

Описание процессов поддержания жизненного цикла, включая информацию об устранении неисправностей и совершенствовании ПО

УТВЕРЖДЕН
приказом ООО «ФТ-СОФТ» от 10.01.2023 № 01

РЕГЛАМЕНТ ПРОЦЕССА УПРАВЛЕНИЯ РАЗРАБОТКОЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

1. Общие сведения

1.1. Термины и определения

Проект	Уникальный процесс, состоящий из совокупности скоординированных и управляемых видов деятельности с датами начала и окончания, предпринятых для достижения цели, соответствующей конкретным требованиям, включая ограничения по срокам, стоимости и ресурсам.
Процесс	Совокупность взаимосвязанных или взаимодействующих видов деятельности, которые используют «вход» для получения намеченного результата (выхода).
Релиз	Устойчивый набор авторизованных изменений в программном продукте, построение, тестирование и развертывание которых выполняется совместно.
Рефакторинг	Внесение изменений в программный код, направленные на его улучшение и не связанные с изменением функциональности программного продукта.
Трудоемкость	Количество рабочих дней, необходимых одному разработчику на выполнение задачи.
Средства автоматизации процесса разработки	Набор программных средств, предназначенных для автоматизации деятельности команды разработки на всех этапах процесса разработки. К основным средствам автоматизации процесса разработки относятся: <ul style="list-style-type: none">– система управления версиями ПО;– система управления задачами;– система непрерывной интеграции;– система внутреннего документирования проекта.

1.2. Цель процесса управления разработкой ПО

Целью процесса управления разработкой ПО является обеспечение стабильно высокого качества новых версий программных продуктов, за счет применения организационных и технических мер, определенных данным регламентом.

1.3. Область действия Регламента

Действие регламента распространяется на все программные продукты, выпускаемые компанией.

1.4. Основные принципы регулирования Регламента

Для достижения целей процесса управления разработкой ПО должны соблюдаться следующие базовые принципы:

- Следование требованиям, определенным данным регламентом, на каждом этапе процесса.
- Информированность сотрудников подразделения разработки компании о наличии регламента, а также о внесении в него изменений.
- Регулярный пересмотр и внесение изменений в регламент, направленных на оптимизацию работы процесса управления релизами.

2. Стадии (этапы) процесса разработки

Процесс разработки является итерационным процессом. Результатом каждой итерации должна являться готовая, отделяемая от разработчиков, версия продукта. Каждая итерация включает следующие этапы работы:

- определение требований,
- проектирование,
- разработка и тестирование,
- выпуск релиза (версии, продукта).

3. Общий порядок действий в ходе выполнения фазы (этапа) проекта

3.1. Определение требований

На этапе определения требований представители команды разработки выполняют обследование объекта автоматизации при взаимодействии с представителями заказчика – специалистами предметной области.

Собранные в результате обследования сведения об объекте автоматизации должны быть формализованы с использованием средств моделирования, используемых командой разработки.

Результатом этапа определения требований должны являться следующие документы:

1. Протоколы проведения обследования объекта автоматизации. Протоколы должны быть согласованы с представителями заказчика.
2. Формализованное описание результатов обследования. Примером формализованного описания является представление требований в виде диаграммы прецедентов (Use Case) языка UML.
3. Техническое задание (разделы технического задания), согласованные с представителями заказчика.

На этом этапе должны быть определены цели и задачи, которые должно выполнять разрабатываемое программное обеспечение (ПО), а также запланированы способы проверки выполняемых ПО функций.

Формы согласования с заказчиком должны обеспечивать впоследствии возможность доказательства достигнутых договоренностей (подписанные бумажные документы, электронная почта и пр.).

На этапе определения требований должны использоваться следующие средства автоматизации процесса разработки:

- система управления задачами, для фиксации задач по обследованию и определению требований,
- система внутреннего документирования проекта, для хранения полученных требований в формализованном виде.

3.2. Проектирование

На этапе проектирования командой разработки разрабатываются проектные решения, реализующие требования, определенные на предыдущем этапе.

Разработка проектных решений выполняется в следующей последовательности:

1. Разработка высокоуровневого описания постановки задачи, которое должно содержать описание поведения, разрабатываемого ПО, описание бизнес- процессов и бизнес-ролей, состав функций, которые необходимо реализовать и т.д. Формальное описание алгоритмов и функций может быть выполнено в виде нотации BPMN, IDEF, диаграмм активностей, последовательности и состояний языка UML.
2. Разработка архитектуры ПО: состав компонентов, определение информационных потоков и т.д.
3. Определение состава задач для разработки. Задачи для разработки должны удовлетворять следующим требованиям:
 - a. Являться результатом декомпозиции высокоуровневых остановок задач.
 - b. Должны содержать конкретные постановки для реализации нового функционала разработчиками.
 - c. Задачи должны быть распределены на итерации, составляющие план выполнения разработки ПО, его версии или релиза.
4. Определение программных платформ, языков программирования, необходимость использования дополнительных сторонних библиотек.
5. Проведение подготовки к этапу разработки.

На этапе проектирования решения вырабатываются и принимаются архитектором системы, ведущими разработчиками, а также в результате совместных обсуждений с участием всех участников разработки. Результаты решений обязательно фиксируются в системе внутреннего документирования и системе управления задачами.

Результатом этапа определения требований должны являться следующие документы:

1. Общие постановки задач, включая схемы и диаграммы в перечисленных выше нотациях в системе внутреннего документирования.
2. Задачи для разработки создаются в системе управления задачами.
3. Необходимый набор документов Технического проекта согласно ГОСТ 34. Разработка документации Технического проекта должна проводиться на основании заранее подготовленных шаблонов документов. При необходимости документация Технического проекта согласуется с заказчиком ПО.

На этапе определения требований должны использоваться следующие средства автоматизации процесса разработки:

- система управления задачами, для создания задач разработчикам, проведения планирования итераций,
- система внутреннего документирования проекта, для хранения результатов проектирования ПО.

При проведении подготовки к этапу разработки должны быть подготовлены:

- проект в системе управления задачами,
- проект в системе управления версиями,
- базовая инфраструктура среды разработки, включая необходимый набор проектов, классов, стандартных и общих библиотек и других компонент, позволяющих осуществить сборку приложения, инфраструктура должна быть размещена в системе управления версиями,
- проект в системе непрерывной интеграции,
- стенды для тестирования разработанного ПО, моделирующие ИТ-инфраструктуру заказчика.

3.3. Разработка и тестирование

Процесс разработки должен быть дискретным и поделен на итерации длительностью от двух до четырех недель. Каждый этап должен заканчиваться выпуском следующей версии. Каждая версия должна содержать набор работоспособных компонент, которые могут быть представлены заказчику в качестве прототипа системы.

Для программных компонентов, вошедших в выпускаемую версию, должны выполняться следующие правила:

- компоненты должны корректно выполнять определенные в описании версии функции;
- программный код, реализующий указанные компоненты, должен иметь 100% покрытие тестами (т.е. все публичные методы должны иметь тесты, проверяющие работоспособность, как с корректными, так и с некорректными данными);
- тесты должны быть пригодны для автоматического тестирования и выполнены согласно технологии JUnit, NUnit;
- программный код должен иметь комментарии. Как минимум должны быть комментарии классов и публичных методов. Синтаксис комментария должен соответствовать правилам формирования комментария для систем автоматической сборки документации, характерной для платформы программирования;
- все задачи, запланированные в данной версии, должны быть зафиксированы в системе управления задачами, при этом для завершенных задач должен присутствовать подробный комментарий с описанием внесенных изменений;
- весь программный код должен быть размещен в системе управления версиями;
- результат сборки должен быть доступен из системы непрерывной интеграции;
- новая версия должна быть развернута на тестовом стенде.

Работа над версией считается завершенной после успешного выполнения функционального тестирования по определенным сценариям и контрольным примерам.

3.4. Выпуск релиза (версии, продукта)

Процедура выпуска релизов подробно описана в документе «Регламент управления релизами».

4. Средства автоматизации процесса разработки

Основной целью внедрения средств автоматизации процесса разработки является обеспечение высокого качества результатов разработки программного обеспечения и повышение эффективности работы команды разработки.

Внедрение средств автоматизации процесса разработки обеспечивает выполнение следующих показателей качества результатов работы команды разработки:

- стабильность и устойчивость работы выпускаемых программных продуктов,
- соблюдение сроков разработки,
- повторяемость результатов (возможность повторного использования компонент в других проектах),
- полнота документирования (информации в документах достаточно для самостоятельного изучения).

Средства автоматизации процесса разработки должны удовлетворять следующим требованиям:

- Должна быть возможность интеграции средств автоматизации процесса разработки с системами разработки (IDE).
- Внедрение средств автоматизации процесса разработки должно обеспечивать:
 - исключение многократной отчетности участников проекта,

- автоматизацию рутинных процессов,
- документирование всех этапов и артефактов проекта,
- унификацию подходов к выполнению критичных этапов проекта (сборка, тестирование, контроль и т.д.),
- реализацию эффективных методов оценки результативности команды и ее членов

УТВЕРЖДЕН
приказом ООО «ФТ-СОФТ» от 10.01.2023 № 01

РЕГЛАМЕНТ ПРОЦЕССА УПРАВЛЕНИЯ РЕЛИЗАМИ

1. Общие сведения

1.1. Термины и определения

Проект	Уникальный процесс, состоящий из совокупности скоординированных и управляемых видов деятельности с датами начала и окончания, предпринятых для достижения цели, соответствующей конкретным требованиям, включая ограничения по срокам, стоимости и ресурсам.
Процесс	Совокупность взаимосвязанных или взаимодействующих видов деятельности, которые используют «вход» для получения намеченного результата (выхода).
Релиз	Устойчивый набор авторизованных изменений в программном продукте, построение, тестирование и развертывание которых выполняется совместно.
Рефакторинг	Внесение изменений в программный код, направленных на его улучшение и не связанных с изменением функциональности программного продукта.
Трудоемкость	Количество рабочих дней необходимых одному разработчику на выполнение задачи.

1.2. Цель процесса управления релизами

Целью процесса управления релизами является обеспечение стабильно высокого качества новых версий программных продуктов за счет применения организационных и технических мер, определенных данным регламентом.

1.3. Область действия Регламента

Действие регламента распространяется на все программные продукты, выпускаемые компанией.

1.4. Основные принципы регулирования Регламента

Для достижения целей процесса управления релизами должны соблюдаться следующие базовые принципы:

- Тесная интеграция процесса выпуска релизов с процессом разработки программного обеспечения.

- Информированность сотрудников подразделения разработки компании о наличии регламента, а также о внесении в него изменений.
- Регулярный пересмотр и внесение изменений в регламент, направленных на оптимизацию работы процесса управления релизами.

2. Процесс управления релизами: этапы и участники

Процесс управления релизами включает следующие виды деятельности (этапы):

- Планирование и согласование релиза.
- Разработка и тестирование релиза.
- Информирование и выпуск релиза.

Участниками процесса являются:

- Руководство компании.
- Руководитель проекта.
- Группа разработки.
- Заинтересованные лица (сервис, маркетинг и пр.)

В таблице ниже представлено распределение ответственности участников процесса на каждом его этапе.

Таблица 1. Описание этапов процесса

Процесс управления релизами			
Этапы процесса →	Планирование и согласование релиза	Разработка и тестирование релиза	Информирование и выпуск релиза
Роли ↓			
Руководство компании	Утверждение плана релиза, бюджета проекта на этап	Приемка результатов разработки	Получение информации о результатах работы
Руководитель проекта	Разработка плана релиза	Подготовка и проведение приемо-сдаточных мероприятий	Отправка информации о результатах работы
Группа разработки		Выполнение разработки программных компонентов, документации, проведение тестирования	
Заинтересованные лица	Получение информации о планируемых работах		Получение информации о результатах работы

3. Реализация Регламента

3.1. Планирование и согласование релиза

Разработка плана релиза осуществляется руководителем проекта совместно с ведущими специалистами в составе команды разработки. План релиза утверждается представителями руководства компании.

При планировании релиза определяется:

- вид релиза,
- состав задач,
- ресурсное обеспечение,
- сроки выпуска.

3.1.1. Виды релизов

В зависимости от значимости и объема внесенных изменений выделяются следующие виды релизов:

- Новая версия программного продукта (значимый релиз): появление новых функций, значительная переработка существующей функциональности программного продукта.
- Развитие существующего функционала (малый релиз): развитие существующей функциональности, оптимизация программного кода (плановый рефакторинг), исправление ошибок.
- Исправление критических ошибок (оперативный релиз, hotfix): срочное исправление ошибок, оказывающих негативное влияние на работоспособность текущих версий программных продуктов, развернутых у заказчиков.

3.1.2. Идентификация релизов

Каждый релиз имеет уникальный идентификатор следующей структуры: XXX.YYY.ZZZ.DDD:

- XXX – порядковый номер Значимого релиза,
- YYY – порядковый номер Малого релиза внутри Значимого релиза,
- ZZZ – порядковый номер Оперативного релиза внутри Малого релиза,
- DDD – код промежуточной версии, используется разработчиками, для передачи на тестирование промежуточных результатов разработки релиза.

3.1.3. Описание релиза

В описание релиза должны входить:

- Уникальный идентификатор релиза.
- Общее описание, содержащее основное направление работы (цель релиза) в виде краткого описания, например, «Развитие интерфейса пользователя», «Разработка подсистемы интеграции и развитие подсистемы аутентификации», «Повышение производительности работы».
- Состав релиза: высокоуровневый список задач.
- Общая трудоемкость.
- Срок выпуска релиза.
- Значимость релиза: вычисляется как среднее значение значимости задач, входящих в состав релиза. Может принимать значение от 1 (низкая значимость) до 4 (высокая значимость).

3.1.4. Состав релиза

В состав релиза входят высокоуровневые задачи, решение которых обеспечивает достижение заявленной цели релиза.

Высокоуровневые задачи (user story) определяют дальнейшее развитие продукта в виде требований к разработке новой функциональности или для рефакторинга. Высокоуровневые задачи далее декомпозируются на конкретные задачи для разработчиков, выполнение которых контролируется в рамках системы управления задачами, используемой в компании. В описании релиза задачи разработчиков не приводятся.

Отдельной высокоуровневой задачей релиза является задача «Исправление ошибок», включающая активности, связанные с устранением ошибок, проблем и замечаний, обнаруженных в работе программного продукта. В формулировке задачи «Исправление ошибок» могут присутствовать ссылки на идентификаторы описаний ошибок, проблем и замечаний из системы управления задачами.

Полный список высокоуровневых задач находится общем плане разработки программного продукта, созданном на основе технического задания.

В описании каждой задачи должны присутствовать трудоемкость ее решения и ее значимость: цифровая характеристика важности решения этой задачи для проекта в целом, принимающая значения от 1 (низкая значимость) до 4 (высокая значимость). Трудоемкость задачи определяется на основании опыта руководителя проекта или оценки разработчика – исполнителя задачи, с последующим согласованием этой оценки руководителем подразделения.

Задачи в состав релиза включаются на основании оценки их значимости, таким образом, чтобы суммарная трудоемкость задач, включенных в релиз, не превышала срок его завершения.

3.1.5. Процедура согласования

Состав высокоуровневых задач, общие трудозатраты и сроки релиза проходят согласование на уровне руководства компании (или ответственного лица, куратора процесса разработки, определенного руководством компании).

3.1.6. Завершение выполнения релиза

Этап разработки релиза считается завершенной при завершении работы над всеми задачами, входящими в состав релиза.

В случае, если по каким-либо причинам все задачи релиза не могут выполнены в установленные для выхода релиза сроки, по согласованию с руководством компании, выполняется один из сценариев:

- В случае, если важно, чтобы все задачи были выполнены, производится коррекция сроков выпуска релиза.
- В случае, если важно, чтобы была выдержана заявленная дата выпуска релиза, производится коррекция состава релиза, при этом этап разработки завершается, невыполненные задачи переносятся в следующий релиз.

После завершения разработки релиза, результаты работы передаются на тестирование, помимо собственно программных компонентов группе тестирования при этом передается список изменений, вошедших в релиз.

Руководитель проекта должен экспертным путем определять время, необходимое для тестирования результатов работы над релизом, для обеспечения соблюдения сроков выхода релиза.

3.2. Разработка и тестирование релиза

Выполнение процесса разработки релиза обеспечивают следующие средства автоматизации процесса разработки:

- Система контроля версий.
- Система управления задачами.
- Сервер непрерывной интеграции.
- Стенды для тестирования результатов разработки:
 - стенд для тестирования промежуточных результатов,

- стенд релизов.

Разработка релиза ведется следующим образом:

- На основании высокоуровневых задач релиза формируется список задач для разработчиков в системе управления задачами.
- Результаты разработки в рамках решения задач сохраняются в ветке разработки в системе управления версиями. Сборка промежуточных результатов выполняется сервером непрерывной интеграции и разворачивается на стенде для тестирования промежуточных результатов.
- После выполнения всех задач, определенных для релиза (или достижения даты завершения релиза, в зависимости от значимости одного из этих событий), формируется отдельная ветка релиза с целью отделения полученных результатов от основного процесса разработки.
- Результаты сборки ветки релиза публикуются на стенде релиза. Ошибки, найденные в результате тестирования, исправляются в ветке релиза.
- После завершения всех работ по релизу выполняется слияние ветки релиза с основной веткой (мастер) и веткой разработки, ветка релиза закрывается, производится выполнение результирующей сборки в основной ветке. Результаты сборки предоставляются заинтересованным лицам для развертывания.
- В случае обнаружения критических проблем между выпусками релизов производится выпуск оперативных релизов (hotfixes), исправляющих выявленные ошибки. Разработка оперативных релизов ведется в отдельных ветках, которые сливаются с основной веткой (мастером) после завершения работы.

3.2.1. Приемка релиза

В зависимости от вида релиза приемка результатов работы осуществляется комиссией (для значимого или малого релизов), состав которой согласуется руководством компании или выполняется сотрудниками подразделения разработки компании в форме тестирования результатов разработки.

Результатом приемки релиза является протокол, содержащий решение о возможности выпуска релиза или список замечаний, сроки их устранения и способ приемки устраненных замечаний (заседание комиссии или отчет об устранении замечаний, подготовленный специалистами отдела разработки).

3.3. Информирование и выпуск релиза

3.3.1. Информирование

После полного завершения работы над релизом, заинтересованным лицам рассылается информация о выходе нового релиза, включая сопроводительный документ и варианты получения и развертывания обновлений, содержащих данный релиз.

Сопроводительный документ релиза, содержит следующую информацию:

1. Номер релиза.
2. Общее описание релиза (тема или цели выполнения работы).
3. Дата выхода релиза.
4. Список задач релиза.
5. Список исправлений (включенных в релиз исправлений ошибок, проблем, замечаний).
6. Состав инсталляционного пакета.
7. Дополнительные инструкции по настройке (при необходимости).

3.3.2. Выпуск релиза

На этапе выпуска релиза, готовые программные компоненты и документация предоставляются заинтересованным лицам для развертывания и дальнейшего использования программного продукта.

Распространение и развертывание релиза осуществляется способом, специфичным для конкретного программного продукта, как правило в виде инсталляционного пакета.

При любом способе распространения обновлений артефакты для них (инсталляционные пакеты, библиотеки, архивы и пр.) должны быть подготовлены автоматически сервером непрерывной интеграции, развернутым в компании

УТВЕРЖДЕН
приказом ООО «ФТ-СОФТ» от 10.01.2023 № 01

РЕГЛАМЕНТ ПРОЦЕССА ПРОВЕДЕНИЯ ТЕСТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

1. Общие сведения

1.1. Термины и определения

Unit (модульное) тестирование	Тестирование отдельных компонентов, модулей или методов, выполняющееся при помощи разработанных Unit- тестов
Автоматическое тестирование	Режим проведения испытаний программного обеспечения, при котором проверки производятся с использованием средств автоматизации процесса разработки, без участия разработчиков или специалистов группы тестирования
Интеграционное тестирование	Проверка корректности взаимодействия смежных доработок, подсистем друг с другом и/или с внешними системами
Нагрузочное тестирование	Тестирование производительности, сбор показателей и определение производительности и времени отклика подсистемы ответ на внешний запрос с целью установления соответствия требованиям, предъявляемым к данной подсистеме
Регрессионное тестирование	Вид тестирования подсистем, направленный на обнаружение ошибок в ранее протестированных версиях и/или на подтверждение работоспособности критически важного функционала.

Ручной режим тестирования	Режим проведения испытаний, при котором проверки производятся с участием разработчиков или специалистов группы тестирования
Программа и методика испытаний (ПМИ)	Документ содержит список проверок программной системы на соответствие заявленным в Техническом задании функциям
Проект	Уникальный процесс, состоящий из совокупности скоординированных и управляемых видов деятельности с датами начала и окончания, предпринятых для достижения цели, соответствующей конкретным требованиям, включая ограничения по срокам, стоимости и ресурсам.
Процесс	Совокупность взаимосвязанных или взаимодействующих видов деятельности, которые используют «вход» для получения намеченного результата (выхода).
Релиз	Устойчивый набор авторизованных изменений в программном продукте, построение, тестирование и развертывание которых выполняется совместно
Рефакторинг	Внесение изменений в программный код, направленные на его улучшение и несвязанные с изменением функциональности программного продукта
Средства автоматизации процесса разработки	<p>Набор программных средств, предназначенных для автоматизации деятельности команды разработки на всех этапах процесса разработки. К основным средствам автоматизации процесса разработки относятся:</p> <ul style="list-style-type: none"> - система управления версиями ПО, - система управления задачами, - сервер непрерывной интеграции, - система внутреннего документирования проекта.
Функциональное тестирование	Тестирование ПО с целью проверки реализованных функциональных требований, выполнения прикладных функций, решающих задачи пользователя

1.2. Цель процесса проведения тестирования ПО

Целью процесса проведения тестирования ПО является обеспечение стабильно высокого качества

новых версий программных продуктов, за счет проведения тестирования выпускаемых версий ПО, выполняемого в автоматическом и ручном режимах.

1.3. Область действия Регламента

Действие регламента распространяется на все программные продукты, выпускаемые компанией.

1.4. Основные принципы регулирования Регламента

Для достижения целей процесса управления разработкой ПО должны соблюдаться следующие базовые принципы:

- Следование требованиям, определенным данным регламентом, на каждом этапе процесса.
- Информированность сотрудников подразделения разработки компании о наличии регламента, а также о внесении в него изменений.
- Регулярный пересмотр и внесение изменений в регламент, направленных на оптимизацию работы процесса.

1.5. Участники процесса

В процессе проведения тестирования участвуют следующие специалисты:

- тестировщики, специалисты группы тестирования – специализированного подразделения, в задачи которого входит разработка методики, технологии и собственно проведение тестирования с целью контроля качества, разрабатываемого ПО;
- разработчики и аналитики в составе группы разработки выполняют тестирование в рамках выполнения своих непосредственных обязанностей;
- аналитики, специалисты предметной области, привлекаются к тестированию с целью определения корректности работы прикладных модулей.

2. Режимы и виды тестирования ПО

В рамках работ по выпуску релизов ПО тестирование выполняется в следующих режимах:

- автоматическое Unit-тестирование, выполняется на основании разработанных Unit-тестов в процессе автоматической сборки;
- автоматическое функциональное тестирование, выполняющееся сервером автоматической сборки с заданной периодичностью;
- ручное регрессионное функциональное тестирование выполняется однократно с использованием актуальной Программы и методике испытаний (ПМИ) перед выпуском релиза;
- нагрузочное тестирование, выполняется однократно перед выпуском релиза.

Функциональное тестирование проводится при добавлении новой или изменении существующей функциональности, с целью проверки доработанного программного обеспечения новым или измененным функциональным требованиям. Функциональное тестирование может проводиться как в ручном, так и в автоматическом режиме.

Нагрузочное тестирование проводится с целью выявить соответствие ПО требованиям производительности, определения пиковой и средней нагрузки, стабильности системы в целом.

Регрессионное тестирование проводится с целью проверки устойчивой работы обновленной версии с учетом новой функциональности и/или исправления дефектов. Регрессионное тестирование может проводиться как в ручном, так и в автоматическом режиме.

Интеграционное тестирование проводится в случае необходимости провести проверку корректности взаимодействия смежных систем друг с другом.

Регрессионное и интеграционное тестирование выполняются в ходе проведения функционального тестирования в автоматическом и ручном режимах.

Работы по проведению функционального, интеграционного и регрессионного тестирования осуществляют специалисты группы тестирования или группы разработки.

3. Порядок проведения тестирования

3.1. Автоматическое Unit-тестирование

Для проведения Unit-тестирования разработчики создают Unit-тесты с использованием специализированных фреймворков (JUnit, NUnit и т.д.). Unit-тесты создаются по факту внесения любых изменений в программный код, с целью подтверждения корректности работы выполненных изменений, а также отсутствия негативного влияния сделанных изменений на другие модули программной системы.

Как минимум, Unit-тесты должны покрывать все публичные методы модуля (класса).

Unit-тест запускается разработчиком для проверки корректности разработанного программного кода.

Все разработанные Unit-тесты запускаются в пакетном режиме сервером непрерывной интеграции в процессе выполнения сборки, после размещения изменений программного кода в системе управления версиями ПО. Сборка считается выполненной успешно только после успешного выполнения полного пакета Unit-тестов.

3.2. Автоматическое функциональное тестирование

Для проведения автоматического функционального тестирования специалистами группы тестирования создаются тесты с использованием специализированных средств для разработки функциональных тестов.

В рамках функционального тестирования выполняется проверка корректности работы функций пользователя, как каждой в отдельности, так и при их совместной работе, например, в рамках автоматизации бизнес-процесса.

В рамках автоматического функционального тестирования также проводится проверка корректности работы интерфейса пользователя.

Работа функций пользователя считается корректной, в случае если она соответствует функциональности, заявленной в Техническом задании на разработку программной системы.

3.3. Ручное функциональное тестирование

Ручное функциональное тестирование выполняется в целях, аналогичных автоматическому, но тестирование проводят специалисты группы тестирования на основании ПМИ.

К ручному функциональному тестированию, также могут быть привлечены аналитики группы разработки и бизнес-аналитики в рамках проверки корректности работы прикладных модулей.

3.4. Нагрузочное тестирование

Нагрузочное тестирование выполняется с целью проверки соответствия выпускаемой версии ПО показателям назначения, определенным в Техническом задании, в части производительности и устойчивости к нагрузкам.

Нагрузочное тестирование выполняется с использованием нагрузочных тестов, разработанных с использованием специализированного программного обеспечения (Apache JMeter и т.д.) специалистами группы тестирования.

Запуск тестов может выполняться в ручном режиме, в ходе мероприятий по выпуску релизов, а также в автоматическом режиме по расписанию с использованием сервера непрерывной интеграции.

4. Разработка и актуализация Программы и методики испытаний

Разработку и/или актуализацию ПМИ при выпуске очередного релиза инициирует руководитель проекта. Доработка ПМИ выполняется аналитиком группы разработки.

Актуализация ПМИ проводится аналитиком группы разработки по результатам внесения изменений в ПО перед выпуском новой версии ПО.

В состав ПМИ должны входить задачи, выполненные на момент завершения работы над текущей версией ПО.

Источником для состава тестируемых функций является система управления задачами.

УТВЕРЖДЕН
приказом ООО «ФТ-СОФТ» от 10.01.2023 № 01

РЕГЛАМЕНТ УСТРАНЕНИЯ НЕСООТВЕТСТВИЙ В ХОДЕ ЭКСПЛУАТАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ «СОЛО. ДОКУМЕНТООБОРОТ»

1. СПИСОК ТЕРМИНОВ И ОПРЕДЕЛЕНИЙ

- 1.1. **Соло** – программное обеспечение (ПО) «Соло. Документооборот».
- 1.2. **Конечный пользователь** – Юридическое лицо, которое на законных основаниях владеет неисключительными правами на использование ПО «Соло. Документооборот».
- 1.3. **Исполнитель** – ООО «ФТ-СОФТ».
- 1.4. **АИС** – автоматизированная информационная система Исполнителя.
- 1.5. **БД** – база данных.
- 1.6. **Сервисный центр** – служба технической поддержки Исполнителя.
- 1.7. **Услуга** – перечень предоставляемых услуг в объеме, установленном Программой технической поддержки (Приложение 1).
- 1.8. **Заявка** – сообщение, направляемое уполномоченным представителем Конечного пользователя из утвержденного перечня Исполнителю в электронной или устной форме, в котором фиксируется возникшая проблема или вопрос, подлежащие решению.
 - 1.9. **Ответственный** – специалист Исполнителя, назначенный для решения Заявки.
 - 1.10. **Рабочий день** – промежуток времени с понедельника по пятницу с 08:00 до 20:00 (UTC +5:00), за исключением выходных и праздничных дней, согласно законодательству Российской Федерации.
 - 1.11. **Рабочий час** – астрономический час в пределах Рабочего дня.
 - 1.12. **Нерабочее время** – промежуток времени с понедельника по пятницу с 20:00 до 08:00 (UTC +5:00), а также выходные и праздничные дни, согласно законодательству Российской Федерации.
 - 1.13. **Работоспособность** – ПО считается работоспособным при его работе с сохранением основных технических параметров, указанных в технической документации, выполнение которых обеспечивает нормальное применение ПО по назначению и использование для выполнения необходимых функций.
 - 1.14. **Время обслуживания** – режим времени, в рамках которого Исполнитель предоставляет Услуги. Варианты времени обслуживания: работа над Заявкой ведется в режиме 12x7 в рабочие дни с 08:00 до 20:00 (UTC +5:00), за исключением выходных дней и общероссийских праздников; 24x7 – работа над Заявкой ведется круглосуточно.
 - 1.15. **Время реакции** – время, в течение которого ответственный специалист начнет работу над Заявкой и, в случае необходимости, свяжется с представителем Конечного пользователя для выдачи рекомендаций или запроса дополнительной информации. Отсчет времени реакции начинается с момента получения Заявки. Время реакции определяется заказанной Услугой. Для Времени обслуживания 12x7 при поступлении запроса позже 20:00 (UTC +5:00) Заявка считается полученной на следующий рабочий день в

08:00 (UTC +5:00).

1.16. **Время в работе** – время, в течение которого ответственный специалист выполняет работу над Заявкой, т.е. общее время с момента регистрации Заявки до момента ее закрытия.

1.17. **Время восстановления** – период времени с момента регистрации Заявки до момента восстановления работоспособности ПО.

1.18. **Периодичность выполнения** устанавливает периодичность оказания услуг. Конкретная дата проведения работ устанавливается по согласованию с Конечным пользователем.

1.19. **Приоритет** – параметр, отражающий важность проблемы для бизнеса и пользователя. Приоритет определяется представителем Конечного пользователя путем оценки влияния проблемы на работоспособность ПО при подаче Заявки в соответствии со следующим описанием:

- **Критичный приоритет** – ПО не работает, что приводит к остановке всех или одной из обслуживаемых систем.

- **Высокий приоритет** – частично нарушена работоспособность, либо значительно снижена отказоустойчивость или производительность ПО Конечного пользователя, но при этом создается критическая ситуация, при которой полная потеря работоспособности всего технологического процесса Конечного пользователя может произойти в любой момент.

- **Средний приоритет** – незначительно нарушена работоспособность, либо производительность ПО, что не оказывает влияния на бизнес Конечного пользователя.

- **Низкий приоритет** – ошибка, которая несущественно ограничивает использование ПО для выполнения необходимых функций, либо требуется консультация специалиста по техническим вопросам, связанным с эксплуатацией, конфигурированием, управлением исправным ПО, поиском причины возникновения неисправности.

ПРИМЕЧАНИЕ: заявленный Конечным пользователем при открытии Заявки приоритет может быть изменен специалистом Исполнителя в случае уточнения в ходе выполнения Заявки фактического влияния заявленной проблемы на работоспособность ПО, а также в случае снижения влияния проблемы на работоспособность ПО.

1.20. **Способ взаимодействия** – возможные способы коммуникации между Конечным пользователем и ответственным специалистом Исполнителя. Услугой предусмотрены следующие способы взаимодействия:

- **Отправка сообщения на электронную почту;**
- **Звонок на горячую линию**

Контактная информация указана в п.3.1.

1.21. **Способ обработки Заявок** – возможные способы обработки Заявок определяются заказанной Услугой. В случае, если Услугой предусмотрено несколько способов обработки Заявок, то способ выбирается ответственным специалистом Исполнителя:

- **Дистанционно** – при данном способе работы над Заявкой взаимодействие Исполнителя и Конечного пользователя осуществляется по электронной почте и телефону.

- **Удаленно** – при данном способе работа над Заявкой осуществляется с помощью удаленного доступа к ресурсам Конечного пользователя.

2. ОБЩИЕ ПОЛОЖЕНИЯ

2.1. Настоящий документ определяет регламент оказания услуг технической поддержки (далее – Регламент).

2.2. Основной целью Регламента является установление порядка взаимодействия специалистов Конечного пользователя со специалистами Исполнителя.

3. КОНТАКТНАЯ ИНФОРМАЦИЯ

3.1. Контакты Сервисного центра:

- **Электронная почта:** main@ft-soft.com.
- **Телефон:** +7 (343) 312-34-51.
- **Юридический адрес:** 620137, г. Екатеринбург, ул. Блюхера 45 - 27

3.2. В случае изменения Исполнителем контактной информации, указанной в Регламенте, Исполнитель должен сообщить Конечному пользователю новую контактную информацию по электронной почте либо по телефону с обязательным дублированием информации электронным сообщением.

4. ОТКРЫТИЕ ЗАЯВКИ

4.1. Способы и время подачи Заявок

- Заявки в Сервисный центр принимаются по электронной почте, телефону (пункт 3.1.) от уполномоченного представителя Конечного пользователя.

- **По телефону:** круглосуточно.

- **По электронной почте:** в рабочий день.

4.2. Содержание Заявки

Заявка подается по установленной форме (Приложение № 2), при заполнении которой указываются следующие сведения:

- Номер договора, номер лицензии, сертификата технической поддержки.
- Название организации Конечного пользователя.
- Контактная информация (ФИО, телефон, электронный адрес) обратившегося.
- Содержание обращения.

4.3. Регистрация Заявки

- Регистрация Заявки начинается с момента получения Заявки.
- При регистрации АИС автоматически присваивает Заявке номер и отправляет уведомление о регистрации на указанный контактный электронный адрес Конечного пользователя.

5. РАБОТА НАД ЗАЯВКОЙ

5.1. Ответственный берет Заявку в работу только во время обслуживания (п. 1.13) и в срок не более обозначенного в Программе технической поддержки. Отсчет Времени реакции начинается с получения всей информации в соответствии с содержанием Заявки (пункт 4.2.).

5.2. Ответственный на основании предоставленных данных производит диагностирование проблемы, либо запрашивает у инициатора Заявки (при необходимости) дополнительную информацию.

5.3. Производится уведомление инициатора Заявки по указанным в Заявке контактными данным о возможных вариантах решения.

5.4. При необходимости выполнения работ способом «Выезд» производится согласование с инициатором Заявки возможности, даты и времени допуска на место эксплуатации (если предусмотрено Программой технической поддержки).

5.5. Организуется прибытие специалистов исполнителя на место эксплуатации ПО в согласованное время (если предусмотрено Программой технической поддержки).

5.6. Производятся работы по устранению неисправности.

5.7. Количество персонала, задействованного в процессе сопровождения: 5 человек.

6. ВЗАИМОДЕЙСТВИЕ С КОНЕЧНЫМ ПОЛЬЗОВАТЕЛЕМ

6.1. В случае необходимости получения дополнительной информации от Конечного пользователя для продолжения работы над Заявкой Ответственный обращается с запросом к специалисту Конечного пользователя. По своему усмотрению Ответственный может выбрать способ обращения: по телефону или с помощью отправки электронной почты с обязательной фиксацией факта обращения (запись телефонного разговора, подтверждение о получении запроса Конечным пользователем по электронной почте или телефону (с записью разговора)).

6.2. При получении запроса о предоставлении дополнительной информации специалист Конечного пользователя обязан оказать содействие Ответственному в дистанционном решении проблемы и предоставить всю информацию, необходимую для осуществления технической поддержки, а также для определения подходящего уровня обслуживания, запустить тесты самодиагностики и/или установить и запустить другие диагностические средства и программы, в соответствии с инструкциями, предоставляемыми специалистом Исполнителя.

6.3. При отсутствии обратной связи от Конечного пользователя в течение суток Ответственный отправляет Конечному пользователю повторный запрос. При отсутствии ответа на повторный запрос по истечении 3 (Трех) дней с момента первого запроса Ответственный переводит Заявку в статус «Закрыта». При получении необходимой информации или обратной связи Конечного пользователя Заявка возвращается в работу.

7. МОНИТОРИНГ И КОНТРОЛЬ КАЧЕСТВА

7.1. Контроль над исполнением обязательств осуществляется специалистами Сервисного центра с помощью АИС.

7.2. АИС обеспечивает своевременную эскалацию выполняемых заявок и запросов по техническому обслуживанию, которая позволяет контролировать сроки решения Заявок, предупреждая нарушение сроков, обозначенных в Программе технической поддержки.

8. ЗАКРЫТИЕ ЗАЯВКИ

8.1. При решении Заявки специалист Исполнителя производит уведомление инициатора Заявки о решении.

8.2. АИС отправляет электронное письмо на подтверждение закрытия Заявки.

8.3. **Закрытие Заявки по подтверждению.** Закрытие Заявок по подтверждению производится специалистами Исполнителя: по факту поступления от Конечного пользователя подтверждения о закрытии Заявки.

8.4. **Автоматическое закрытие Заявки.** В случае если Конечный пользователь после получения уведомления от Исполнителя не выслал подтверждение или не сообщил о неисполнении Заявки в течение 3 (Трех) календарных дней, Заявка закрывается автоматически.

9. ОТЧЕТНОСТЬ

9.1. При способе обработки Заявок: «Выезд» Конечным пользователем и Исполнителем в обязательном порядке оформляется и подписывается акт выполненных работ по Заявке (Приложение № 3). Акт выполненных работ по Заявке подписывается Ответственным и представителем Конечного пользователя.

10. ДОПОЛНИТЕЛЬНЫЕ УСЛОВИЯ

10.1. Конечный пользователь обязан подавать заявки на функциональные модули ПО, которые были исправны на момент подписания Регламента.

10.2. Конечный пользователь обязан соблюдать условия правообладателя в отношении использования ПО, в том числе: наличие действующих лицензий и условия эксплуатации, указанные в техническом описании условий эксплуатации, поставляемом вместе с ПО, а также – на web-сайте производителей.

10.3. Обслуживание не предоставляется по Регламенту, но может быть предоставлено за отдельную плату в рамках отдельного договора, заключаемого с Исполнителем:

- при несоблюдении условий эксплуатации/использования ПО;
- при установлении факта выхода из строя ПО до момента подписания Регламента, в таких случаях предварительное восстановление работоспособности обслуживаемого ПО не предусмотрено условиями Регламента;
- при неработоспособности обслуживаемых компонентов, вызванной работой других, не обслуживаемых Исполнителем компонентов;
- при неработоспособности обслуживаемого ПО вследствие эксплуатации без электрического заземления, а при соединении нескольких устройств – без единого контура заземления;
- при неработоспособности обслуживаемого ПО вследствие механического повреждения оборудования;
- при неработоспособности вследствие внесения изменений в версию программного

обеспечения и/или конфигурацию обслуживаемых систем без уведомления Исполнителя в порядке, установленном пунктом 10.4 Регламента;

- при обращении представителей Конечного пользователя вне утвержденного списка уполномоченных лиц;
- при необходимости доработки ПО, вызванной изменением автоматизируемых процессов.

10.4. При желании Конечного пользователя внести какие-либо изменения в ПО, находящееся на обслуживании Исполнителя, Конечный пользователь обязан производить только изменения, рекомендованные правообладателем, и использовать только программное обеспечение, на которое у Конечного пользователя имеются надлежащим образом оформленные права с уведомлением Исполнителя за 1 (Один) рабочий день до проведения изменений. Изменения, необходимые для работы с новыми версиями других компонентов (не относящихся к ПО), производятся за счет Конечного пользователя.

11. ПРИЛОЖЕНИЯ

- 11.1. Приложение № 1. Программа технической поддержки на 1 л.
- 11.2. Приложение № 2. Форма заявки на 1 л.
- 11.3. Приложение № 3. Форма акта выполненных работ по Заявке на 2 л.

ПРОГРАММА ТЕХНИЧЕСКОЙ ПОДДЕРЖКИ

<p>Объем услуг</p>	<ul style="list-style-type: none"> ● Сервисный Центр ● Консультации ● Восстановление работоспособности ПО ● Внесение изменений в текущую конфигурацию системы ● Профилактическое обслуживание 															
<p>Сервисный центр</p>	<p>Время обслуживания: 24x7</p> <p>Услуга включает в себя:</p> <ul style="list-style-type: none"> ● учет Заявок от Конечного пользователя в АИС; ● информирование Конечного пользователя о регистрации, изменении статуса, закрытии Заявки; ● хранение информации об обслуживаемой инфраструктуре. При внесении изменений в инфраструктуру информация в системе изменяется, что позволяет специалистам оперативно получать доступ к информации о текущем состоянии ПО, актуальных установленных версиях продуктов. 															
<p>Консультации</p>	<p>Время обслуживания: 12x7</p> <p>Время реакции: в течение 4 (четырёх) часов</p> <p>Способы обработки заявок: дистанционно</p> <p>Услуга предоставляет возможность получения консультаций специалистов по различным техническим вопросам, связанным с эксплуатацией, конфигурированием, управлением <i>исправным</i> ПО. Консультации предоставляются по вопросам, ответ на который отсутствует в переданной Конечному пользователю эксплуатационной документации.</p>															
<p>Восстановление работоспособности ПО</p>	<p>Время обслуживания: 12x7</p> <table border="1" data-bbox="558 1176 1548 1691"> <thead> <tr> <th>Приоритет SLA</th> <th>Критичный</th> <th>Высокий</th> <th>Средний</th> <th>Низкий</th> </tr> </thead> <tbody> <tr> <td>Время реакции</td> <td>2 (Два) часа</td> <td>4 (Четыре) часа</td> <td>8 (Восемь) часов</td> <td>8 (Восемь) часов</td> </tr> <tr> <td>Время исправления</td> <td>4 (Четыре) часа</td> <td>8 (Восемь) часов</td> <td>5 (Пять) дней</td> <td>30 (Тридцать) дней</td> </tr> </tbody> </table> <p>Способы обработки заявок: дистанционно, удаленно</p> <p>Услуга включает в себя:</p> <ul style="list-style-type: none"> ● обновление программного обеспечения; ● содействие Конечному пользователю при проведении изменений в конфигурации компонентов системы, а также анализ негативного воздействия планируемых изменений. 	Приоритет SLA	Критичный	Высокий	Средний	Низкий	Время реакции	2 (Два) часа	4 (Четыре) часа	8 (Восемь) часов	8 (Восемь) часов	Время исправления	4 (Четыре) часа	8 (Восемь) часов	5 (Пять) дней	30 (Тридцать) дней
Приоритет SLA	Критичный	Высокий	Средний	Низкий												
Время реакции	2 (Два) часа	4 (Четыре) часа	8 (Восемь) часов	8 (Восемь) часов												
Время исправления	4 (Четыре) часа	8 (Восемь) часов	5 (Пять) дней	30 (Тридцать) дней												

Форма Заявки

Номер договора	
Наименование компании Конечный пользователь	
Обратившийся	<i>ФИО, телефон, электронная почта.</i>
Суть Заявки	<i>Не более 2-3 предложений, выражающих суть вопроса или проблемы.</i>
Описание Заявки	<p><i>Дополнительная информация, которая помогает раскрыть суть проблемы или вопроса (например, текст ошибки, которую видит пользователь, или снимок экрана). Рекомендуется также предоставить следующую информацию:</i></p> <ul style="list-style-type: none"> ● <i>дата обнаружения неисправности;</i> ● <i>периодичность появления неисправности;</i> ● <i>предпринятые действия;</i> ● <i>если программное обеспечение находится в эксплуатации, когда возможна остановка для проведения работ.</i>
Программное обеспечение	<p><i>Детальная информация:</i></p> <ul style="list-style-type: none"> ● <i>о программном обеспечении: наименование, номер версии.</i>
Приоритет	<ul style="list-style-type: none"> ● Критичный приоритет – программное обеспечение не работает, что приводит к остановке всех или одной из обслуживаемых систем. ● Высокий приоритет – частично нарушена работоспособность, либо значительно снижена отказоустойчивость или производительность программного обеспечения Конечного пользователя, но при этом создается критическая ситуация, при которой полная потеря работоспособности всего технологического процесса Конечного пользователя может произойти в любой момент. ● Средний приоритет – незначительно нарушена работоспособность, либо производительность программного обеспечения, что не оказывает влияния на бизнес Конечного пользователя. ● Низкий приоритет – ошибка, которая не существенно ограничивает использование программного обеспечения для выполнения необходимых функций, либо требуется консультация специалиста по техническим вопросам, связанным с эксплуатацией, конфигурированием, управлением исправным программным обеспечением, поиском причины возникновения неисправности.

Форма акта выполненных работ по Заявке №

г. _____

« ____ » _____ 202__ г.

**КОНЕЧНЫЙ
ПОЛЬЗОВАТЕЛЬ:**

ИСПОЛНИТЕЛЬ:

Настоящий Акт составлен представителем Исполнителя:

и представителем Конечного пользователя:

в том, что Исполнителем выполнены работы на следующем программном обеспечении:

Выявлены следующие неисправности:

Выполнены следующие работы:

Выданы следующие рекомендации:

Специалисты, выполняющие работы:

Дата и время выполнения работ: с _____ по _____

Длительность: _____ ч _____ мин. **Фактическая трудоемкость работ (чел.ч.):** _____

Замечания Исполнителя:

Замечания Конечного пользователя и оценка качества работ:

Стороны подтверждают достоверность приведенной в Акте информации по объему, длительности, трудоемкости работ, дополнительных замечаний не имеют.

Подписи Сторон

от Исполнителя:

_____/_____/

«__» _____ 202__ г.

от Конечного пользователя:

_____/_____/

«__» _____ 202__ г.