомендаций к грамотной настройке элементов системы.

Глава 1. Проектирование решения

Проектирование решения – это процесс, позволяющий спланировать конечный образ системы и ее элементов, предусмотреть различные сценарии использования. Весь процесс проектирования можно разделить на несколько этапов: составление концепции решения, построение объектной модели, написание технического задания.

1.1 Концепция решения

Концепция решения – это документ, описывающий то, как ДОЛЖНО выглядеть готовое решение. Концепция не содержит в себе подробного описания технической реализации, а скорее описывает конечный образ решения и путь его реализации. Концепция решения строится вокруг конкретных проблем и сценариев использования системы, решение которых и является целью внедрения системы.

При проектировании решения необходимо помнить, что главный инструмент аналитика – логика. В своем решении старайтесь опираться на существующие условия: бюджет и сроки, применимость решения, здравый смысл.

В концепцию решения должны быть включены следующие аспекты:

- цели и задачи внедрения;
- сценарии использования системы;
- роли;
- архитектура системы;
- объектная модель;
- контуры (этапы реализации).

1.2 Объектная модель

Объект – это сущность, имеющая определенный набор заданных параметров и свойств. Как вам должно быть известно из краткого руководства по Платформе ELMA BPM, все данные в системе ELMA хранятся в виде объектов и взаимосвязей между ними.

Объектная модель – это совокупность объектов и перечислений, существующих в системе, их свойств, параметров и взаимосвязей. Объектная модель определяет структуру хранения пользовательских данных в системе. На Рис. 1 приведен пример взаимосвязей объектов системы и их структуры.

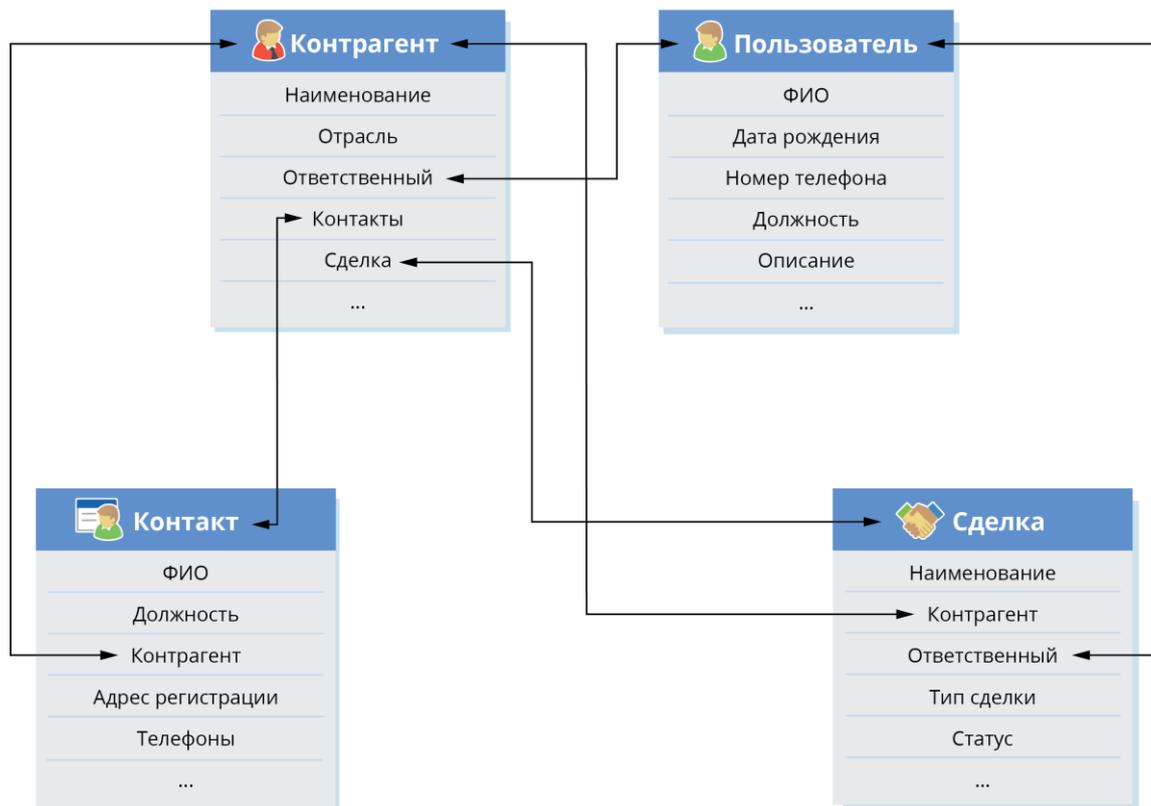


Рис. 1. Объектная модель

Объекты в системе ELMA могут быть созданы путем наследования от родительского объекта. При наследовании происходит полное копирование всех свойств родительского объекта (базового класса) с возможностью добавления новых свойств, форм и пр. На Рис. 2 приведен пример наследования объектов. Синим цветом выделен базовый класс (родительский объект), а голубым – унаследованные от него объекты. В объектах-наследниках черным шрифтом отмечены свойств унаследованные от родительского объекта, а синим – собственные свойства этого объекта.

Пользовательские объекты и типы документов лучше помещать в отдельную пользовательскую папку.

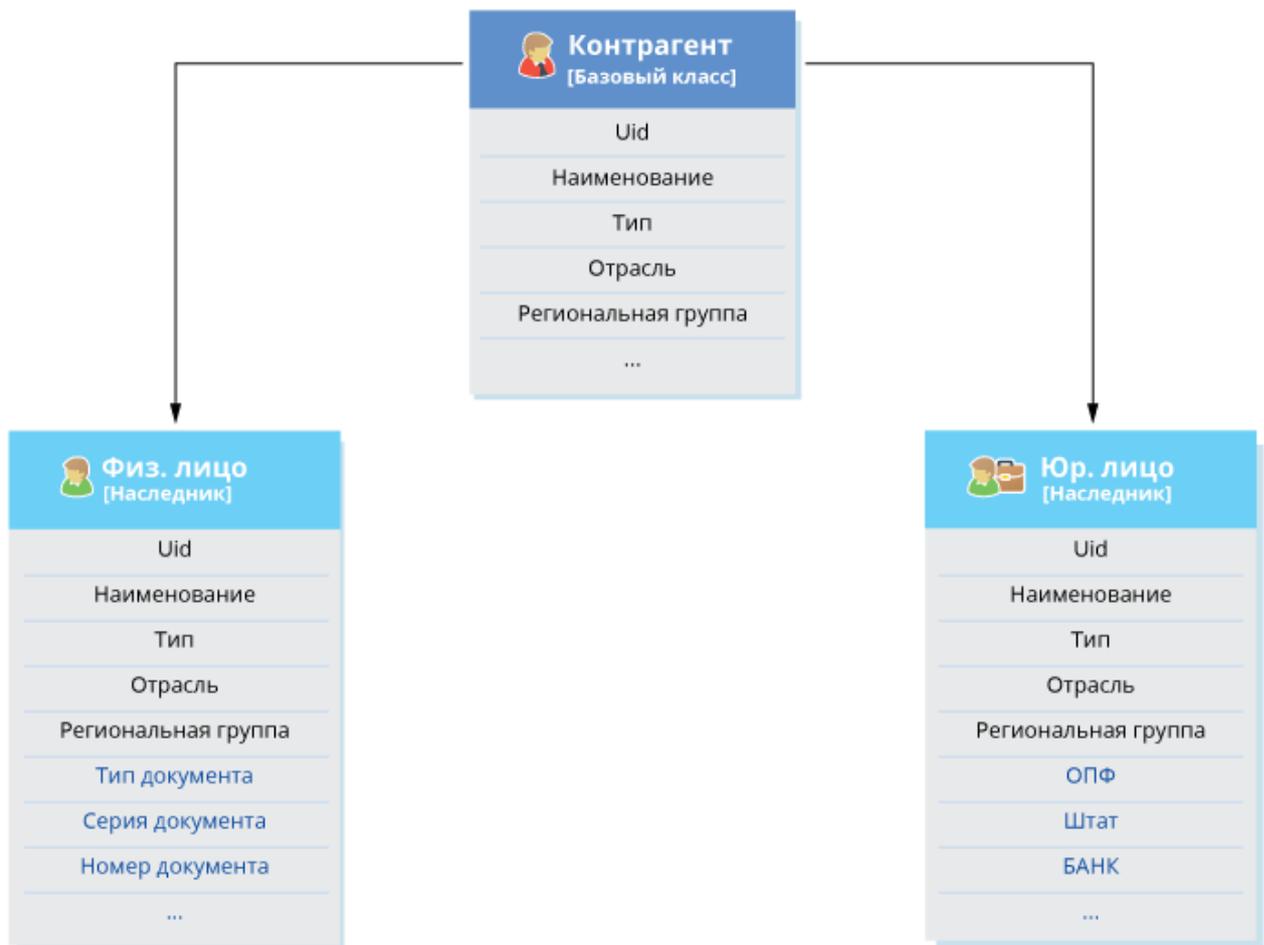


Рис. 2. Наследование объектов

Следует отметить, что выбор базового класса доступен только на этапе создания объекта. После первой публикации и перезагрузки сервера удалить или изменить базовый класс в объекте будет невозможно. Использовать родительский объект можно там, где часть свойств у документов совпадает (как на примере выше).

Отображаемое имя всех объектов и атрибутов в системе должно быть лаконичным и корректным, но не должно содержать сокращений.

Имя класса (имя свойства) должно быть переведено на английский язык. Транслитерация в названии не допускается. Для схожих объектов и/или свойств старайтесь использовать одинаковые (или схожие по смыслу) наименования.

В таблице ниже приведен пример свойств с примером корректных наименований (Таблица 1).

Таблица 1. Пример наименований параметров

Название переменной	Правильно	Неправильно
Отображаемое имя	Имя свойства	Имя свойства
Инициатор	Initiator	Iniciator
Заголовок процесса	ImprovedProcessHeader	ZagolovokProcessa
Экземпляр улучшаемого процесса	ImprovedWorkflowInstance	EHkzemplyarUluchshaemogoProcessa
Задача	Task	Zadazha
Описание	Description	Opisanie
Владелец процесса	ProcessOwner	VladelecProcessa
Исполнитель	Worker	Ispolnitel
Дата исполнения	WorkEndDate	Datalspolneniya
Комментарий владельца процесса	ProcessOwnerComment	KommentarijVladelcaProcessa
Резолюция исполнителя	WorkerComment	Rezolyuciyalspolnitelya

При создании пользовательских объектов и при расширении системных объектов пользовательскими переменными необходимо добавлять в название префикс проекта, в рамках которого осуществляется расширение системы. Например, объект типа "Входящий документ", разработанный на проекте "Automatic Solutions", может иметь наименование "AS_IncomingDocument" (Рис. 3). Это необходимо для того, чтобы при дальнейшем обновлении системы эти переменные и объекты не были случайно перезаписаны.

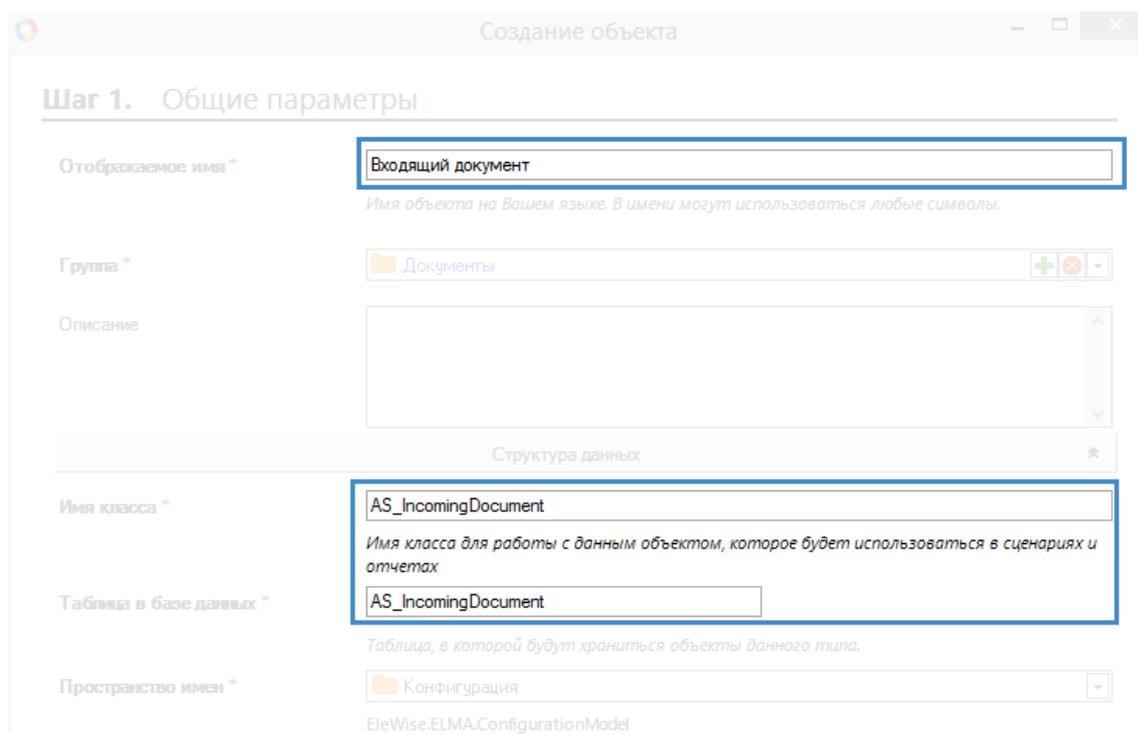


Рис. 3. Пример использования префикса проекта

При создании справочников рекомендуется добавить свойства по умолчанию (**Наименование**, **Дата создания**, **Автор создания**, **Дата изменения**, **Автор изменения**) (Рис. 4). При создании свойств по умолчанию проблема открытых UID автоматически устраняется.

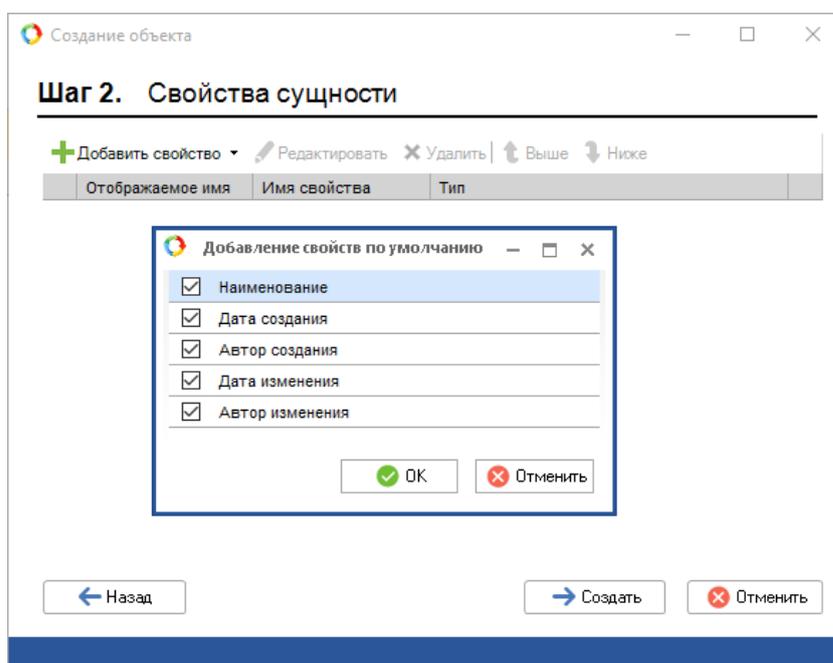


Рис. 4. Пример добавления свойств по умолчанию

Если хотя бы у одного поля справочника добавлено правило на отображение любого поля, базовая переменная "Uid записи" также автоматически скрывается (Рис. 5).

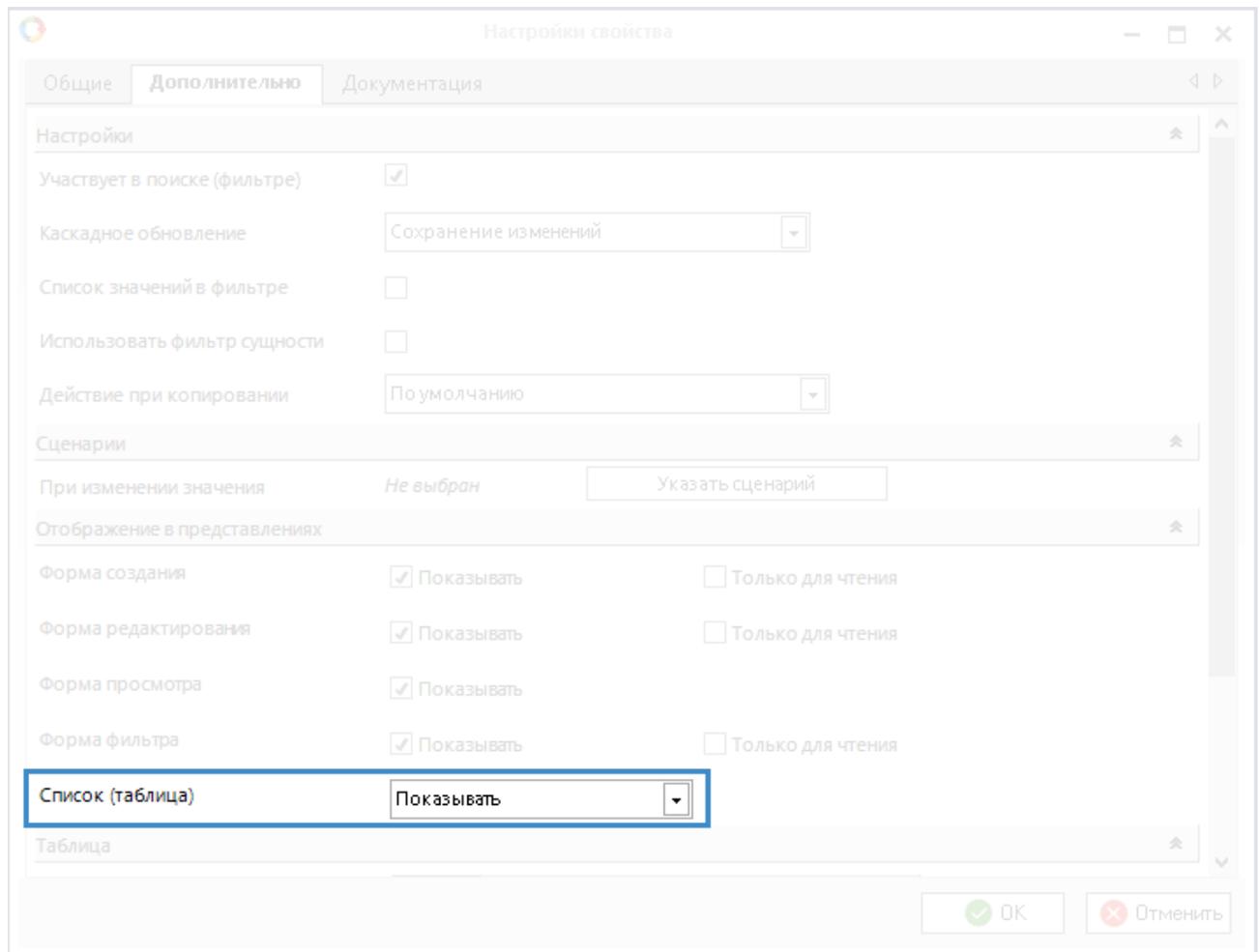


Рис. 5. Отображение в представлениях

Опубликованный объект не может быть удален из системы. Это необходимо для корректной работы системы, и связано с поддержкой версионности процессов. По той же причине все процессы, документы, их атрибуты и пр. подвергаются только "мягкому удалению". Другими словами, объект отмечается как удаленный и не отображается у пользователей, но в системе все же остается. Поэтому перед созданием объектов в процессе внедрения необходимо сперва спроектировать наиболее полную объектную модель. Для лучшего восприятия и самоконтроля можно изобразить эту модель в виде схемы.

Перед первой публикацией объекта убедитесь, что все переменные, их настройки – типы, названия классов, наследование – корректны. Чтобы понять, насколько полной получилась модель, можно взглянуть на нее с точки зрения

различных ролей, которые предполагаются в системе, различных сценариев использования.

Заранее продумайте интерфейсы пользователей с точки зрения их роли в системе. Набор необходимых инструментов: отчеты, портлеты, фильтры, метрики и показатели.

Если какой-либо процесс, объект, тип документа или переменная больше не используются, вместо удаления лучше переместить их в отдельную папку-архив, а отображаемое название изменить, добавив "_" в начале. Это позволит без труда их отличить. При этом **Имя свойства/объекта** в БД изменять не следует, так как при изменении уже опубликованного объекта в базе данных создается еще одна таблица, а не изменяется существующая.

Неиспользуемый процесс также можно переместить в отдельную папку-архив, предварительно сняв галочку запуска процесса из веб-приложения (Рис. 6).

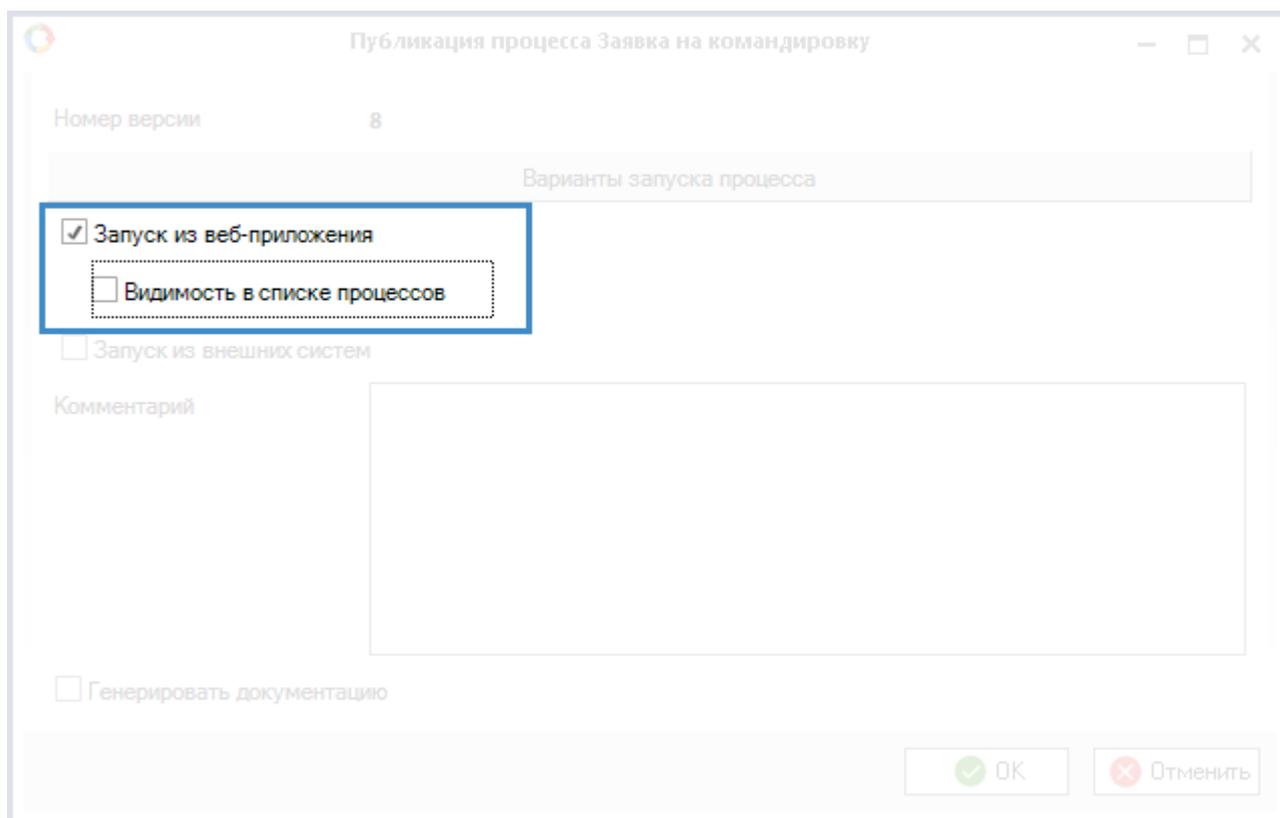


Рис. 6. Окно публикации процесса

1.3 Интеграция с внешними системами

На этапе проектирования необходимо выяснить наличие внешних систем, с которыми, возможно, предстоит интегрировать систему ELMA.

При планировании интеграции необходимо понимать цель и сложность такой интеграции. Интеграция имеет смысл только там, где польза от реализации выше требуемых усилий. Не стоит забывать также и о необходимости дальнейшего сопровождения решения.

Широкий инструментарий ELMA позволяет объединять в себе различные функции – иногда более правильным решением будет перенести функционал внешних систем в ELMA. Например, модуль ELMA CRM позволяет вести работу с контрагентами. Таким образом, можно избежать дублирования функционала, перенеся данные и возможности из действующей CRM системы в ELMA.

Не всегда можно заменить функционал специализированных внешних систем. Например, система ELMA не может быть использована в качестве учетной системы, банковских систем (АБС, СППР и т.д.), системы технологического управления (управление станками). Нужно помнить, что система ELMA предназначена для движения информации в рамках бизнес-процессов и не подходит для хранения больших объемов информации.

При оценке трудоемкости интеграции необходимо определить точки интеграции и логику передачи информации: какие данные будут передаваться, каком объеме (частота и размер транзакций, какой способ доставки сообщений между системами использовать (файлы, web-запросы и пр.), в какой момент будет происходить передача информации.

Иногда требуется учесть безопасность передаваемых данных. Например, ее можно обеспечить за счет использования защищенных протоколов (HTTPS), размещения всех данных внутри одного контура.

Начиная с версии 3.11.x в ELMA появилась функция "Гарантированная доставка сообщений", которая позволит избежать потери данных при передаче между системами.

1.4 Техническое задание

Техническое задание – это документ, содержащий основные технические требования к продукту и отвечающий на вопросы: что данная система должна делать, как работать, при каких условиях и как это должно быть реализовано.

Если на проекте не предусмотрено техническое задание (например, при работе по гибкой методологии внедрения), необходимо составить список функциональных требований и план реализации. Для удобства восприятия далее по тексту мы будем использовать единое сокращение – **ТЗ**.

ТЗ во многом опирается на концепцию решения, однако содержит в себе больше технических деталей, необходимых для реализации решения.

При составлении ТЗ необходимо учесть все цели, задачи, роли и сценарии использования системы. Однако здесь также должны существовать рамки разумного. Система класса BPM предназначена для организации движения фактов (информации) и гораздо меньше подходит для хранения и анализа фактов. При проектировании ИТ-архитектуры не стоит пытаться сделать из BPM-системы лучшую базу данных для хранения чего-либо. Система гораздо лучше подходит как инструмент регламентации взаимодействия вокруг какого-то объекта.

Нужно учитывать уровень зрелости компании, ее внутренних регламентов и процессов. Не пытайтесь заложить в решение все и сразу или проектировать слишком детально. Чаще всего стоит сделать в первом контуре более упрощенный вариант и запустить его в работу. Он будет закрывать основные потребности и сценарии использования системы. А уже после этого усложнить логику, добавляя функционал по мелочам, в особенности в тех компаниях, которые начинают осмыслять свои процессы только при внедрении системы.

Помните, что любые нестандартные решения и пользовательские доработки, как правило, генерируют сложности и дополнительные трудозатраты в реализации и дальнейшем сопровождении решения. Оцените реальную пользу от такого решения по отношению к рискам и сложностям.

Все договоренности (устные или письменные), которые были достигнуты в ходе обсуждения с заказчиком, должны быть зафиксированы в ТЗ. По всем спорным (неутвержденным) моментам необходимо получить от заказчика какое-то решение. Они будут или закреплены в ТЗ, или их последующее внесение будет считаться доработкой.

Глава 2. Моделирование бизнес-процессов

Бизнес-процесс – это совокупность взаимосвязанных мероприятий или задач, направленных на получение определенного результата.

Контур – это совокупность процессов и объектов, объединенных по смыслу, например, в рамках одного или нескольких сценариев использования.

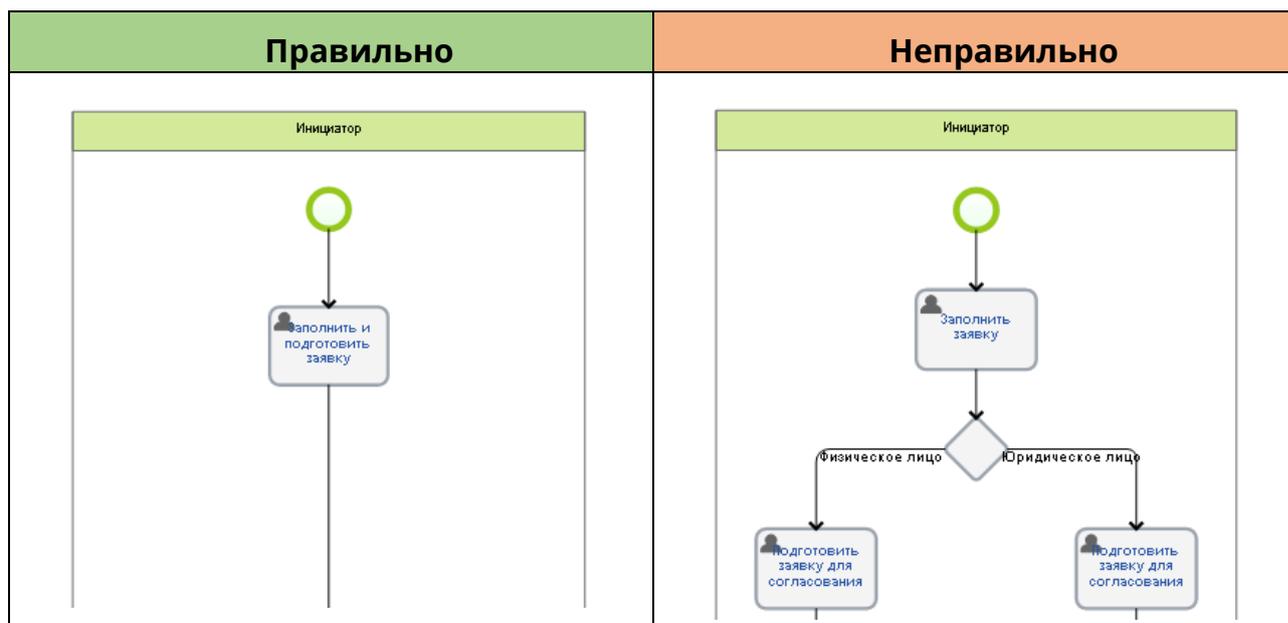
2.1 Моделирование решения

Модель бизнес-процесса должна адекватно отражать мероприятия или задачи предприятия. При моделировании бизнес-процесса старайтесь руководствоваться здравым смыслом. Решение должно быть достаточно простым и удобным для пользователя системы.

При разработке бизнес-процесса можно объединить несколько последовательных шагов в рамках одной роли в одну операцию. Это позволит сократить количество действий и переходов, которые необходимо совершить пользователю. Если уровень компьютерной грамотности конечного пользователя довольно низок, стоит разбить операцию на несколько простых шагов, сделав пошаговый мастер (даже в рамках одной бизнес-операции).

Не допускается размещение двух и более задач подряд (либо Оповещения и Задачи) в одной зоне ответственности. Излишняя детализация бизнес-процесса ошибочна (Таблица 2). При необходимости для объединения нескольких бизнес-операций можно использовать динамическую форму задач с заданными параметрами на форме (Рис. 7).

Таблица 2. Моделирование решения



The image shows two screenshots of the 'Заявка на кредит' (Credit Application) form in the ELMA system. The top screenshot shows the form for 'ООО "Практика"', with a dropdown menu for 'Условия кредитования' (Credit conditions) showing options: 'Кредит наличными' (Cash credit), 'Лизинг' (Leasing), and 'Рефинансирование' (Refinancing). The bottom screenshot shows the form for 'Иван Иванов Иванович', where the dropdown menu for 'Условия кредитования' shows options: 'Автокредит' (Auto loan), 'Ипотека' (Mortgage), 'Кредит наличными' (Cash credit), 'Кредитная карта' (Credit card), and 'Рефинансирование' (Refinancing). The bottom screenshot also displays additional fields: 'Процентная ставка (%)' (Interest rate) set to 0, and 'Ежемесячный платеж (руб)' (Monthly payment) set to 0.00, along with a 'Рассчитать условия' (Calculate conditions) button.

Рис. 7. Пример задачи с динамической формой

Метрики и показатели процесса, позволяют собирать параметры процесса. Значения показателя экземпляра процесса можно рассчитать из значений контекстных переменных или времени исполнения этапа, задачи или экземпляра процесса в целом. Показатель процесса вычисляется на основе показателей экземпляров этого процесса. В системе ELMA реализована возможность рассчитать показатель процесса как минимальное или максимальное значение,

сумму или среднее значение метрик и показателей экземпляров этого процесса (Рис. 8, Рис. 9).

The screenshot shows a software interface with a navigation bar at the top containing icons for 'Графическая модель', 'Контекст', 'Матрица ответственности', 'Метрики и показатели', 'Формы', and 'Сценарии'. Below the navigation bar, there are two tables. The first table is titled 'Показатели экземпляра процесса' and has columns for 'Название', 'Тип', and 'Правило вычисления'. It lists 'Сумма счета - метрика' (Technical type) and 'Время оплаты счета' (Time interval type). The second table is titled 'Показатели процесса' and has columns for 'Название', 'Тип', 'Правило вычисления', 'Период', and 'Связан с показателем(KPI)'. It lists 'Средняя сумма счета' and 'Среднее время оплаты счета', both with a calculation rule of 'По завершению бизнес-процесса' and a period of 'Месяц'.

Показатели экземпляра процесса		
Название	Тип	Правило вычисления
Сумма счета - метрика	Технический(Дробное число)	
Время оплаты счета	Временной(Временной интервал)	Время выполнения бизнес-процесса

Показатели процесса				
Название	Тип	Правило вычисления	Период	Связан с показателем(KPI)
Средняя сумма счета	Технический(Дробное число)	По завершению бизнес-процесса	Месяц	
Среднее время оплаты счета	Технический(Временной интервал)	По завершению бизнес-процесса	Месяц	

Рис. 8. Пример работы с разделом "Метрики и показатели процесса"

The screenshot shows a web application interface with a dropdown menu 'Метрики и показатели' expanded. Below it is a table with columns 'Показатель / Пользователь', 'План', 'Факт', and 'Статус'. Two rows are visible: 'Средняя сумма счета (Август 2014)' with a plan of 50 000,00 and a fact of 25 000,00, and 'Среднее время оплаты счета (Август 2014)' with a plan of 3 дн. and a fact of 3 мин. 19 сек.

Показатель / Пользователь	План	Факт	Статус
Средняя сумма счета (Август 2014)	50 000,00	25 000,00	🟡
Среднее время оплаты счета (Август 2014)	3 дн.	3 мин. 19 сек.	🟢

Рис. 9. Пример работы с разделом "Метрики и показатели процесса" в веб-приложении

Если данные используются за пределами конкретного экземпляра процесса, для удобного хранения и работы с информацией необходимо использовать объекты (Рис. 10). Не надо хранить все в контексте процесса.

The screenshot shows a web application interface for a 'Поставщики' (Suppliers) reference page. It includes a search bar with the text 'Что искать?' and a search button. Below the search bar, there is a table with columns 'Наименование', 'Адрес', and 'Регион'. The table lists four suppliers: 'АиА' (Moscow), 'ООО "Пул"' (Yekaterinburg), 'ЗАО "Мани"' (Orel), and 'ООО "Билд"' (Kursk). The interface also shows navigation buttons like 'Назад' and 'Создать', and user information for 'Саратов Н. С.'.

Наименование	Адрес	Регион
АиА	г. Москва	Центральный
ООО "Пул"	г. Екатеринбург	Уральский
ЗАО "Мани"	г. Орел	Центральный
ООО "Билд"	г. Курск	Центральный

Рис. 10. Страница справочника "Поставщики"

Используйте перечисления в тех случаях, когда в разных разделах системы вы используете одни и те же выпадающие списки – например, перечисление статусов заявки. Обратите внимание, что, в отличие от справочников, набор значений для перечисления нельзя изменить в веб-приложении (только через Дизайнер ELMA). Перечисления в веб-приложении могут отображаться в виде выпадающего списка, горизонтальной/вертикальной группы переключателей или одного переключателя (Рис. 11).

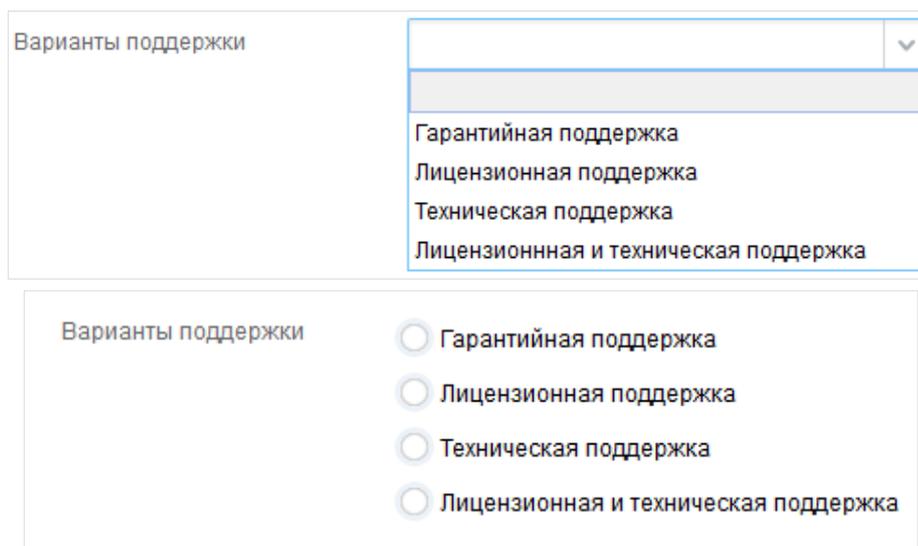


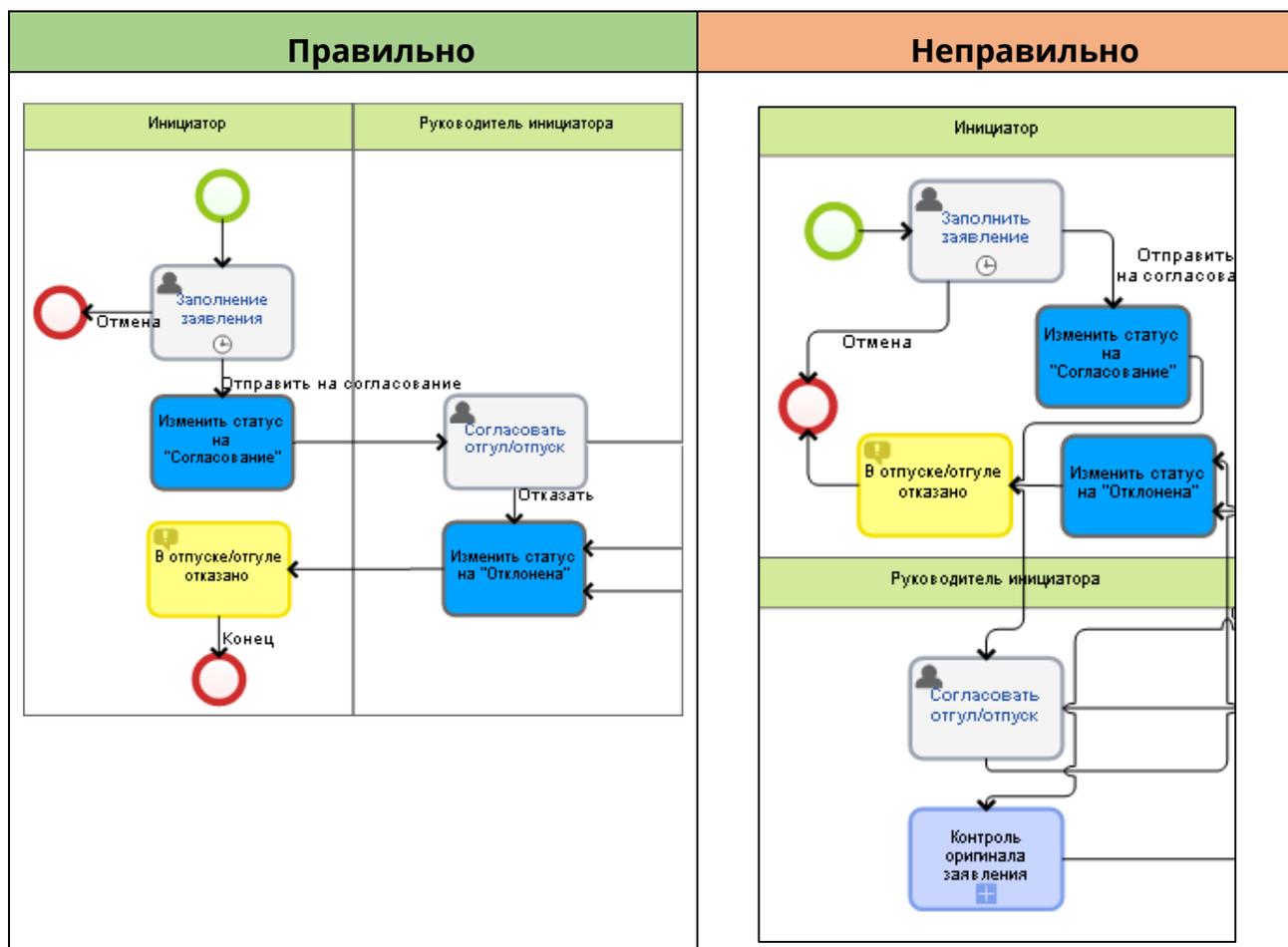
Рис. 11. Примеры работы с перечислениями на форме задачи

2.2 Графическая модель процесса

Графическая модель не только влияет техническую роль, но и является визуальной схемой (картой) процесса. Поэтому она должна наглядно изображать весь процесс, быть легко читаемой.

Зоны ответственности на графической модели должны располагаться вертикально (слева направо). Называть их следует в соответствии с ролью ответственного в процессе (Таблица 3).

Таблица 3. Графическая модель процесса

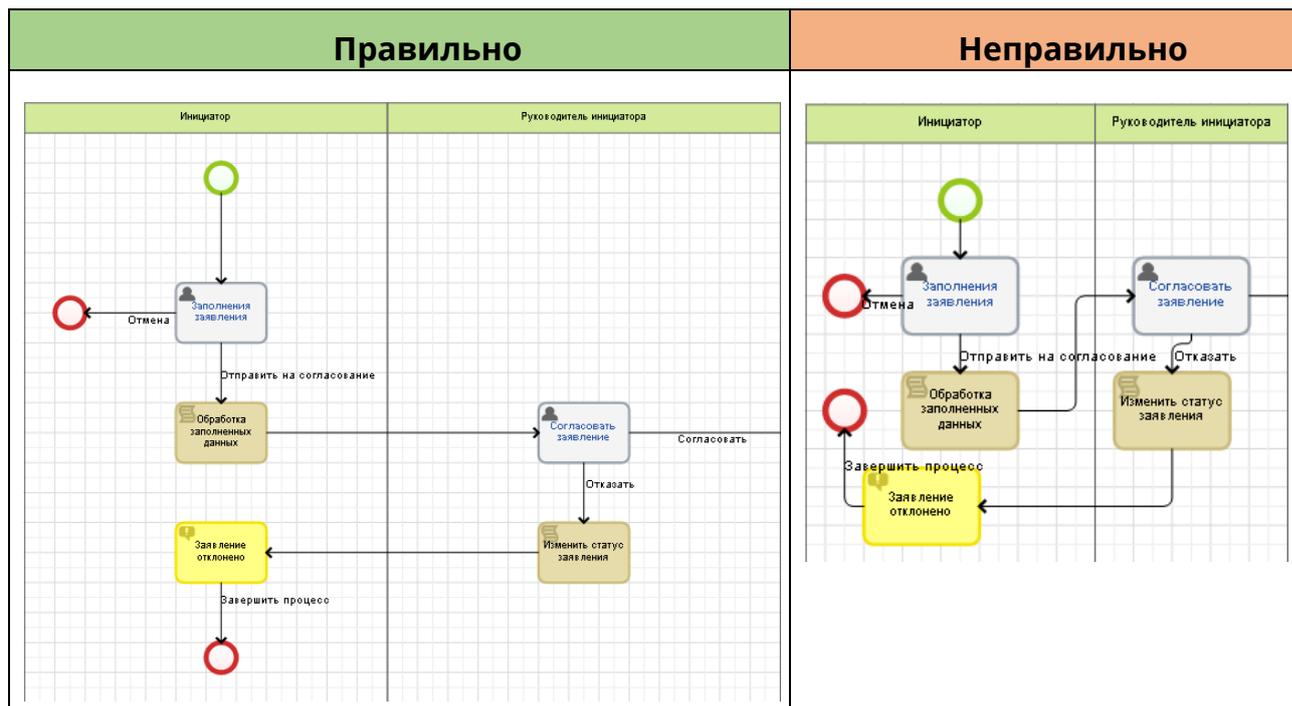


Расположение блоков на схеме должно соответствовать последовательности исполняемых шагов. Блоки должны быть одинакового размера, а их расположение необходимо выравнять относительно друг друга по горизонтали и вертикали, используя координатную сетку.

Старайтесь располагать блоки так, чтобы переходы не накладывались друг на друга и не переплетались, а их названия можно было легко прочесть (Таблица 4).

При составлении графической модели оставляйте место для дальнейшего расширения процесса.

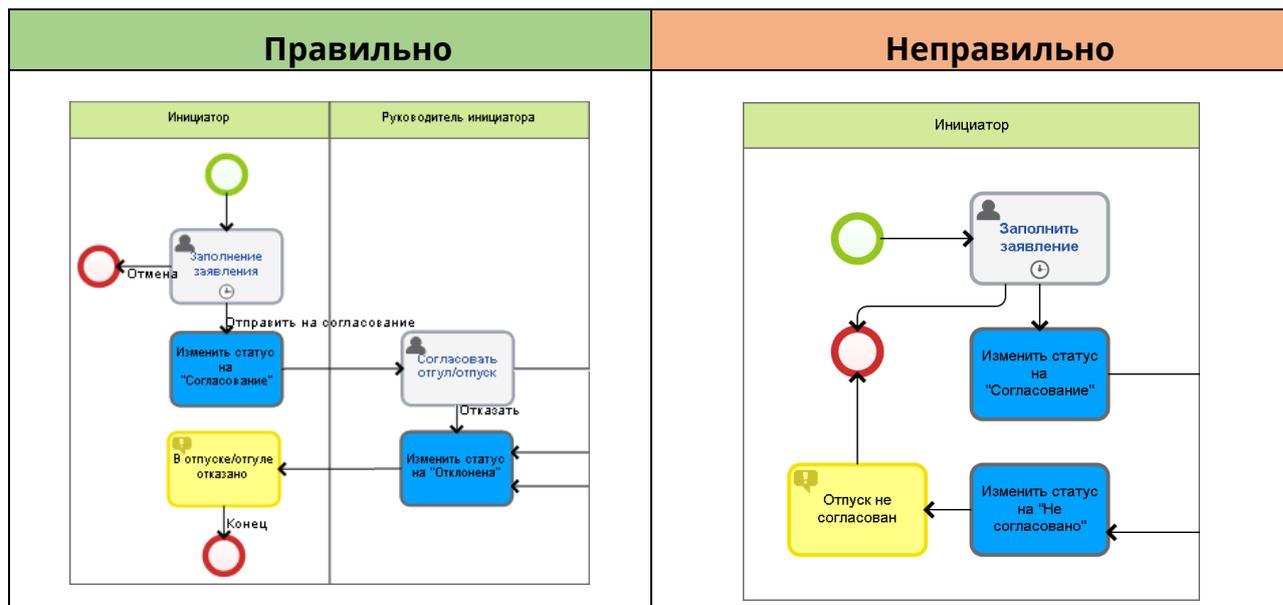
Таблица 4. Переходы на графической схеме бизнес-процесса



Все события, шлюзы и переходы должны быть подписаны и иметь уникальные лаконичные наименования. Не допускается использование технических и автоматических названий, таких как "Задача 1", "Событие 2", "Шлюз 3" и т.д.

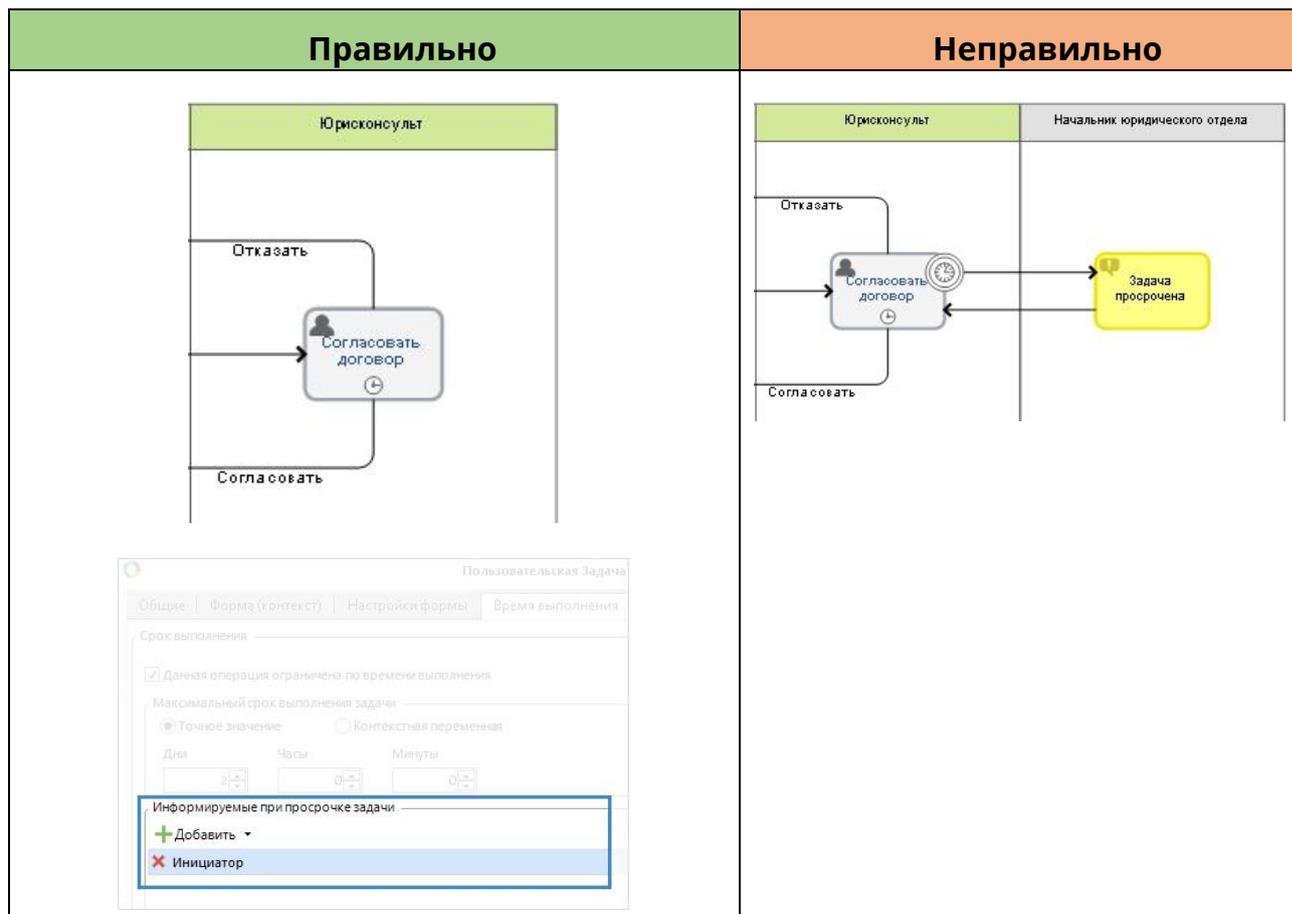
Особое внимание обратите на переходы в конечные события. Каждая точка выхода из процесса должна быть снабжена своим элементом завершающего события. Если в процессе используются статусы, убедитесь, что перед завершением процесса добавлено соответствующее изменение статуса (Таблица 5).

Таблица 5. Графическая модель бизнес-процесса с операцией "Изменение статуса"



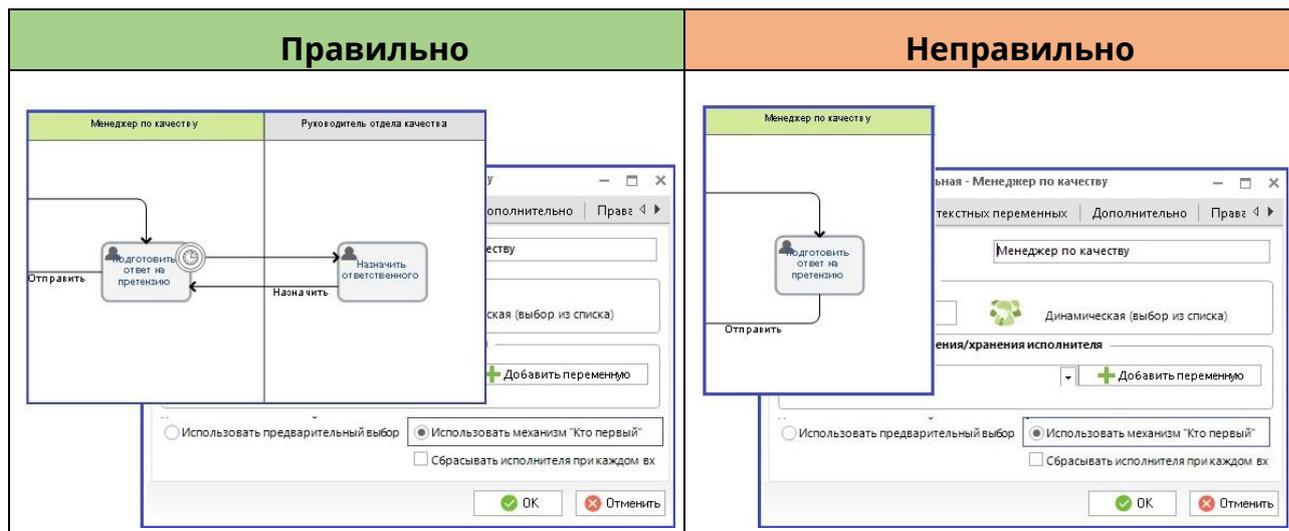
При настройке уведомления об истечении времени выполнения задачи не должен использоваться переход из задачи и обратно. В этом случае задача будет завершена и назначена повторно. Для оповещения пользователей можно использовать настройки **Срок выполнения** в задаче (Таблица 6).

Таблица 6. Графическая модель бизнес-процесса. Уведомление о просрочке задачи



При использовании механизма **Кто первый** в динамической зоне ответственности рекомендуется предусмотреть эскалацию, которая будет предполагать определение конкретного ответственного за задачу, иначе использование такого механизма может стать узким местом в бизнес-процессе.

Таблица 7. Графическая модель бизнес-процесса. Эскалация



Если модель получается слишком большой, то возможно часть процесса стоит перенести в подпроцессы. Если в разных процессах часть модели повторяется, то ее можно вынести во внешний подпроцесс.

2.3 Запуск бизнес-процессов

Стартовое событие – это начало любого процесса. По умолчанию пользователь, в зоне ответственности которого расположено стартовое событие, назначается ответственным за экземпляр процесса. Настройки зоны ответственности, содержащей стартовое событие, определяют список должностей, групп сотрудников и групп пользователей, участники которых могут выполнить запуск данного процесса (Рис. 12).

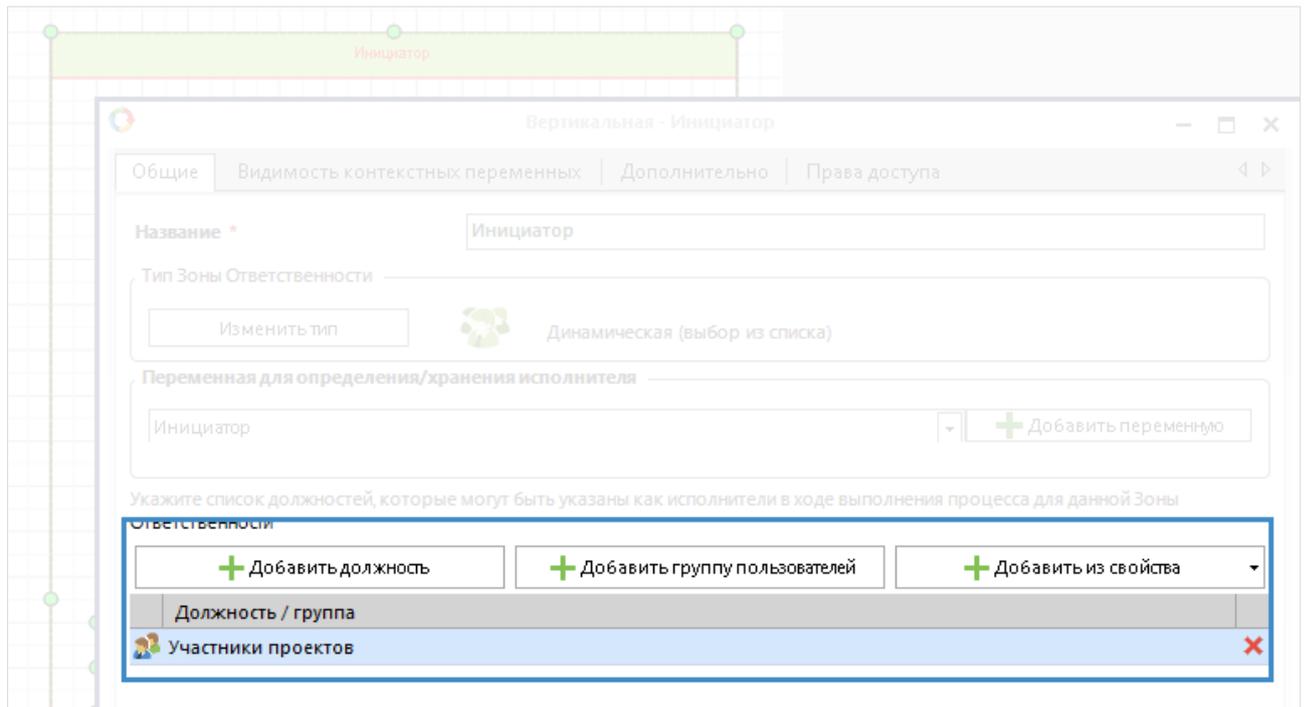


Рис. 12. Настройка списка пользователей для зоны ответственности

Всегда указывайте шаблон для автоматического наименования экземпляра процесса, чтобы пользователю не приходилось каждый раз набирать название вручную (Рис. 13).

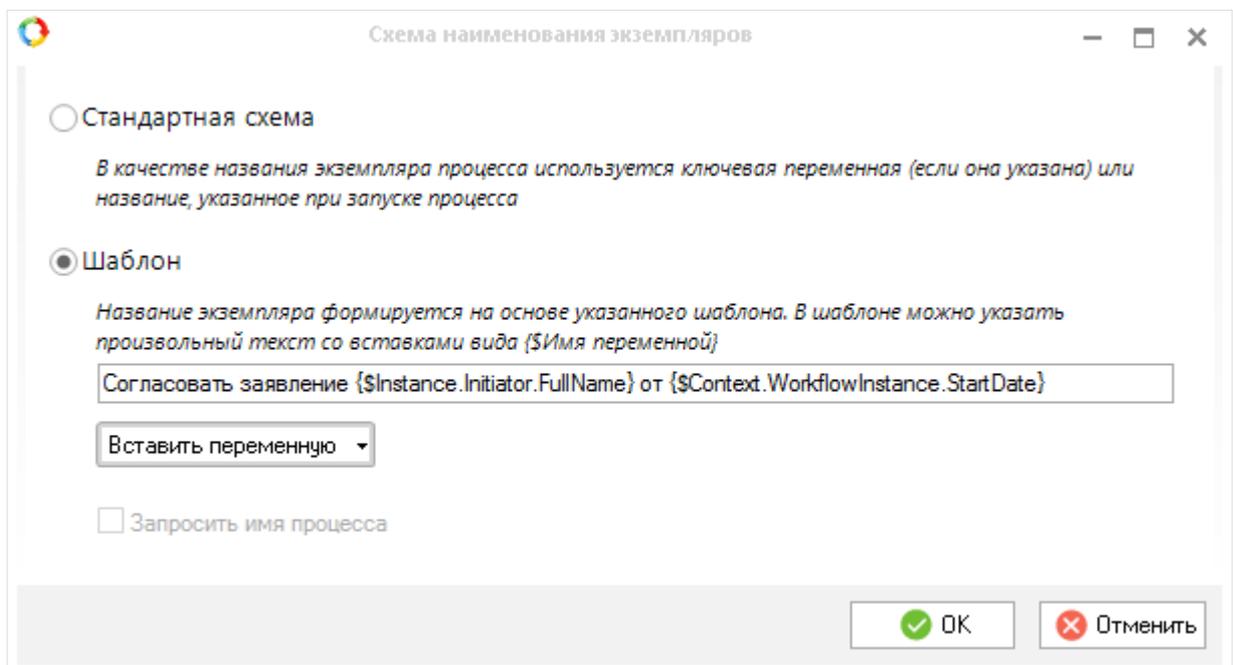


Рис. 13. Настройка формирования названия экземпляра по шаблону

Название экземпляра процесса должно быть понятным и лаконичным, должно отражать назначение процесса, а также желательно должно содержать информацию о ключевых объектах процесса (например, контрагенте или названии документа, который участвует в этом экземпляре процесса).

Старайтесь избегать объемных сценариев в начале процесса. Если они все же необходимы, то необходимо предусмотреть уведомление пользователя о том, что процесс запущен. Это необходимо для того, чтобы пользователь видел результат запуска процесса, понимал, что в данный момент происходит с процессом, и что ему дальше необходимо делать – закрыть вкладку или дождаться, пока откроется какая-то форма задачи.

2.4 Настройка стартового события

Бизнес-процесс может быть запущен несколькими способами. Обычно используется ручной запуск с помощью кнопки **Запустить процесс** на верхней панели в веб-приложении. Но существуют и другие способы. Настройки условий запуска необходимо осуществлять в зависимости от назначения процесса, удобства пользователя и уровня его компьютерной грамотности.

Ниже рассмотрим дополнительные настройки для запуска процесса.

Запуск из внешнего процесса. С помощью блоков **Внешний подпроцесс** и **Внутренний подпроцесс** можно настроить запуск процесса из внешнего процесса. Этот способ используется, когда необходимо разделить внешний процесс на части или часть операций бизнес-процесса повторяется в нескольких процессах.

Портлет "Запуск процесса". Портлет отображает все процессы, доступные пользователю для запуска. Использовать такой портлет стоит только в том случае, если список доступных процессов небольшой и все они используются одинаково часто.

Запуск по кнопке. С помощью HTML-кода можно добавить кнопку или ссылку для запуска процесса. Такую кнопку можно добавить в портлет HTML, веб-документ или через создание пользовательской формы веб-приложения. Этот способ подходит в тех случаях, когда пользователь часто использует один или несколько процессов (например, создание заявки оператором).

Запуск по таймеру. В стартовом событии процесса можно включить условие запуска по таймеру. В этом случае процесс будет запущен в указанные дату и время. Такой способ подходит, когда процесс должен запускаться автоматически

с некоторой периодичностью (например, для формирования ежеквартального отчета). Обратите внимание, что запуск процесса по таймеру возможен, только если в зоне ответственности выбран конкретный пользователь или должность (единичная). Для автоматических процессов стоит предусмотреть оповещение ответственных о возможных ошибках при исполнении процесса.

Запуск по событию. В этом случае процесс запускается после получения сообщения из очереди сообщений. Данный способ используется в тех случаях, когда процесс должен запускаться автоматически при определенных событиях. Как и в случае с использованием таймера, данное событие должно находиться в статической зоне ответственности.

Запуск из внешних систем. Для запуска процесса из внешних систем используется **SOA Connector**. Взаимодействие между системой ELMA и внешним приложением происходит в виде обмена сообщениями через HTTP-запросы (поддерживаются HTTP-GET и HTTP-POST запросы) или же через веб-сервис ELMA (используется протокол SOAP). Это позволяет создать ссылку, которая удаленно запускает процесс в ELMA из внешних приложений. Например, это можно использовать для постановки задач из почтового клиента.

Для служебных процессов и подпроцессов, для которых не предусмотрен самостоятельный запуск пользователем, желательно отключить возможность запуска из веб-приложения.

В настройках стартового события можно дополнительно вынести контекстные переменные, которые можно будет заполнить еще до запуска самого процесса (Рис. 14). Не стоит выносить на стартовое событие большой список переменных – используйте этот инструмент, например, для маршрутизации процесса (сразу после его запуска).

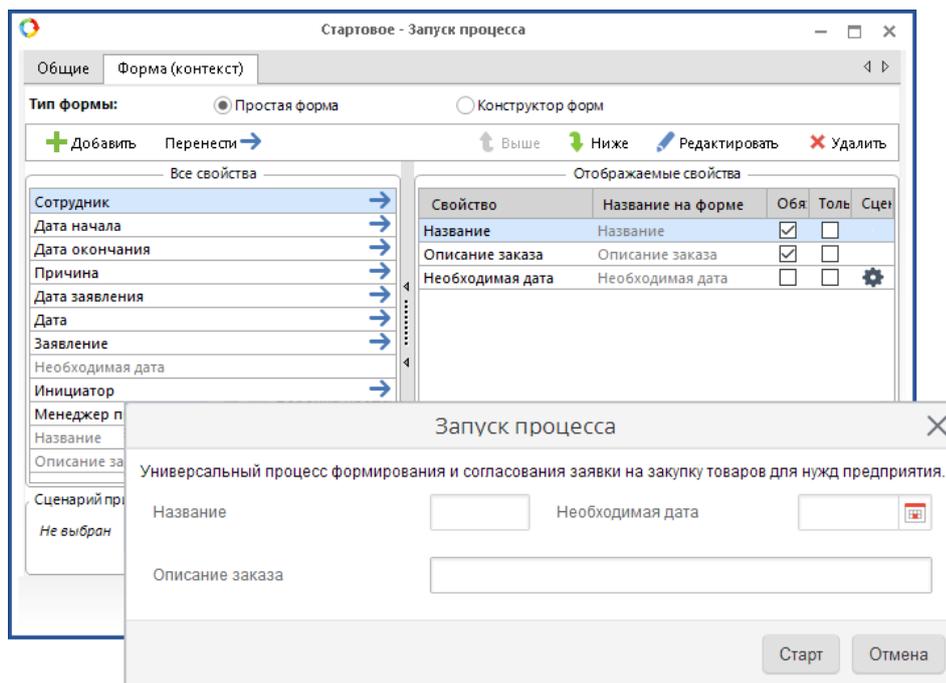


Рис. 14. Настройка формы стартового события

2.5 Настройка пользовательских задач

Название задачи и ее описание должно быть понятным и лаконичным. Они помогают пользователю понять, что за задача перед ним стоит, к какому процессу она относится. Используя ключевые объекты из контекста процесса, можно формировать название задачи по шаблону (Рис. 15). Это упростит поиск и обработку задачи.

Пользовательская Задача - Регистрация заявки

Общие | Форма (контекст) | Настройки формы | Время выполнения | Дополнительно | Регламент

Название * Регистрация заявки

Название задается по шаблону

Регистрация заявки {\$Context.Iniciator.FullName} от {\$Instance.StartDate}

Название задачи формируется на основе указанного шаблона. В шаблоне можно указать произвольный текст со вставками вида {\$Имя переменной}

Вставить переменную ▾

Описание

Описание задается по шаблону

Тип Операции

Изменить тип Операции Пользовательская Задача

Маркер

Изменить маркер Отсутствует

Доступность во внешних устройствах и приложениях

Задача доступна во внешних устройствах и приложениях

Нельзя переназначать

OK Отменить

Рис. 15. Настройка шаблона названия задачи

Форма задачи должна иметь удобный и понятный интерфейс. Полезные комментарии (описание задачи, переменных или текстовые вставки) могут помочь сориентироваться в информации.

Но старайтесь не перегружать форму информацией – важные элементы оставлять на виду, дополнительную информацию можно скрыть в свернутые панели. Для сложных задач, требующих пояснений, вместо множества комментариев, можно прикрепить справку-инструкцию с подробным описанием.

Переменные должны располагаться на форме в прямой последовательности, так чтобы их заполнение происходило справа налево и сверху вниз. Необходимо учитывать возможности мониторов пользователей. В рамках одной системы старайтесь соблюдать схожий порядок переменных, даже для разных процессов.

Обязательные для заполнения поля должны иметь соответствующий флаг. И наоборот, необязательные переменные должны оставаться необязательными для заполнения, чтобы не перегружать форму. С помощью динамики на форме можно изменять обязательность заполнения поля, в зависимости от каких-либо условий. Поля, используемые только для информирования и ознакомления в рамках задачи должны быть доступны только для чтения (Рис. 16).

Оформление заявки

> Информация о процессе

Главная страница | История

Ответственный: Иванов И. И.

Документ * + 🔍 ▾

Контрагент * + 🔍 ▾

▼ Дополнительная информация

Контактное лицо (ФИО)

Телефон:

Описание

Рис. 16. Пример формы пользовательской задачи

Для лучшего восприятия можно сгруппировать информацию с помощью конструктора форм. Для создания более сложных форм можно использовать Razor-формы. Однако их создание и настройки более трудоемкие.

За счет сценариев можно настроить динамическое скрытие/отображение переменных на форме (Рис. 17). Это позволит разгрузить форму для пользователей, сделать ее более удобной. Используйте сценарии на изменение переменной, только когда это влияет на заполнение/отображение полей на форме. В ином случае выносите эти сценарии в отдельный шаг процесса, так как время на исполнение сценария сказывается на быстродействии системы.

В сценариях на загрузке формы нельзя создавать/удалять/изменять объекты. Это приведет к тому, что при каждом обновлении страницы/изменении переменной будет создаваться новый объект.

Рис. 17. Пример динамической формы задачи

Дочерние формы позволяют упростить работу по созданию и настройке в тех случаях, когда содержание форм частично или полностью дублируется в рамках одного процесса (Рис. 18). Чтобы не дублировать свойства из объектов в контекст процесса, можно выносить на форму его вложенные свойства.

Обратите внимание, что при использовании вложенных свойств сам объект становится недоступен для редактирования на форме. Если одна контекстная переменная в разных задачах несет разную информационную нагрузку – в конструкторе можно изменить ее отображаемое имя и описание прямо на форме.

Рис. 18. Пример использования дочерней формы задачи

Использование блоков на форме сильно увеличивает информационную нагрузку – трижды подумайте, как пользователь будет с ними работать. Крайне не рекомендуется использовать вложенность блока в блоке.

При работе с блоками помните, что они имеют ряд ограничений и особенностей, связанных с использованием сценариев.

Регулярное выражение позволяет настроить шаблон, по которому будет проверяться заполнение строковой переменной. В случае несоответствия в веб-приложении может быть выведено сообщение об ошибке, указанное в соответствующем поле (Рис. 19).

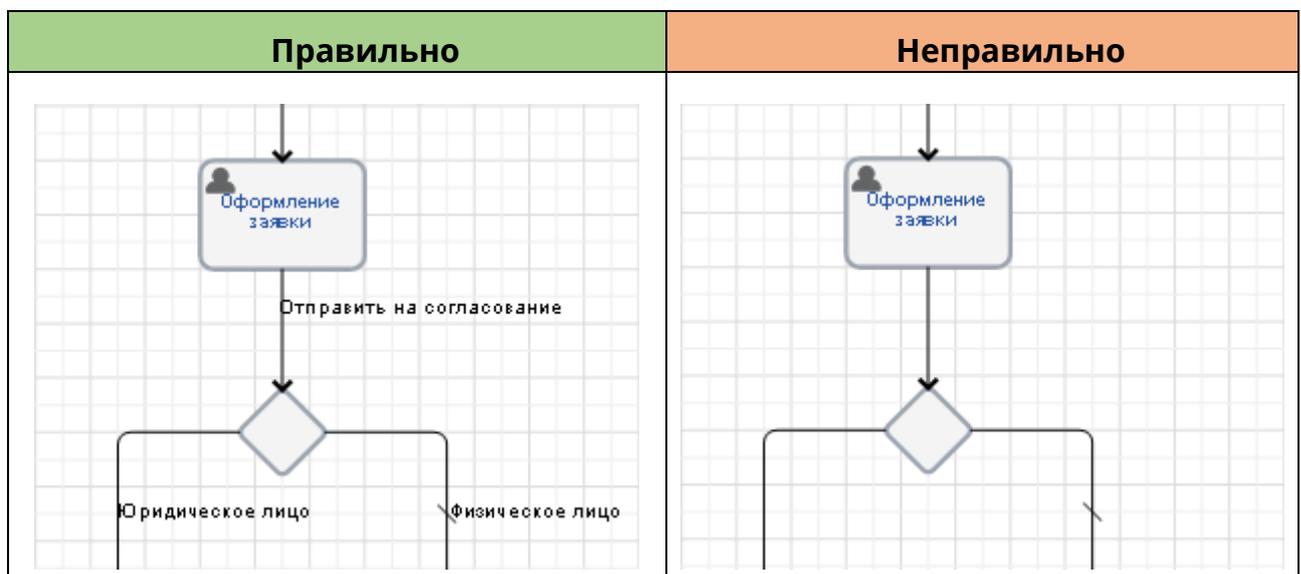
Рис. 19. Валидация значения поля на форме задачи

В настройках пользовательской задачи можно **установить Сообщение после выполнения**, которое будет приходить пользователю после завершения задачи.

2.6 Настройка переходов

Переходы изображают маршрут движения бизнес-процесса. Старайтесь сопровождать каждый переход лаконичным названием (и при необходимости комментарием), чтобы улучшить читаемость карты процесса (Таблица 8). Переходы из задач всегда должны иметь название (краткое и понятное).

Таблица 8. Графическая схема бизнес-процесса. Настройки переходов



В веб-приложении переходы из блока задачи отображаются в виде кнопок. Старайтесь соблюдать одинаковую последовательность и название кнопок во всех процессах системы (особенно в рамках одного процесса).

Принято располагать переходы в следующем порядке: первым ставится основной переход, последним – завершение или возврат по маршруту (например, отказ в согласовании), между ними в порядке приоритета и частоты использования ставятся дополнительные переходы.

Цветовое отображение перехода позволит повысить удобство использования. Старайтесь соблюдать ту же логику, что и с последовательностью: основной переход – зеленый; завершение процесса – красный; дополнительные переходы – синие, либо желтые (Рис. 20).

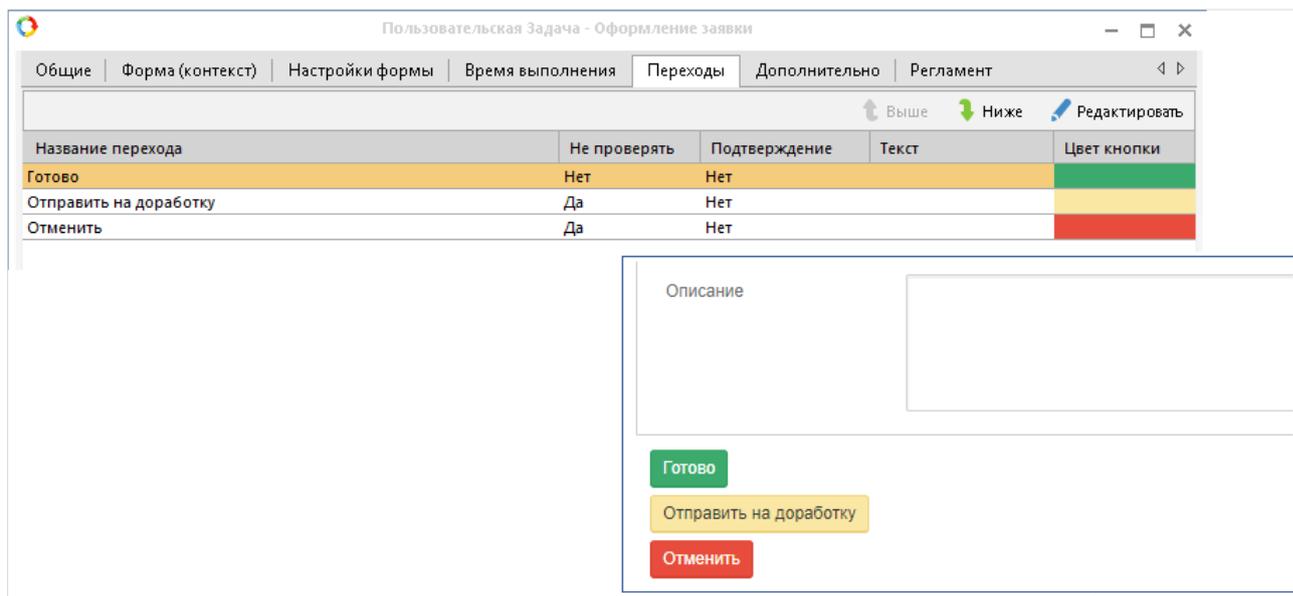


Рис. 20. Пример отображения переходов в процессной задаче

В некоторых задачах процесса должен быть предусмотрен переход с прерыванием процесса. На таких переходах можно добавить настройку **Не проверять обязательность заполнения контекстных переменных**, чтобы пользователю не приходилось заполнять обязательные поля для прерывания процесса.

Для важных переходов процесса (например, для прерывания) можно предусмотреть подтверждение (Рис. 21). Не стоит злоупотреблять подтверждениями, или нагружать их объемными комментариями, так как это неудобно для пользователей и будет вызывать лишь раздражение.

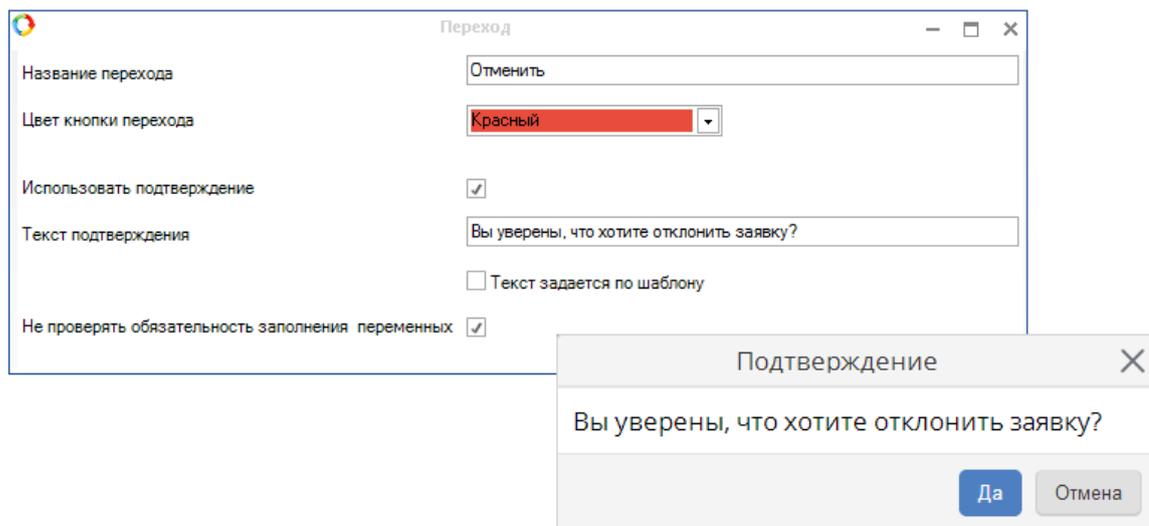


Рис. 21. Окно подтверждения перехода

2.7 Сценарии

Прежде чем приступать к написанию сценария, всегда хорошо продумайте логику. Старайтесь избегать громоздких функций. Дублирования кода можно избежать с помощью разделения большого кода на более мелкие методы. Если код используется в нескольких процессах/объектах – лучше вынести эту часть кода в глобальный метод или создать пользовательское расширение.

Работая в команде, старайтесь использовать общий стиль написания кода. Например, для написания публичных членов, типов и пространств имен, состоящих из нескольких слов, подойдет стиль Pascal, где первая буква идентификатора и первая буква каждого последующего присоединенного слова являются прописными. Для именования временных переменных используйте стиль lowerCamelCase, где первая буква идентификатора является строчной, а первая буква каждого последующего слова – прописной.

Для повышения удобства работы выравнивайте код, сопровождая каждый элемент поясняющими комментариями (Рис. 22). Наименования в сценариях и переменных (даже временных) должны иметь понятное и корректное английское название.

```
1 //определим тип сущности на основе типа документа.
2 //Вместо "DocType" необходимо ввести имя класса Вашего документа. Обратите внимание что в конструкции используется его интерфейс
3 var docType = new ReferenceOnEntityType { TypeUid = InterfaceActivator.UID<IDokType>(false) };
4 //найдем вариант регистрации
5 var reg_variant = RegistrationVariantManager.Instance
6     .Find(new InstanceOf<IRegistrationVariantFilter>(){New =
7     {
8         //укажем тип документопотока "Входящие"
9         Docflow = DocumentFlowType.Incoming,
10        //зададим тип документа
11        DocumentType = docType,
12        //укажем UID для DepositoryGroup. Как определить его значение описано ниже
13        DepositoryGroupUid = new Guid("64AEC71A-D9F3-488E-8E4F-1AD47E31D373"),
14        //зададим имя варианта регистрации
15        Query = "Name = 'Вариант1'"
16    }}.New, new FetchOptions(0, 1))
17     .FirstOrDefault();
18 //если вариант регистрации не найден - выход из сценария
19 if (reg_variant == null) {
20     Console.WriteLine("Variant not found");
21     return;
22 }
```

Рис. 22. Пример использования комментариев для кода

В некоторых случаях можно предусмотреть эскалацию для обработки ошибки в блоке сценария (Рис. 23). Это особенно полезно в автоматических процессах, где пользователь не всегда контролирует исполнение процесса на протяжении всего маршрута.

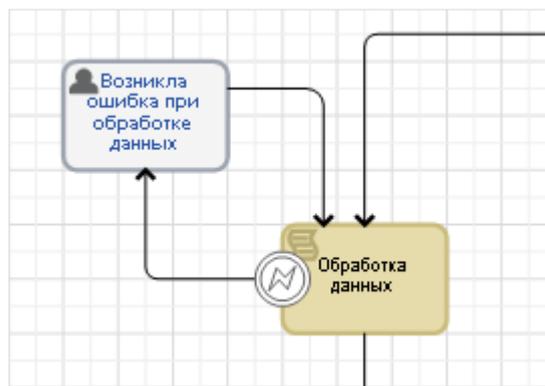


Рис. 23. Пример использования эскалации для обработки ошибки

В сложных сценариях можно предусмотреть обработку ошибок через try-catch (там, где это не навредит логике процесса). Логирование – это полезный инструмент для обработки ошибок. Используйте логирование там, где могут возникнуть сложности. Вся информация о событиях и ошибках пишется в лог-файлы. Не стоит злоупотреблять этим инструментом, чтобы не перегружать лишней информацией лог-файлы.

Старайтесь использовать PublicAPI, вместо обычных функций. Это сильно упростит чтение и поддержку сценария в дальнейшем. В этом вам поможет справка: https://www.elma-bpm.ru/kb/help/elma_api_help/.

По умолчанию в системе настроен таймаут исполнения сценария – 10 минут. Если в сценарии предусмотрена обработка большого количества записей, желательно разбить его на циклы (Рис. 24).

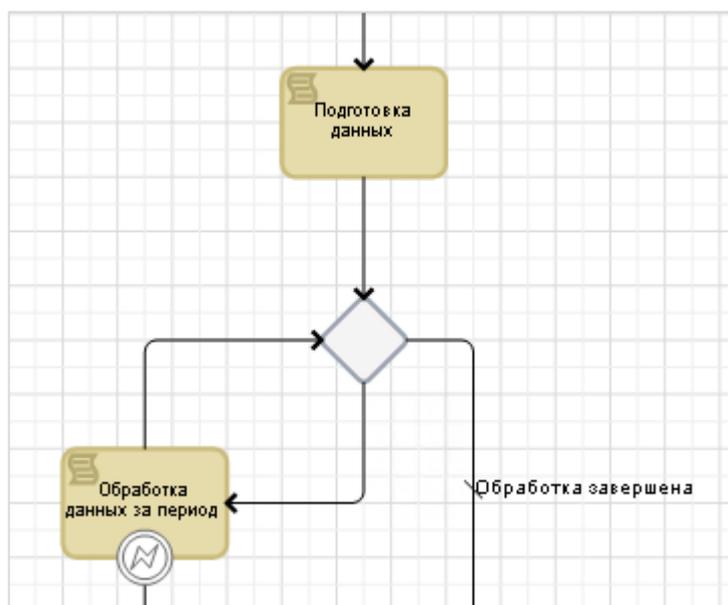


Рис. 24. Пример использования цикла для обработки большого объема данных

Для отладки и тестирования сценария используйте **Эмуляцию** (Рис. 25). Функция эмуляции позволяет проверить работу выбранного сценария при заданных параметрах контекста. При необходимости можно использовать эмуляцию для внесения изменений в данные системы с помощью сценария. По умолчанию в эмуляции включен режим **Окат после исполнения** – все изменения не будут сохранены после исполнения сценария. Обратите внимание, что в режиме эмуляции нельзя вести работу с формами процесса.

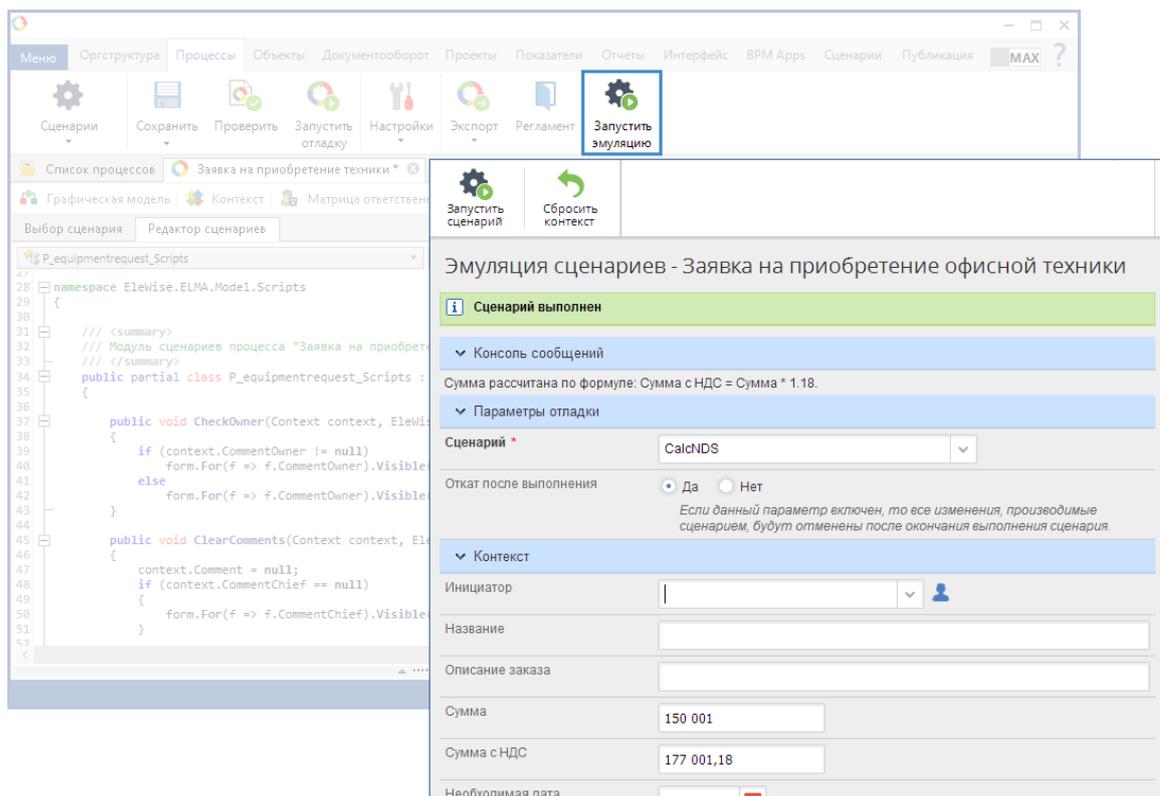


Рис. 25. Консоль сообщений при эмуляции сценария

Изменения сценария в объектах применяются только после перезагрузки системы. Предварительная проверка сценария в режиме эмуляции (например, в тестовом процессе) позволит отладить сценарий без лишних перезагрузок.

Старайтесь не использовать в сценарии статические переменные, обращаясь к ним через Id или Uid, так как это ухудшает читаемость и отказоустойчивость кода. Если все же без них не обойтись – объявляйте такую переменную, как константу в начале метода, чтобы позже ее легко можно было заменить.

Полный список объектов системы (в том числе скрытых) и их свойств можно посмотреть по адресу **<http://<адрес сервера>/Api/Help>**.

Одна из самых частых ошибок при работе со сценариями – это попытка обращения к пустому значению (NullReferenceException). Чтобы избежать ее,

каждый раз после получения объекта в сценарии (например методом Find()) или при обращении к вложенным свойствам, необходимо добавлять проверку на NULL.

Старайтесь не использовать метод FindAll() и многоуровневые переборы (например, foreach в foreach или while в while и т.д.). Это плохо сказывается на производительности системы. Чаще всего их можно заменить за счет создания фильтра.

2.8 Администрирование процессов

Для каждого процесса нужно настроить права в соответствии с ролевой моделью. Группе "Администраторы ELMA" необходимо предоставлять полные права на все процессы системы. Это необходимо для комфортной настройки и отладки процессов, а также их дальнейшего администрирования (Рис. 26).

Исполнитель	Владелец	Участник	Информируется	Куратор
Финансовый директор (Финансовый директор)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Системный администратор (Администратор)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Технический директор (Технический директор)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ИТ-директор (ИТ-директор)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Все пользователи (Инициатор)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Рис. 26. Матрица ответственности процесса

Проверьте корректность настроек в Зонах ответственности – убедитесь, что пользователи выбраны корректно, им назначены соответствующие должности, что права в **Матрице ответственности** выставлены верно, что для каждой Зоны ответственности настроен список контекстных переменных, доступных для просмотра, что переменные привязаны к Зонам ответственности там, где это требуется, а лишние или технические переменные скрыты для отображения. Убедитесь, что исполнители задач имеют права на вложенные объекты. Уровень прав должен соответствовать роли, исполняемой, в рамках задачи. Заранее предусмотрите, нужно ли забирать эти права после завершения задачи.

Перед публикацией процесса, убедитесь в его корректной работе. Для этого можно воспользоваться механизмом **Отладка процесса** (Рис. 27).

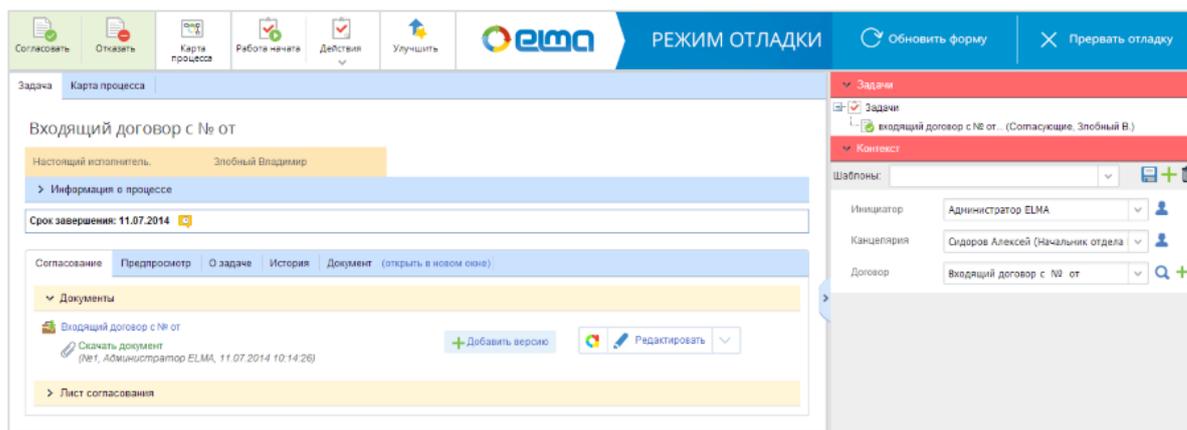


Рис. 27. Форма отладки процесса

Проверяйте различные сценарии использования процесса. Не должно быть ситуации, при которой процесс прерывается с ошибкой. Например, когда какие-либо данные не заполнены или указаны с ошибкой.

По завершении разработки процесса необходимо обязательно определить видимость, права на видимость и поиск по списку контекстных переменных для процесса (Рис. 28). Все служебные переменные должны быть исключены из пользовательского отображения (Рис. 29).

Отображаемое имя	Имя свойства	Тип	Поиск
Базовые свойства			
Инициатор	Initiator	Пользователь (Объект)	<input checked="" type="checkbox"/>
Статус экземпляра процесса	WorkflowInstanceStatus	Выпадающий список	<input checked="" type="checkbox"/>
Номер договора	ContractNumber	Строка	<input checked="" type="checkbox"/>
Дата договора	ContractDate	Дата / время	<input checked="" type="checkbox"/>
Лист согласования	ApprovalList	Лист согласования (Объект)	<input type="checkbox"/>
Ошибка	Error	Ошибка выполнения сценария	<input type="checkbox"/>
Итератор	Iterator	Целое число	<input type="checkbox"/>

Рис. 28. Настройка участия контекстных переменных в поиске

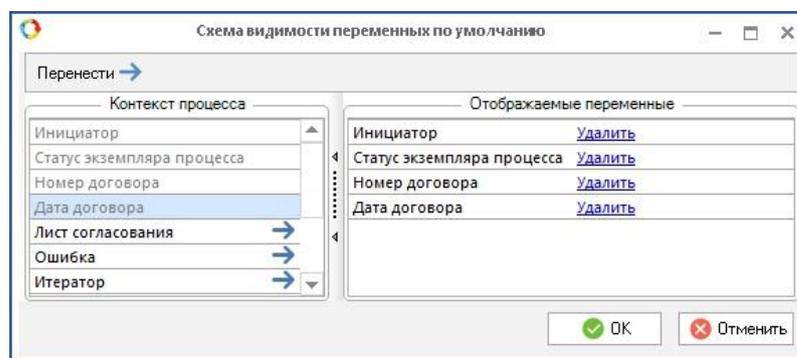


Рис. 29. Настройка видимости переменных процесса

Если в системе существует большое количество пользовательских объектов (справочников, документов, процессов, типов документов и пр.), для удобства работы необходимо группировать их по папкам (Рис. 30). Название папок должно быть понятным и лаконичным, без использования цифр и технических обозначений. Однако помните, что у пользователя должен быть простой доступ к часто используемым объектам. Чем чаще используется, тем ближе должно быть расположение (без необходимости дополнительно совершать лишние шаги).

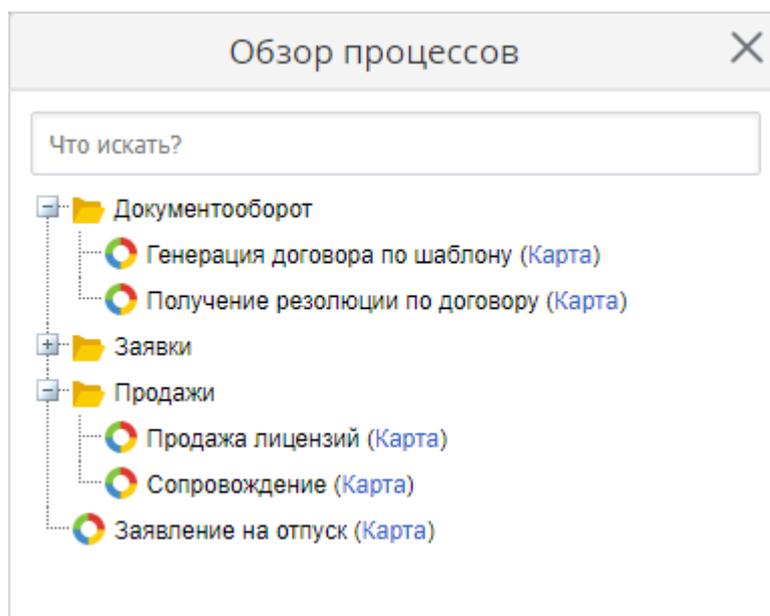


Рис. 30. Окно выбора процесса

Неиспользуемые/ненужные процессы системы лучше всего поместить в архивную папку и скрыть их из веб-приложения (отключив возможность ручного запуска). Удалять их нежелательно, т. к. в дальнейшем при создании или импорте процесса с аналогичным названием, могут возникнуть сложности. То же самое касается и работы с объектами.

В случае если работа ведется одновременно в нескольких системах (например, есть среда разработки и основная рабочая среда), создание и редактирование всех элементов системы необходимо производить на одной среде, а на остальные их переносить только с помощью импорта. В ином случае может возникнуть конфликт или дублирование объектов.

Глава 3. Контроль и мониторинг

Система ELMA имеет широкий спектр инструментов, позволяющих осуществлять контроль бизнес-процессов. Все функции можно разделить на две категории: оперативный и заключительный контроль.

Для осуществления контроля, в первую очередь, необходимо понять, в рамках какой роли будет осуществляться контроль процессов? В какой момент будет производиться контроль, и какие данные для этого необходимы? За какой период времени необходимо предоставить информацию для контроля?

Необходимо четко сформулировать роли бизнес-процесса и матрицу ответственности. Информация для каждого участника бизнес-процесса должна предоставляться в соответствии с его ролью. Обязательно убедитесь, что все пользователи имеют доступ и необходимый набор инструментов для осуществления контроля в рамках своей роли.

Исполнитель – пользователь, осуществляющий работу с процессом в рамках поставленных задач. В рамках этой роли, пользователю важно своевременно выполнить задачу. Поэтому для него будут полезны такие инструменты, как список задач (отсортированные по приоритету или сроку исполнения), уведомления о собственных просроченных задачах, сводная статистика по собственной эффективности (Рис. 31).

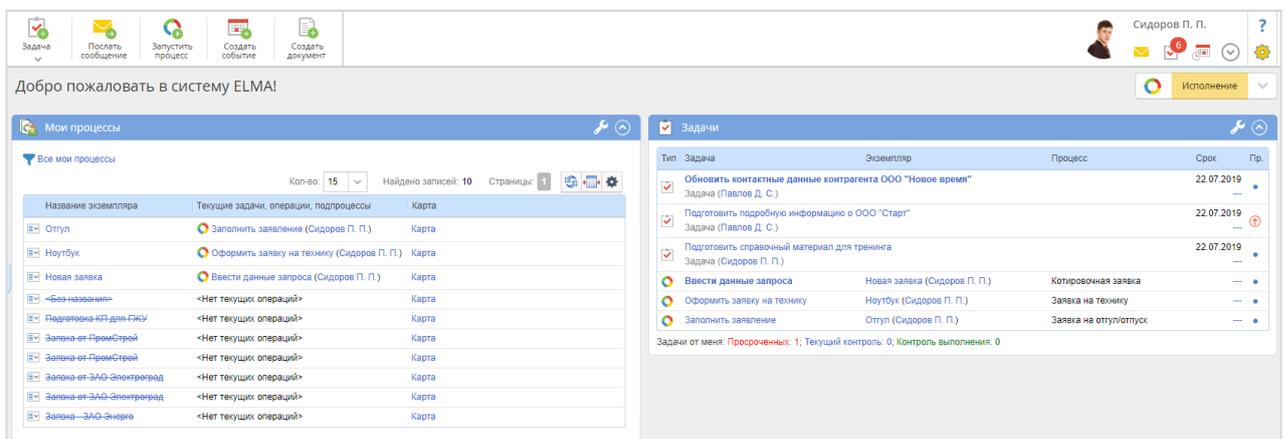


Рис. 31. Пример интерфейса для пользователя в роли Исполнителя

Ответственный – пользователь, отвечающий за исполнение процесса и его результат (Рис. 32). Здесь важно понимать уровень ответственности сотрудника. Условно можно разделить его на **Владельца** и **Куратора**. Куратор осуществляет текущий контроль за исполнением процесса и его результатами. Для Владельца

важно получить сводную информацию о результатах работы в рамках бизнес-процесса.

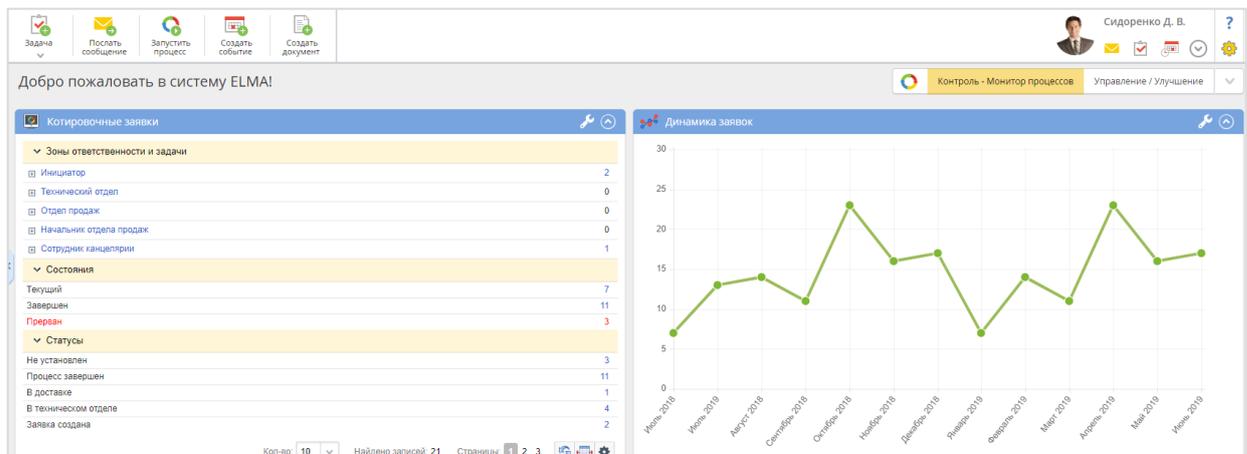


Рис. 32. Пример интерфейса для пользователя в роли Ответственного

Прежде всего проанализируйте, какие инструменты необходимы пользователям в соответствии с их ролью. В некоторых случаях для осуществления контроля будет достаточно базовых инструментов, например, фильтров и портлетов. Такие инструменты, как отчеты, метрики, показатели и пр., более трудоемкие в реализации, и их использование должно быть оправданным. За основу можно взять перечень текущих инструментов, которые реально используются пользователями. Но необходимо учитывать и существующие проблемы. В некоторых случаях можно попробовать комбинировать или разделять инструменты для более удобного использования.

3.1 Оперативный контроль

Оперативный контроль направлен на изучение информации о ходе исполнения бизнес-процесса. Для осуществления оперативного контроля можно использовать следующие инструменты: страница экземпляра процесса, фильтры задач и процессов, метрики и показатели экземпляра процесса, портлеты, монитор процессов.

В процессе можно настроить отображение переменных на странице экземпляра процесса, а также включить возможность поиска по ключевым переменным процесса. Скрывайте лишние и технические контекстные переменные, чтобы информационно не перегружать страницу экземпляра процесса.

Для задач процесса, которые имеют регламентированный срок исполнения, необходимо в настройках установить ограничение максимального срока выполнения задачи (например, время заполнения заявки оператором). В этом случае ответственному за исполнение задачи придет уведомление о превышении срока исполнения задачи (Рис. 33).

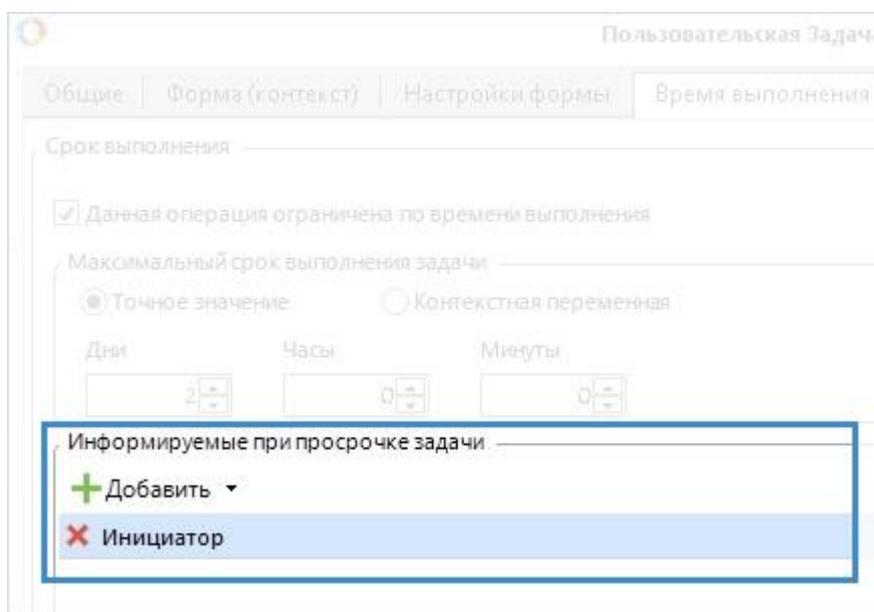


Рис. 33. Настройка информируемых при просрочке задачи

Для важных задач, срок исполнения которых влияет на весь бизнес-процесс (например, оплата счета), можно предусмотреть эскалацию по таймеру (Рис. 34). С помощью данного перехода можно настроить передачу задачи ответственному, чтобы гарантировать ее исполнение в срок. Это подходит только в тех случаях, когда срок исполнения задачи строго регламентирован



Рис. 34. Эскалация задачи по таймеру

Использование метрик и показателей процессов позволит быстро и легко осуществлять текущий контроль о состоянии процесса и его проблемных местах. Собирать данные имеет смысл только для показателей, необходимых для осуществления контроля процесса и предусмотренных регламентом (а не взятых в произвольном порядке). Данные о состоянии процессов можно вывести в соответствующем портлете на странице исполнителя и ответственного (Рис. 35). Также для удобства контроля можно добавить портлеты с фильтрами процессов

и задач. Это могут быть все задачи и процессы или, например, только те, у которых истекает срок исполнения.

Продажа товара №1449			
Название процесса	Оплата по счету (версия 5)		
Экземпляр процесса	Продажа товара №1449	Инициатор	Иванова С. Ю.
Дата запуска	06.08.2014 14:49	Ответственный	Иванова С. Ю.
Состояние	Завершен	Дата завершения	06.08.2014 14:52
<div style="background-color: #e1eef6; padding: 2px;"> ▼ Метрики и показатели </div>			
Показатель / Пользователь	План	Факт	Статус
Сумма счета - метрика		25 000,00	
Время оплаты счета	3 дн.	3 мин. 19 сек.	●
<div style="background-color: #e1eef6; padding: 2px;"> Список контекстных переменных История </div>			
Дата выставления счета	06.08.2014		
Срок действия счета	08.08.2014		
Номер счета	1449		
Контрагент	НЕО		
Сумма счета	25 000,00		

Рис. 35. Пример использования метрик и показателей экземпляра процесса

Не забывайте рассказать пользователям о базовых инструментах системы. Даже такой привычный инструмент, как **Карта процесса** может оказаться очень полезной для оперативного контроля.

3.2 Заключительный контроль

Заключительный контроль направлен на анализ сводной информации по бизнес-процессу. Основными инструментами в этом случае являются монитор процессов, метрики и показатели процесса, построение отчетов.

Монитор процессов позволяет получить сводную информацию по всем запущенным экземплярам процессов, на мониторинг которых пользователь имеет права (Рис. 36). Данный раздел позволяет организовать удобную навигацию в тех случаях, когда необходимо осуществить мониторинг большого количества процессов и уже нецелесообразно отображать их плоским списком.

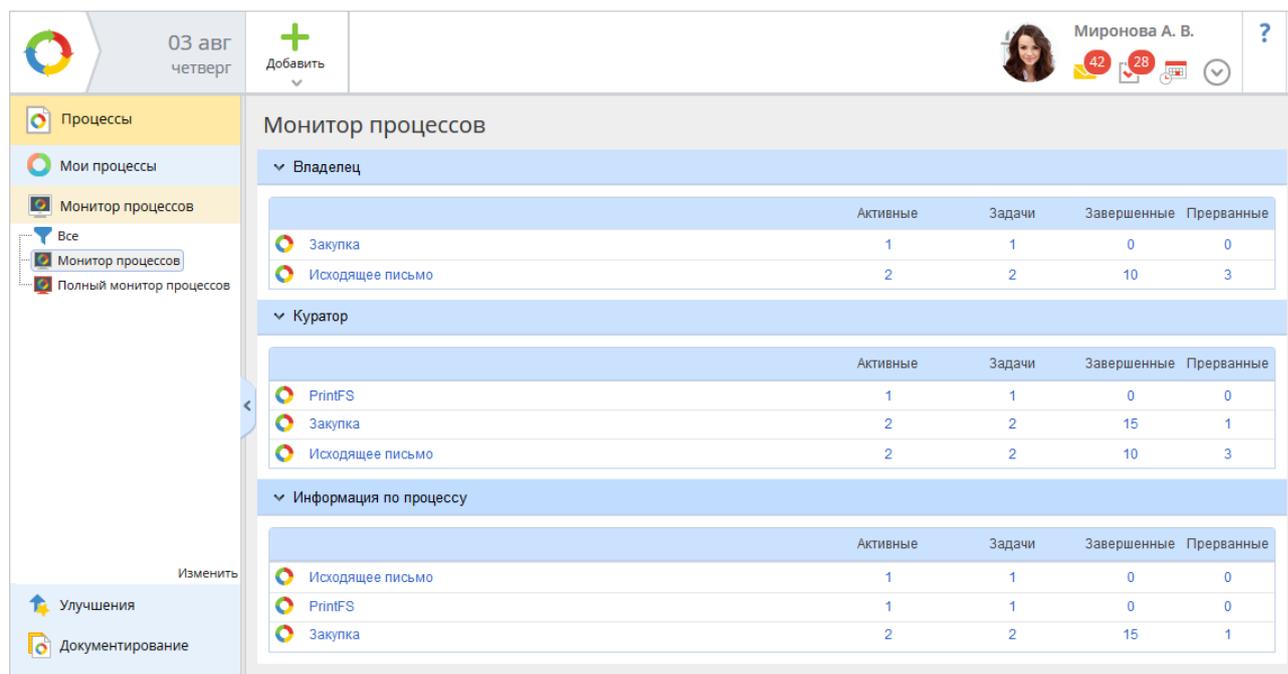


Рис. 36. Страница монитора процессов

Фильтры позволяют отображать список объектов (задач, процессов, документов и пр.) в соответствии с параметрами выборки (Рис. 37). Создавая пользовательские фильтры, можно настроить удобные списки для каждой роли бизнес-процесса. Добавив фильтр в портлет, можно, например, вывести все просроченные заявки выбранного подразделения.

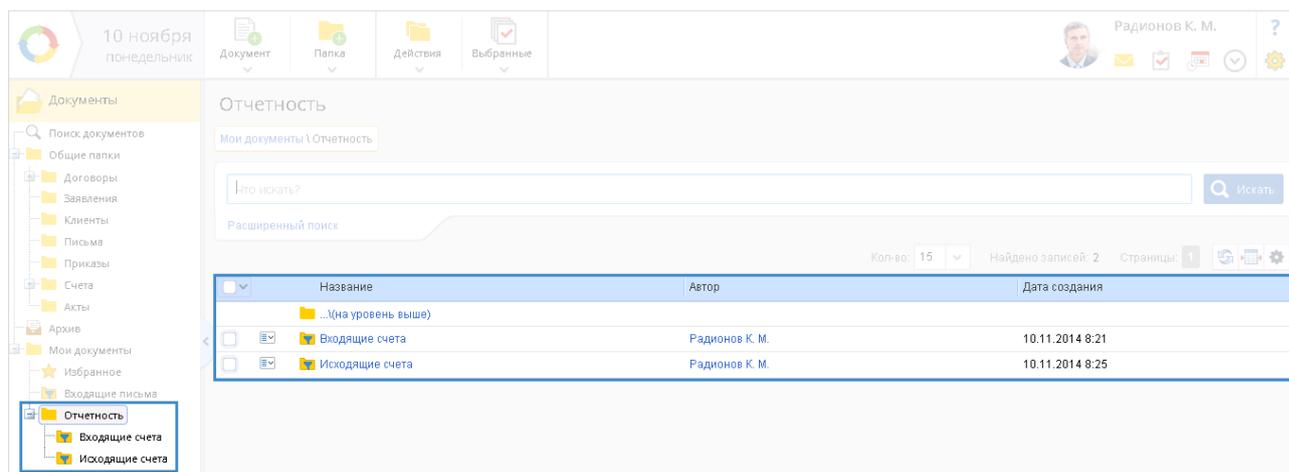


Рис. 37. Пример использования папок-фильтров

Метрики и показатели процессов позволяют настроить измерение и оценку параметров бизнес-процесса (Рис. 38). С помощью метрик и показателей можно, например, рассчитать средний чек за месяц, или доходы в разрезе по

сотрудникам. Это довольно гибкий, но трудоемкий инструмент. Используйте его в тех случаях, когда он действительно необходим.

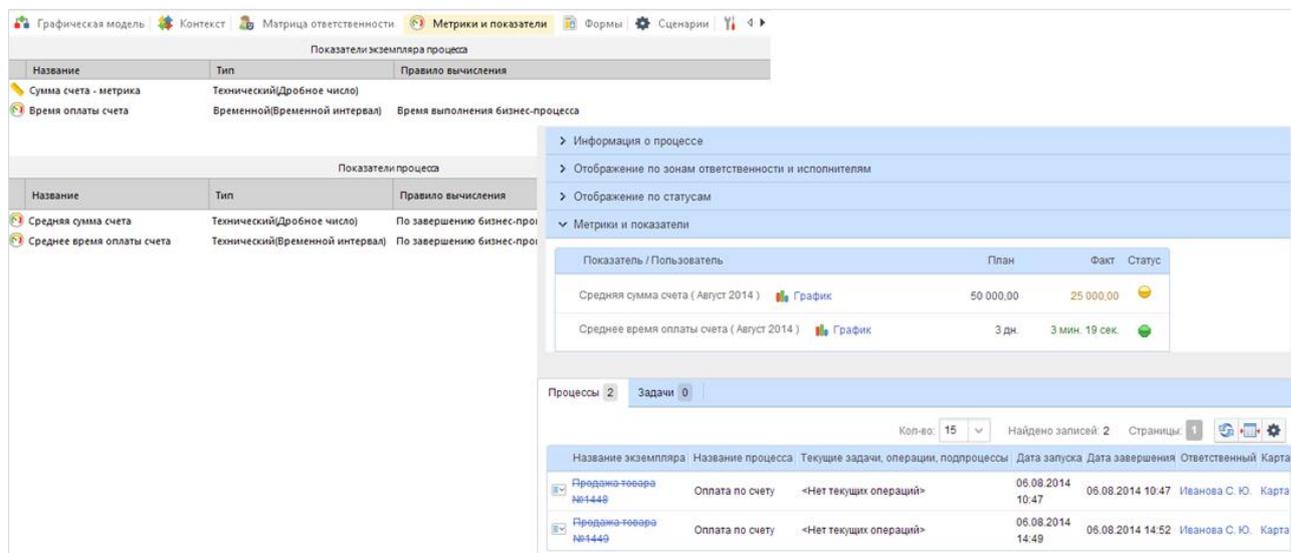


Рис. 38. Пример использования метрик и показателей процесса

Отчет – это выборка данных, систематизированных по определенным критериям и предназначенных для проведения анализа деятельности пользователей или компании в целом (Рис. 39). Отчет можно построить практически по любым данным системы, однако создание и поддержка этого инструмента довольно трудоемкие. Сформированный отчет можно выгрузить или распечатать в виде готового документа.

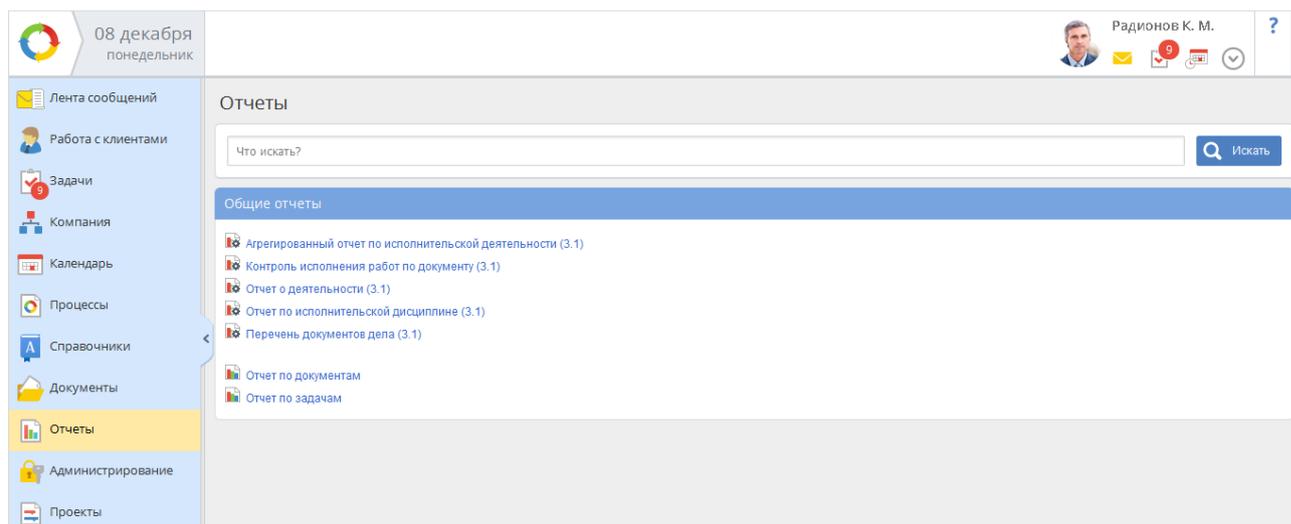


Рис. 39. Страница раздел «Отчеты»

Все это довольно объемные инструменты, которые позволяют производить глубокий анализ данных по работе в рамках бизнес-процесса. Собранная

информация должна предоставляться пользователю в удобном и читаемом виде. Для этого необходимо спроектировать и собрать интерфейс так, чтобы вся информация была доступна в пределах одного перехода. Если информации много, можно разбить ее по страницам.

Глава 4. Внутренний портал

4.1 Настройка внутреннего портала

Внутренний портал – это основное рабочее пространство для пользователей системы ELMA. Внутренний портал системы состоит из страниц, ссылок на разделы системы. Набор разделов и действий зависит от набора установленных модулей и доступа пользователей. Система ELMA позволяет производить гибкую настройку интерфейсов под определенные требования.

Режим настройки портала позволяет осуществить изменение и гибкую настройку интерфейсов без необходимости знания языков программирования.

Настройка цветовой темы позволяет привести внешний вид внутреннего портала в соответствии с брендовым стилем компании.

Хотя система позволяет производить настройку для каждого отдельного пользователя, нужно понимать, что каждый пользователь осуществляет свою работу в рамках определенной роли в системе. Поэтому лучше всего производить настройку интерфейсов для групп пользователей, в соответствии с ролевой моделью.

4.2 Настройка интерфейсов

Интерфейс системы – это основной инструмент пользователя. Он должен помогать пользователю выполнять свои основные задачи в соответствии с ролью и обязанностями.

Основные инструменты необходимо выносить "ближе". Более часто используемые инструменты должны располагаться выше (Рис. 40). Например, для отдела продаж на главной странице можно расположить шаблоны коммерческого предложения, ссылки на процесс регистрации потенциальных клиентов и т.д., а для работников канцелярии ссылки для регистрации документов и инструкции по обработке корреспонденции.

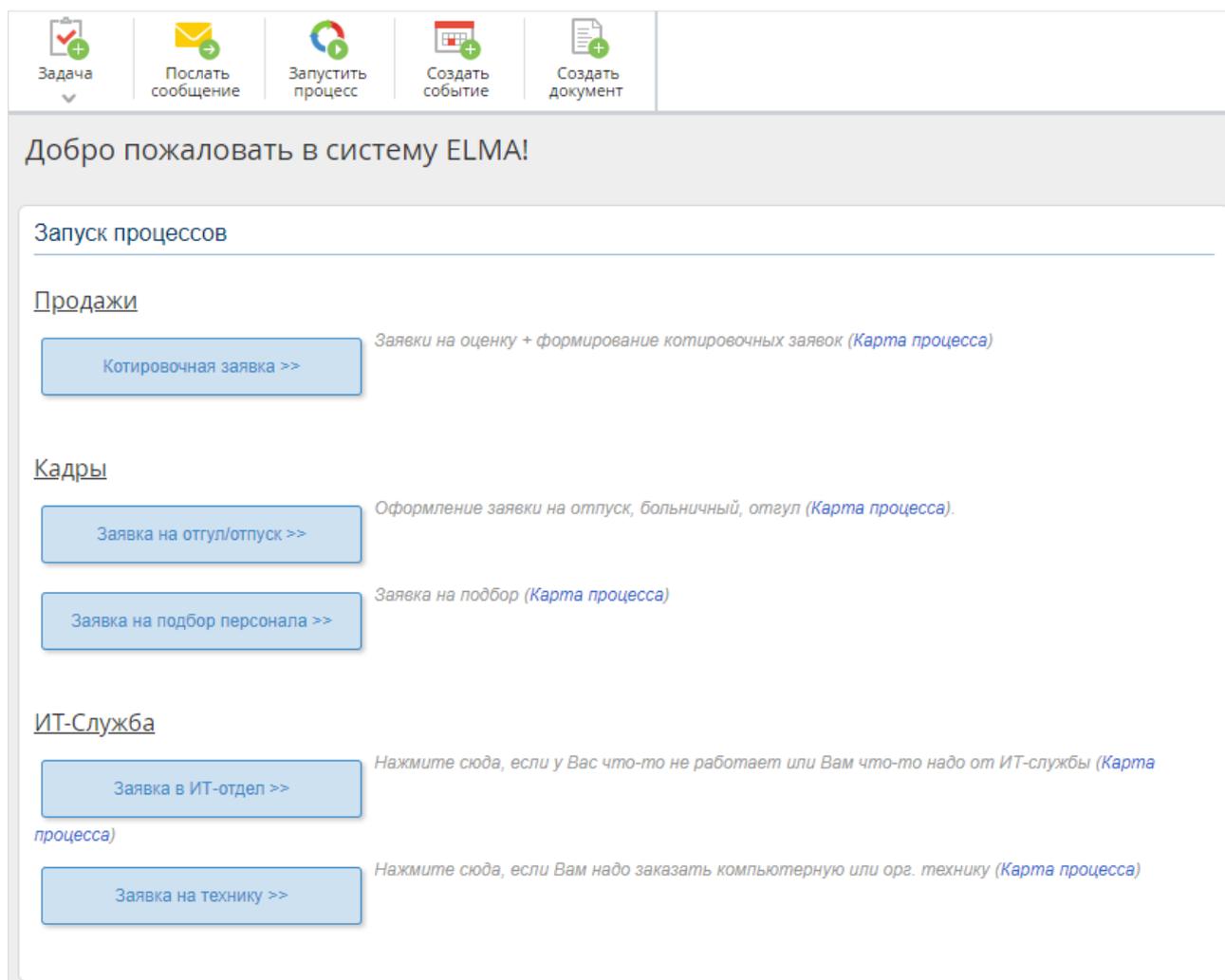


Рис. 40. Пример настройки интерфейса для быстрого доступа к информации

Изначально в системе существует два интерфейса: **Интерфейс по умолчанию** и **Lite**. Помимо них, можно добавить собственные интерфейсы.

Интерфейс по умолчанию присваивается всем пользователям по умолчанию и содержит настроенные страницы по умолчанию и разделы, согласно разделению прав доступа.

Lite интерфейс – специальный упрощенный интерфейс, созданный для категорий пользователей, которые пользуются лишь ограниченным набором функций, и насыщенность интерфейса только мешают работе (Рис. 41).

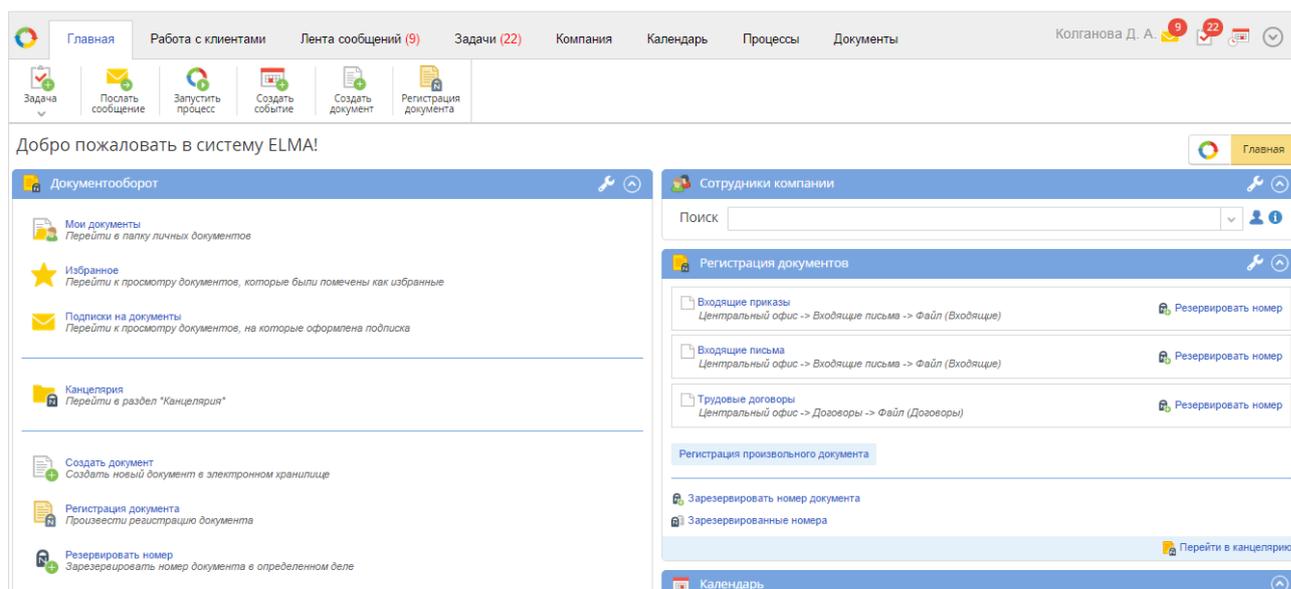


Рис. 41. Пример отображения Lite-интерфейса

Необходимо учитывать, что пользователи могут работать с системой с разных устройств. Интерфейс пользователя должен принимать во внимание особенности той или иной среды.

Функция **ELMA Rich E-Mail (ELMA.RE)** включает отправку на электронную почту письма, содержащего функционал системы ELMA, необходимый для выполнения задач непосредственно из почтового сервиса. В этом случае письмо будет содержать данные с формы задачи и кнопки переходов из задачи. При нажатии на поля для редактирования или кнопку перехода почтовый клиент перейдет в браузер для редактирования или отправки задачи (Рис. 42).

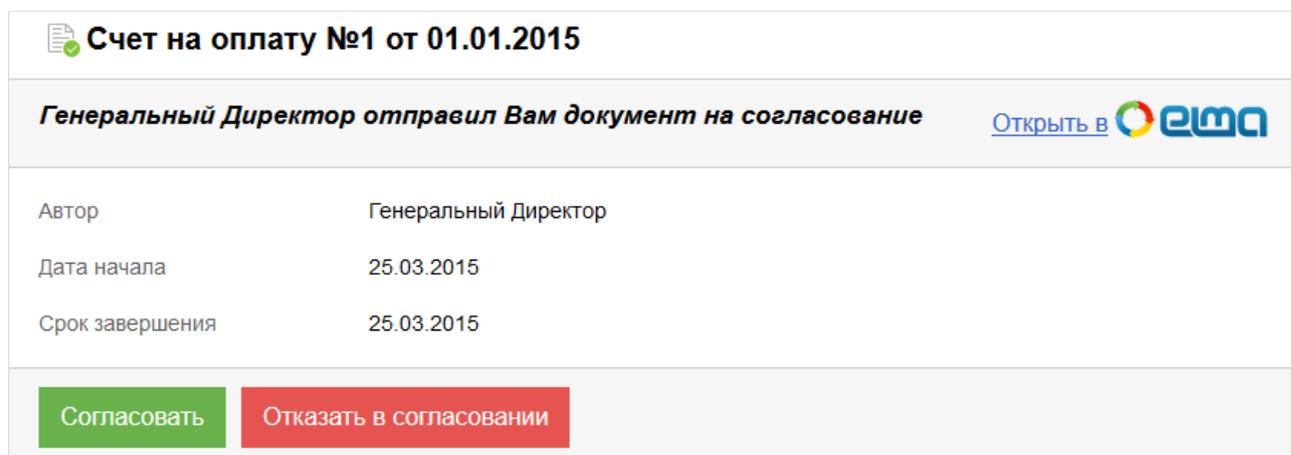


Рис. 42. Пример отображения Rich-оповещения для согласования документа

Функция **SMS-уведомление** включает отправку SMS-сообщения на указанный пользователем номер мобильного телефона об определенных действиях, производимых в системе и касающихся данного пользователя.

Программа **ELMA Агент** устанавливается локально на компьютер пользователя, и позволяет осуществлять работу с личными сообщениями и электронными документами, не заходя в веб-интерфейс ELMA (Рис. 43).

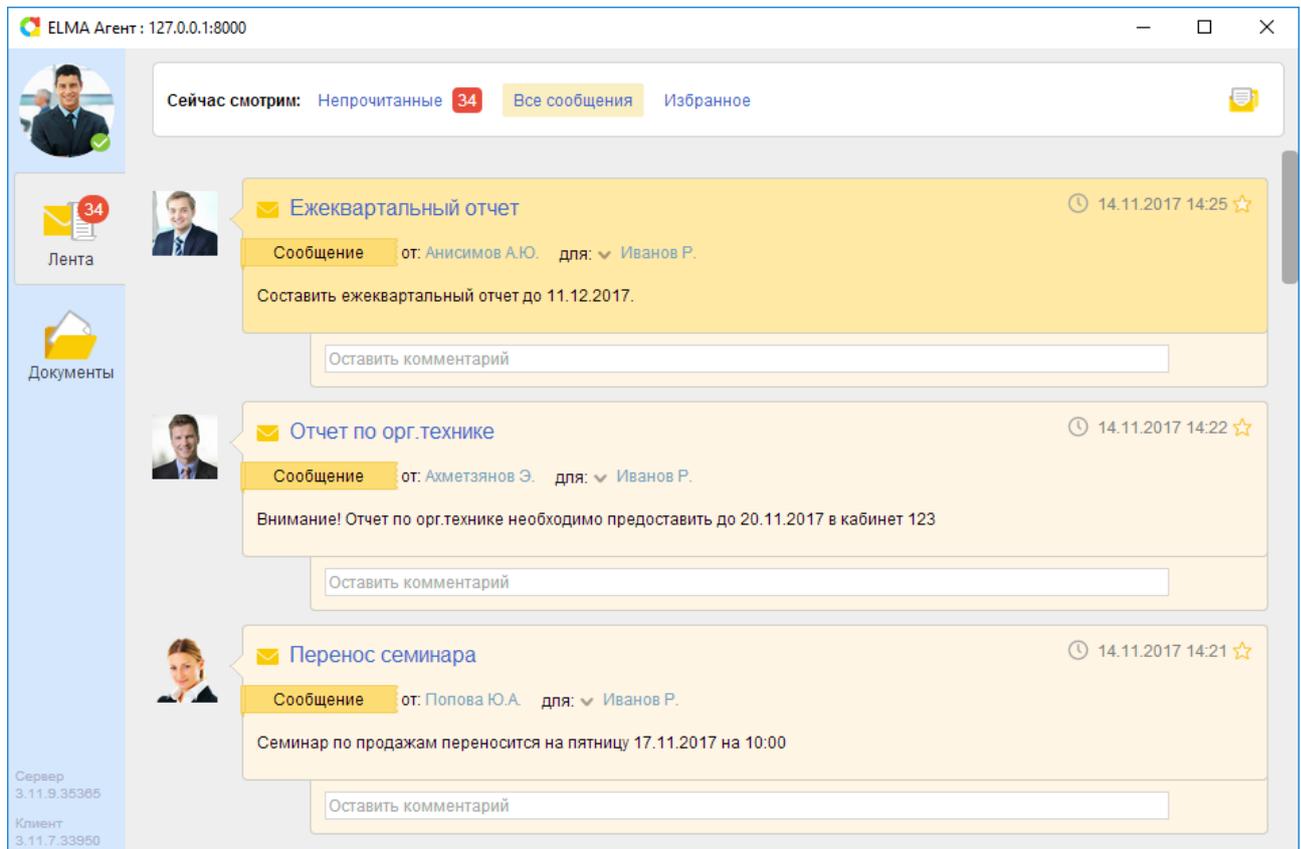


Рис. 43. ELMA Агент. Раздел "Лента"

Мобильное приложение **ELMA Mobile** позволяет всегда оставаться в курсе событий, происходящих в системе, контролировать, координировать работу и оперативно принимать решения, не имея прямого доступа к рабочему месту. Подробнее о мобильном приложении см. [справку по системе](#).

Помимо этого, в системе существует еще множество функций, позволяющих гибко настраивать интерфейс пользователей. Следует помнить, что любые пользовательские доработки интерфейса усложняют его дальнейшую поддержку – используйте их только когда это оправдано.

Глава 5. Документирование решения

Документирование решения позволит сохранить и использовать в будущем накопленный опыт. В документ необходимо включить не только описание концепции решения и список всех разработок, но и описание проблем, с которыми пришлось столкнуться в рамках проекта.

Подробный реестр всех разработок необходимо наполнять по ходу внедрения системы. Старайтесь давать подробное описание с инструкцией и примерами использования. Это не только позволит применить накопленный опыт, но и может быть использовано как инструкция при дальнейшей поддержке системы и для обучения пользователей.

По завершении проекта на основе реестра необходимо сформировать следующие документы:

Инструкция для пользователя – руководство для пользователей должно описывать функционал системы, используемого в рамках конкретных сценариев использования системы. Методическое пособие составляется для каждой группы пользователей в рамках ролевой модели и должно помочь обучиться и приступить к работе в системе. Для простоты и лучшего восприятия желательно сопровождать инструкцию пользователя понятными картинками с описанием.

Инструкция для администратора – подробное руководство по функционалу системы. Инструкция должна включать описание основных инструментов по настройке и поддержке системы. Документ предназначен для использования администратором системы.

Справка для поддержки – реестр всех разработок, существующих в данной системе с подробным техническим описанием. Документ предназначен для аналитиков и специалистов технической поддержки, которые будут осуществлять дальнейшее сопровождение, развитие и поддержку системы.

Все пожелания, которые были озвучены в ходе внедрения Заказчиком, но не были включены в исходную концепцию решения (не зафиксированные в техническом задании), необходимо фиксировать по ходу проекта. После завершения контура на основании этого списка можно будет сформировать следующий контур развития системы.

Глава 6. Обучение пользователей

Необходимо понимать, что даже привычные для пользователей задачи требуют время для адаптации в новой среде. Интуитивно понятный интерфейс позволит значительно упростить этот процесс. Желательно найти идеальный баланс между новым (но удобным и понятным) и прежним (но привычным) интерфейсом. Чем меньше возможностей для ошибки со стороны пользователей, тем лучше!

Удобное методическое пособие поможет познакомиться с системой и, возможно, снимет массу вопросов при поддержке. Но не смотря на безусловную пользу документации и инструкций для накопления опыта и обучения, конечный пользователь, скорее всего, никогда не прочтет эту инструкцию. Поэтому обучение пользователей работе в системе необходимо.

Помните, что большинство пользователей используют в системе только знакомый функционал. Если при обучении пользователю чего-то не показали, этого для него не существует. Но и перегружать пользователя лишней информацией о "дополнительных" возможностях системы не стоит. Каждому необходима информация только в рамках его функциональных обязанностей.

Несмотря на то, что пользователи чаще всего не будут проявлять инициативу в освоении системы, выходя за рамки привычного функционала, они обязательно попробуют использовать решение не так, как было задумано. Всегда необходимо предусматривать "защиту от дурака".

Глава 7. Чек-лист

Чек-лист – это набор действий для каждого проекта. Чек-лист может быть составлен на основании технического задания. В случае его отсутствия можно использовать концепцию решения, список функциональных требований или план проекта.

После окончания разработки обязательно сверьтесь с чек-листом, чтобы убедиться, что все пункты решения были реализованы в системе. Также стоит свериться с теорией внедрения (например, используйте Цикл Деминга для BPM или "ромашку" документооборота для ESM), чтобы убедиться, что ваше решение охватывает весь жизненный цикл бизнес-процесса, формируя полный контур.

Глава 8. Полезные ресурсы

Помимо текущего руководства, существуют аналогичные издания, в которых описываются основные возможности приложений системы ELMA:

- [Краткое руководство по Платформе ELMA BPM](#);
- [Краткое руководство по использованию внутреннего портала ELMA](#);
- [Краткое руководство по приложению ELMA ECM+](#);
- [Краткое руководство по приложению ELMA CRM+](#);
- [Краткое руководство по приложению ELMA Проекты+](#);
- [Краткое руководство по приложению ELMA KPI](#).

Данные руководства знакомят читателя с ключевыми особенностями системы. Подробное и исчерпывающее описание функционала системы ELMA содержится в справке, которая входит в поставку системы, а также всегда доступна в сети Интернет: <http://www.elma-bpm.ru/kb/help>.

Справочные материалы по каждому приложению разбиты на три категории: для пользователя, для внедрения и для администратора, что позволяет быстро найти нужную информацию.

Общее описание приложений и условия их приобретения доступны на **сайте ELMA**: <http://www.elma-bpm.ru>. Также на данном сайте всегда можно обратиться в компанию ELMA с помощью кнопки **Задать вопрос** или посредством онлайн-чата.

Ключевые возможности приложений и основные способы их использования продемонстрированы в **online демоверсии** <http://www.elma-bpm.ru/download>. Если же вы хотите подробнее изучить какое-либо из приложений, по этой же ссылке доступно скачивание демоверсии с такими же настройками, как и в online-версии.

Система ELMA постоянно развивается, и на базе Платформы и приложений разрабатываются компоненты, предназначенные для решения различных более узких и конкретных задач. Со списком и условиями приобретения таких готовых решений вы можете ознакомиться в **ELMA Store**: <https://store.elma-bpm.ru>.

При разработке собственных решений полезными окажутся материалы **Базы знаний ELMA**: <http://www.elma-bpm.ru/kb>.

Если же при работе в системе возникли вопросы технического характера, можно обратиться на сайт технической поддержки ELMA: <http://support.elma-bpm.ru>.

Для получения консультаций по системе ELMA или по сотрудничеству с компанией ELMA позвоните нам:

- Ижевск: +7 (3412) 93-66-93;
- Москва: +7 (499) 921-02-87;
- Казань: +7 (843) 567-17-69;
- Киев: +38 (067) 788-47-12;
- Алматы: +7 (727) 313-15-04.