

УТВЕРЖДЕН

RU.1107746574308.078-01 98 01-ЛУ

**Платформа разработки программного обеспечения БЗ.
Модуль Комплексная экологическая платформа (КЭП БЗ)**

Описание жизненного цикла

RU.1107746574308.078-01 98 01

Листов 29

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

2025

АННОТАЦИЯ

Настоящий документ содержит описание жизненного цикла (далее – ЖЦ) программного обеспечения «Платформа разработки программного обеспечения БЗ. Модуль Комплексная экологическая платформа (КЭП БЗ)» RU.1107746574308.078-01 (далее – Модуль) и содержит сведения о процессах разработки и устранения неисправностей, выявленных в ходе эксплуатации программного обеспечения, сведения о совершенствовании программного обеспечения, а также информацию о персонале, необходимом для обеспечения его разработки, совершенствования и поддержки.

СОДЕРЖАНИЕ

1. Общие сведения	5
1.1. Наименование программы	5
1.2. Назначение программы	5
1.3. Особенности применения	6
2. Жизненный цикл	7
2.1. Стадии и процессы жизненного цикла	7
2.2. Стадия «Разработка»	10
2.2.1. Общие сведения.....	10
2.2.2. Процессы разработки.....	13
2.2.3. Методы и средства	16
2.2.4. Персонал.....	18
2.3. Стадия «Производство»	18
2.4. Стадия «Поддержка» (сопровождение)	20
2.4.1. Общие сведения.....	20
2.4.2. Процессы поддержки	20
2.4.3. Методы и средства	27
2.4.4. Персонал.....	29

ПЕРЕЧЕНЬ ТЕРМИНОВ, СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

Используемые в настоящем документе термины и основные понятия области автоматизированных систем определены в ГОСТ Р 59853-2021 (таблица 1).

Таблица 1 – Термины и основные понятия

Обозначение	Описание
API	от англ. Application Programming Interface – программный интерфейс взаимодействия
АС	Автоматизированная система
ВМ	Виртуальная машина
ГОСТ	Государственный стандарт
ЖЦ	Жизненный цикл
КЭП БЗ, Модуль	Модуль Комплексная экологическая платформа на базе платформы БЗ
ОС	Операционная система
ПТО УОНВОС	Программно-техническое обеспечение учета объектов, оказывающих негативное воздействие на окружающую среду
СУБД	Система управления базами данных

1. ОБЩИЕ СВЕДЕНИЯ

1.1. Наименование программы

Полное наименование программы: Платформа разработки программного обеспечения БЗ. Модуль Комплексная экологическая платформа (КЭП БЗ) RU.1107746574308.078-01.

Сокращенное наименование программы: Модуль КЭП БЗ. В рамках настоящего документа употребляется также термин «Модуль».

Обозначение программы: RU.1107746574308.078-01.

КЭП БЗ RU.1107746574308.078-01 – это российское программное обеспечение, организация-разработчик: Общество с ограниченной ответственностью «Большая Тройка» (ООО «Большая Тройка»).

Сайт организации-разработчика: <https://big3.ru>.

Организация-правообладатель: Общество с ограниченной ответственностью «Большая Тройка» (ООО «Большая Тройка»).

1.2. Назначение программы

Модуль предназначен для объединения информации из всех систем автоматического контроля, автоматизации сбора, анализа и передачи данных о выбросах/сбросах на предприятиях, в рамках организации и её зависимых компаний. Программа позволяет передавать и получать данные о выбросах/сбросах в программно-техническое обеспечение учета объектов, оказывающих негативное воздействие на окружающую среду (ПТО УОНВОС).

Модуль может быть установлен в инфраструктуре пользователя или использоваться путем дистанционного доступа, имеет web-интерфейс настройки визуализации и работы с данными, подключается в качестве js-компонента, а также содержит API для получения данных от программных средств измерения выбросов/сбросов, обмена данными с ПТО УОНВОС.

Программа имеет возможность настроек способов представления данных, хранения, преобразования, фильтрации данных из SQL-хранилищ.

Модуль является отдельным функциональным компонентом платформы БЗ. Он поддерживает плотную интеграцию со всеми модулями, реализованными на базе платформы БЗ, но может быть использован независимо.

1.3. Особенности применения

Модуль включает в себя следующие компоненты, представляющие собой разделы пользовательского интерфейса Модуля:

- Главная страница;
- Профиль и настройка;
- Заявка на добавление организации;
- Мои объекты;
- Мои связанные компании;
- Мои системы контроля;
- Счёт (платежи);
- Уведомления.

Модуль обеспечивает взаимодействие с пользователем по технологии «тонкий клиент» с применением распространенных веб-браузеров (Mozilla Firefox, Microsoft Edge, Google Chrome, Safari, «Яндекс.Браузер»).

2. ЖИЗНЕННЫЙ ЦИКЛ

2.1. Стадии и процессы жизненного цикла

Согласно ГОСТ Р 57193-2016 «Системная и программная инженерия. Процессы жизненного цикла систем» стадия – это период в пределах жизненного цикла программного обеспечения, который относится к состоянию его описания или реализации. При этом стадии относятся к основному развитию и достижению контрольных точек в течение жизненного цикла программного обеспечения.

Согласно ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств» процесс жизни любой системы или программного продукта может быть описан посредством модели жизненного цикла, состоящей из стадий. При этом процесс – это совокупность взаимосвязанных или взаимодействующих видов деятельности, преобразующих входы и выходы.

С учетом ГОСТ Р 57193-2016 «Системная и программная инженерия. Процессы жизненного цикла систем» полный жизненный цикл Модуля включает следующие стадии жизненного цикла программного обеспечения:

- замысел;
- разработка;
- приобретение и поставка;
- производство;
- эксплуатация;
- поддержка;
- снятие с эксплуатации (списание).

Каждая стадия жизненного цикла Модуля предполагает реализацию определенных процессов, а также применения соответствующих методов и средств их реализации.

Описание каждой стадии включает в себя:

- перечень процессов, реализуемых на данной стадии;
- описание применяемых методов;
- описание применяемых средств;
- описание требований к персоналу, необходимому для выполнения соответствующих процессов.

В настоящем документе описаны процессы, методы и средства реализации этих процессов (включая требования к персоналу), связанные со следующими стадиями жизненного цикла Модуля:

- «разработка»;
- «производство»;
- «поддержка».

Указанные стадии жизненного цикла Модуля направлены на достижение следующих целей:

- удовлетворение пользователей (организаций-пользователей, конечных пользователей);
- улучшение качества программного обеспечения;
- устранение проблем, выявленных в ходе эксплуатации программного обеспечения;
- расширение функциональности программного обеспечения.

Стадии «замысел», «приобретение и поставка» и «списание» в настоящем документе не рассматриваются, поскольку реализуются единовременно (с учетом особенностей Модуля).

Стадия «эксплуатация» в настоящем документе также не рассматривается отдельно, поскольку эксплуатация выполняется пользователями, которые одновременно являются активными участниками процессов стадии «поддержка».

Описание представлено с учетом требований ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств», а также в соответствии с принятыми в организации-разработчике политиками и процедурами в отношении процессов ЖЦ разрабатываемого программного обеспечения.

Каждый процесс описывается в терминах следующего набора ключевых атрибутов:

- наименование процесса (передает область применения процесса как целого);
- цели (описывает конечные цели выполнения процесса);
- выходы (представляют собой наблюдаемые результаты, ожидаемые при успешном выполнении процесса);
- деятельность (является перечнем действий, используемых для достижения выходов);

- задачи (требования, рекомендации и допустимые действия, предназначенные для поддержки достижения выходов процесса).

На протяжении всего ЖЦ Модуля стадии могут повторяться и осуществляться параллельно с учетом результатов выполнения процессов на каждой из них.

Процессы жизненного цикла Модуля обеспечиваются участием множества специалистов организации-разработчика, включающего следующие роли сотрудников:

- менеджер проекта;
- эколог-аналитик;
- технический писатель,
- дизайнер интерфейсов;
- руководитель группы разработки;
- программист;
- руководитель группы тестирования;
- инженер по тестированию;
- руководитель группы технической поддержки;
- инженер службы технической поддержки.

Высокий уровень качества программного обеспечения достигается за счет следующих аспектов:

- применение проверенных методик, формализация процессов разработки и управления требованиями (в том числе применение механизмов «user story» и «use cases»);
- автоматизация процессов разработки, управления версиями, тестирования и сборки путем применения специализированного программного обеспечения;
- применение практик непрерывной интеграции (Continuous Integration, CI) и непрерывной поставки (Continuous Delivery, CD);
- тестирование на всех этапах выпуска программного обеспечения («code review», unit-тесты, ручное и автоматизированное тестирование сборок);
- документирование ключевых операций и знаний в базе знаний организации-разработчика;
- эффективная работа службы технической поддержки в рамках взаимодействия с организацией-пользователем Модуля.

Основными инструментами автоматизации для поддержания жизненного цикла Модуля являются:

- система управления задачами (Yandex Tracker);

- система контроля версий (Git);
- средство автоматизированной сборки (Gitlab CI);
- система ведения базы знаний (Yandex Wiki).

Система управления задачами предназначена для:

- организации совместной работы сотрудников организации-разработчика над задачами разработки;
- управления задачами по добавлению новой функциональности и исправлению ошибок;
- мониторинга показателей развития продукта.

Система контроля версий предназначена для:

- обеспечения эффективной совместной работы специалистов группы разработки и группы тестирования программного обеспечения;
- обеспечения возможности одновременного ведения нескольких ветвей разработки, версий промежуточных состояний и релизов.

Средство автоматизированной сборки обеспечивает сборку версий Модуля в заданном окружении по требованию или по расписанию.

Система ведения базы знаний обеспечивает возможность накопления и использования ранее полученных знаний для более эффективного решения возникающих проблем при эксплуатации Модуля, а также для определения приоритетных направлений развития функциональности Модуля. База знаний также используется для формализации контрольных процедур и внутренних политик, обеспечивающих системность, повторяемость и качество процессов разработки и сопровождения Модуля.

2.2. Стадия «Разработка»

2.2.1. Общие сведения

Разработка Модуля (размещение инфраструктуры разработки и размещение разработчиков) осуществляется по адресу местонахождения организации-разработчика.

В рамках стадии «Разработка» выполняется разработка и совершенствование Модуля, в том числе добавление новой функциональности, подключение дополнительных сервисов/компонентов, выпуск версий.

Каждая существенная доработка Модуля выполняется как самостоятельный проект в соответствии с формализованной методикой. Этапы разработки и

совершенствования Модуля выполняются с учетом требований ГОСТ 19.102–77 «Единая система программной документации. Стадии разработки» и ГОСТ 34.601–90 «Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания», в которых определены стадии создания и разработки, а также соответствующие им этапы и содержание работ для программного обеспечения и автоматизированных систем (см. таблицу 2).

Таблица 2 – Этапы создания программного обеспечения

Стадии создания (по ГОСТ 34.601)	Стадии разработки (по ГОСТ 19.102)	Этапы работ (по ГОСТ 19.102 и ГОСТ 34.601)	Содержание работ (по ГОСТ 19.102)
1. Формирование требований к АС	1. Техническое задание	Обоснование необходимости разработки программы. Формирование требований пользователя	Постановка задачи. Сбор исходных материалов
2. Разработка концепции АС		Разработка вариантов концепции АС, удовлетворяющей требованиям пользователя	Определение структуры входных и выходных данных. Предварительный выбор методов решения задач
3. Техническое задание		Разработка и утверждение технического задания	Определение требований к техническим средствам. Определение требований к программе. Определение стадий, этапов и сроков разработки программы и документации на нее
4. Эскизный проект	2. Эскизный проект	Разработка предварительных проектных решений. Утверждение эскизного проекта	Предварительная разработка структуры входных и выходных данных. Уточнение методов решения задачи. Разработка общего описания алгоритма решения задачи

Стадии создания (по ГОСТ 34.601)	Стадии разработки (по ГОСТ 19.102)	Этапы работ (по ГОСТ 19.102 и ГОСТ 34.601)	Содержание работ (по ГОСТ 19.102)
5. Технический проект	3. Технический проект	Разработка проектных решений по Модулю и его частям. Разработка документации на Модуль и его части. Утверждение технического проекта	Уточнение структуры входных и выходных данных. Разработка алгоритма решения задачи. Определение формы представления входных и выходных данных. Разработка структуры программы. Окончательное определение конфигурации технических средств
6.Рабочая документация	4. Рабочий проект	Разработка программной документации. Разработка рабочей документации на Модуль и его части. Испытания программы	Программирование и отладка программы. Разработка программных документов в соответствии с требованиями ГОСТ 19.101. Разработка, согласование и утверждение программы и методики испытаний. Проведение испытаний. Корректировка программы и программной документации по результатам испытаний
6.Ввод в действие	5. Внедрение	Подготовка и передача программы на производство. Подготовка персонала. Проведение опытной эксплуатации	Подготовка и передача программы и программной документации для сопровождения и (или) изготовления. Оформление и утверждение акта о передаче программы на сопровождение и (или) изготовление

2.2.2. Процессы разработки

Согласно требованиям ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств» на стадии разработки Модуля реализуются следующие процессы:

– **Определение и анализ требований.** Процесс реализуется менеджером продукта при участии программистов, инженеров по тестированию и инженеров службы технической поддержки. Процесс координируется менеджером проекта.

– **Проектирование и конструирование** (включая проектирование архитектуры и детальное проектирование). Процесс реализуется руководителем группы разработки при участии программистов. Процесс координируется менеджером продукта.

– **Программирование и комплексирование.** Процесс реализуется программистами при участии руководителя группы разработки с применением средств автоматизированной сборки. Процесс координируется менеджером продукта.

– **Тестирование.** Процесс реализуется инженерами по тестированию и инженерами службы технической поддержки. Процесс координируется руководителем группы тестирования.

На стадии «Разработка» реализуются виды деятельности и решаются задачи в соответствии с принятыми в организации политиками и регламентами в отношении процессов разработки программного обеспечения, а также с учетом требований ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств».

Описание процессов, реализуемых на стадии «Разработка» приведено в таблице 3.

Таблица 3 – Описание процессов стадии «Разработка» для Модуля

Процесс «Определение и анализ требований»	
Цели	Выявление требований, которые должны быть учтены при разработке (совершенствовании) Модуля. Формализация совокупности технических требований (функциональных и нефункциональных), которые должны быть учтены (реализованы) при разработке (совершенствовании) Модуля
Выходы	Определена совокупность функциональных и нефункциональных требований, обеспечивающих удовлетворение потребностей пользователей. Требования внесены в систему управления задачами (Yandex Tracker). Сформированы описания «user story» и «use cases» для групп требований. Определена согласованная приоритетность реализации требований

Участники	Менеджер проекта, аналитик, руководитель группы разработки
Применяемые средства автоматизации	Yandex Tracker (учет и отслеживание согласования/исполнения требований, описаний «user story» и «use cases»)
Координатор	Менеджер проекта
Деятельность	Спецификация требований
Задачи	Сбор требований от пользователей. Анализ накопленных сведений, получаемых от службы технической поддержки. Анализ планируемого применения Модуля и планируемых траекторий развития продукта и рынка в целом. Идентификация, оценка и регистрация требований. Определение требуемых характеристик и условий использования Модуля. Определение ограничений для системных решений. Формирование основы для ведения переговоров с заказчиками (организациями-пользователями) и заключения контрактов. Разработка спецификации требований, которая описывает: а) функции и возможности Модуля; б) требования деловой среды; в) требования по безопасности, защищенности, эргономике, интерфейсам, рабочим операциям и сопровождению; г) проектные ограничения и квалификационные требования. Анализ требований: а) анализ системных требований на предмет реализуемости, корректности и тестируемости; б) анализ воздействия системных требований на среду применения; в) упорядочение требований (требования расставляются по приоритетам, утверждаются и обновляются); г) оценка затрат, рисков, приоритетности реализации
Процесс «Проектирование архитектуры»	
Цели	Определение особенностей и общих механизмов реализации требований в составе Модуля. Разработка программно-технической архитектуры решения, позволяющей реализовать необходимые требования
Выходы	Определены архитектурно-технические решения для разработки (совершенствования). Разработаны оперативные постановки задач (частные технические задания) на разработку отдельных функций и компонентов Модуля. Требования распределены по компонентам Модуля. Определены интерфейсы взаимодействия (внутренние и внешние)
Участники	Руководитель группы разработки, программист, менеджер проекта, технический писатель

Применяемые средства автоматизации	Yandex Wiki (учет и отслеживание частных технических заданий, описаний архитектуры и т.п.)
Координатор	Менеджер проекта
Деятельность	Проектирование, документирование и согласование архитектурно-технических решений, которые необходимо реализовать
Задачи	Разработка архитектурного проекта, в соответствии с которым выполняется идентификация верхнего уровня архитектуры и элементов Модуля и удовлетворяются заданные требования. Распределение требований по элементам Модуля. Определение внутренних и внешних интерфейсов. Верификация между системными требованиями и архитектурой Модуля. Документирование и согласование архитектурных решений. Разработка частных постановок задач
Процесс «Программирование и комплексирование»	
Цели	Реализация и сборка Модуля (новой версии Модуля, в случае совершенствования)
Выходы	Функционирующая версия Модуля, готовая к прохождению процедур внутреннего и внешнего тестирования
Участники	Программист, дизайнер интерфейсов, руководитель группы разработки, менеджер проекта
Применяемые средства автоматизации	Yandex Tracker (учет и отслеживание реализации требований и задач). Yandex Wiki (база знаний). Система контроля версий Git. Средство автоматизированной сборки Gitlab CI
Координатор	Менеджер проекта
Деятельность	Реализация требований в соответствии с постановками задач
Задачи	Реализация требований в соответствии с задачами из Yandex Tracker (разработка отдельных компонентов и загрузка их в систему контроля версий). Проведение процедур «code review», выполнение unit-тестов. Слияние ветвей проекта (при необходимости). Разработка конфигурации сборки. Автоматизированная сборка Модуля. Верификация доработанной версии Модуля. Проведение регрессионных тестов. Разработка сценариев тестирования реализованных требований (включая тестовые примеры, входные и выходные данные и т.п.). Разработка документации и материалов для базы знаний
Процесс «Тестирование»	
Цели	Обеспечение готовности версии Модуля к релизу

Выходы	Функционирующая версия Модуля, готовая к релизу
Участники	Инженер по тестированию
Применяемые средства автоматизации	Yandex Tracker (учет и отслеживание реализации требований и задач). Yandex Wiki (база знаний)
Координатор	Руководитель группы тестирования
Деятельность	Проверка функционирования Модуля по контрольным спискам (чек-листам). Проверка покрытия требований
Задачи	Проверка функционирования Модуля по контрольным спискам (чек-листам). Протоколирование результатов тестирования. Фиксация выявленных ошибок в Yandex Tracker. Классификация выявленных ошибок и выработка решений по их устранению (при необходимости – проведение повторного тестированию)

На выходе стадии «Разработка» Модуль представляет собой набор Docker-образов, который поступает на вход стадии «Производство» для развертывания и запуска в кластере посредством Docker Swarm.

2.2.3. Методы и средства

Разработка и описание требований к Модулю ведется с применением методов «user story» и «use cases», с использованием инструментов Yandex Tracker.

Модуль разработан с использованием следующих инструментов и технологий:

- языки программирования:
 - серверная часть: Python, TypeScript
 - веб-интерфейс: JavaScript.
- фреймворки:
 - серверная часть: Django,
 - веб-интерфейс: Angular.
- СУБД: свободная объектно-реляционная СУБД PostgreSQL;
- сервис брокера сообщений: RabbitMQ;
- сервис фоновых процедур: Celery;
- сервис регламентных заданий: Cron;
- сервис WebSocket;
- сервис полнотекстового поиска: Manticore Search;
- балансировщик и маршрутизатор запросов: Traefik;

- контейнер сервисов: Docker;
- хостовая ОС: Linux Debian.

Сборка осуществляется в автоматическом режиме с использованием подготовленных Docker-файлов.

Подробная актуальная информация о сборке и подготовке Модуля к релизу размещается во внутренней базе знаний организации-разработчика. В том числе там содержится актуальная информация по следующим разделам:

- как собрать Модуль;
- как подготовить сборку;
- порядок работы с версиями;
- как узнать версию сборки;
- работа с кодом и подготовка новых версий.

После сборки и установки Модуля проводится тестирование (автоматизированное и ручное).

В рамках автоматизированного тестирования разрабатываются unit-тесты и интеграционные тесты всех компонентов Модуля. Автоматизированное тестирование выполняется в GitLab CI.

В рамках ручного тестирования осуществляется подход ко всем сценариям, доступным пользователям.

Проверки осуществляются в следующем объеме (конкретные методы проверок содержатся в базе знаний организации-разработчика):

- совместимость Модуля с программным окружением;
- проверка выполнения функции входа в Модуль (аутентификация);
- проверка орфографии в текстовых элементах экранных форм и меню;
- проверка отклика пользовательского интерфейса;
- проверка переключения по разделам в пользовательском интерфейсе;
- набор стандартных контрольных проверок:
 - проверка длительности выполнения операции по открытию форм Модуля;
 - проверка длительности выполнения поисковой операции в компонентах Модуля;
 - проверка длительности появления первого отклика Модуля с момента постановки задачи на выполнение работы.

Результаты тестирования версии (сборки) Модуля для каждой проверяемой инсталляции документируются в протоколах тестирования и размещаются в соответствующих разделах базы знаний организации-разработчика.

2.2.4. Персонал

В ходе реализации процессов ЖЦ стадии «Разработка» участвует персонал, состав, роли и требования к квалификации которого указаны в таблице 4.

Таблица 4 – Сведения о персонале, участвующем в процессах стадии «Разработка»

Роль	Кол-во, чел.	Квалификация
Менеджер проекта	1	Релевантный опыт руководства проектами, знания в области тенденций развития рынка современных информационных технологий. Опыт работы на аналогичной позиции не менее 3 лет
Аналитик	1	Релевантный опыт работы. Знания в области аналитики. Навыки прогнозирования и стратегического планирования. Опыт работы на аналогичной позиции не менее 3 лет
Технический писатель	1	Релевантный опыт работы. Знания в области разработки и ведения документации ПО. Опыт работы на аналогичной позиции не менее 3 лет
Дизайнер интерфейсов	1	Релевантный опыт работы дизайнером пользовательских интерфейсов, владение соответствующими инструментами (в том числе Figma).
Руководитель группы разработки	1	Опыт разработки и веб-приложений. Релевантный опыт руководства коллективом разработчиков веб-приложений
Руководитель группы тестирования	1	Опыт тестирования веб-приложений. Релевантный опыт руководства коллективом инженеров по тестированию
Программист	8	Релевантный опыт разработки веб-приложений (серверная и клиентская часть. Обязательно: знания языков разработки Python, TypeScript, JavaScript
Инженер по тестированию	1	Релевантный опыт тестирования и разработки веб-приложений

2.3. Стадия «Производство»

В рамках стадии «Производство» осуществляется реализация установки и настройки релиза Модуля.

Модуль устанавливается на вычислительных мощностях виртуальных машин, требования к необходимым характеристикам которых представлены в таблице 5.

Таблица 5 – Необходимые характеристики в ВМ

Наименование параметра, единицы измерения	Значение	
	ВМ Приложений	ВМ БД Master
Частота процессора, ГГц	не ниже 2,1	не ниже 2,1
Количество ядер (процессоров)	16	16
Количество оперативной памяти, ГБ	64	64
Объем диска, ГБ	60	400
Канал связи, Гбит/сек:		
для локальной сети	не ниже 10	не ниже 10
для сети Интернет	не ниже 1	-

Необходимые характеристики дискового пространства ВМ представлены в таблице 6.

Таблица 6 – Необходимые характеристики дискового пространства ВМ

Дисковое пространство в разрезах:	Единицы измерения	Значение	
		ВМ Приложений	ВМ БД Master
ОС и системное ПО	ГБ	30	30
Дистрибутивы и исходные коды	ГБ	10	20
Логи	ГБ	20	300
Итого:	ГБ	60	350

Обобщенный перечень проверок приведен в п. 2.2.3.

Выявленные в ходе проверки ошибки классифицируются по следующим типам:

– критические – ошибки, которые не позволяют использовать основную функциональность Модуля;

– некритические – ошибки, которые не мешают использовать основную функциональность Модуля, проявляются редко и на общую работоспособность влияют незначительно.

К типовым критическим ошибкам относятся следующие ошибки:

– ошибки, блокирующие действия пользователя и мешающие пользоваться Модулем;

– нарушения дизайна или функционирования пользовательского интерфейса.

Устранение выявленных проблем выполняется сотрудниками организации-разработчика в рамках выполнения технической поддержки процессов производства.

2.4. Стадия «Поддержка» (сопровождение)

2.4.1. Общие сведения

Поддержка (сопровождение) Модуля осуществляется по адресу местонахождения организации-разработчика.

Поддержка (сопровождение) программного обеспечения – процесс улучшения, оптимизации и устранения дефектов программного обеспечения после передачи в эксплуатацию. В ходе сопровождения в Модуль вносятся изменения с тем, чтобы исправить обнаруженные в процессе использования дефекты и недоработки, а также для добавления новой функциональности с целью повышения удобства использования и применимости Модуля.

Сопровождение программного обеспечения производится с учетом требований ГОСТ Р ИСО/МЭК 14764-2002 «Информационная технология. Сопровождение программных средств».

Процесс сопровождения состоит из работ и задач, реализуемых персоналом, задействованного при реализации процессов стадии «Разработка».

2.4.2. Процессы поддержки

Согласно требованиям ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств» на стадии поддержки Модуля реализуются следующие процессы:

- менеджмент документации;
- менеджмент конфигурации;
- верификация;
- устранение неисправностей;
- совершенствование;
- техническая поддержка пользователей.

2.4.2.1. Процесс менеджмента документации

Цель процесса менеджмента документации заключается в разработке и сопровождении (поддержании в актуальном состоянии) документации на Модуль.

Основными задачами процесса менеджмента документации являются:

- разработка стратегии идентификации документации, которая реализуется в течение жизненного цикла;

- определение стандартов, которые применяются при разработке программной документации на программное обеспечение;
- определение документации, которая разрабатывается в рамках определенного процесса;
- определение и утверждение содержания и целей документации;
- разработка документации и организация доступа к ней;
- сопровождение документации (поддержание актуальности, устранение ошибок).

Стратегия менеджмента документации оформляется в виде плана, определяющего документы, которые разрабатываются в течение жизненного цикла Модуля. При планировании разработки документации определяется сводный перечень разрабатываемых документов, который по каждому документу содержит:

- наименование;
- цели и содержание;
- круг пользователей, которым документ предназначен;
- процедуры и ответственность при формировании исходных данных, разработке, ревизиях, модификации, утверждении, производстве, хранении, распространении;
- соответствие графику выпуска релизов Модуля (соответствие версии Модуля).

Документированием сопровождаются все процессы жизненного цикла Модуля. Основные применяемые пути документирования:

- разработка программной документации на Модуль;
- разработка инструкций для пользователей (руководство оператора Модуля, инструкция по установке и проверке функционирования Модуля);
- формирование и размещение материалов (информационных статей) в базе знаний предприятия-разработчика;
- формирование списка изменений в версиях Модуля;
- разработка внутренней рабочей документации (описания технических решений, частные технические задания и постановки задач на разработку Модуля в Yandex Wiki).

Подготовленные документы рассматриваются и редактируются по формату, техническому содержанию и стилю представления в соответствии со стандартами на документацию.

Документы изготавливаются и поставляются в соответствии с планом. При производстве и распределении документов может использоваться бумажный или

электронный вид носителя. Важные материалы хранятся в соответствии с требованиями по содержанию записей, защищенности, сопровождению и резервированию.

Изменения в документацию вносятся при выполнении процессов сопровождения Модуля.

2.4.2.2. Процесс менеджмента конфигурации

Цель процесса менеджмента конфигурации заключается в установлении и сопровождении целостности составных частей Модуля и обеспечении их доступности для заинтересованных сторон.

Основными задачами процесса менеджмента конфигурации являются:

- разработка плана менеджмента конфигурации программного обеспечения;
- контроль модификаций и выпусков составных частей Модуля;
- обеспечение доступности модификаций и выпусков (релиз-версий Модуля) для соответствующих заинтересованных сторон;
- регистрация и предоставление (по запросу) информации о статусе составных частей и модификаций Модуля;
- обеспечение завершенности и согласованности составных частей;
- контроль хранения, обработки и поставки составных частей.

План менеджмента конфигурации программных средств описывает:

- действия менеджмента конфигурации;
- процедуры и графики работ для выполнения этих действий.

В рамках плана устанавливается схема идентификации программных составных частей, а их версии контролируются в рамках проекта. Для каждой программной составной части и ее версий определяются документация, ссылки на версии и другие детали идентификации.

На основании плана осуществляется управление конфигурацией, которое включает в себя:

- идентификацию и регистрацию заявок на изменения;
- анализ и оценку изменений;
- принятие или отклонение заявок;
- реализацию, верификацию и выпуск модифицированной составной части;
- проверочные испытания, на основании которых можно проследивать каждую модификацию, ее причины и полномочия на проведение изменений;

– управление и аудит доступа к контролируемым программным составным частям, связанным с выполнением критических функций по безопасности или защите.

Для регистрации и отслеживания заявок на изменения в организации-разработчике применяется система Yandex Tracker. Управление составными частями кода, версиями и ветвями проекта осуществляется руководителем группы разработки программного обеспечения с применением системы контроля версий Git и методик управляемых слияний ветвей, а также действующих регламентов работы с репозиторием, принятых в организации-разработчике.

Важные копии исходных кодов, конфигураций сборки и документации поддерживаются в течение всего срока жизни Модуля. Исходный код и документация обрабатываются и хранятся в соответствии с действующими регламентами организации-разработчика, обеспечивающими наличие резервных копий и возможность восстановления в случае сбоев.

2.4.2.3. Процесс верификации

Цель процесса верификации заключается в подтверждении того, что Модуль соответствует заданным требованиям.

Основными задачами процесса верификации являются:

- разработка и осуществление стратегии верификации;
- определение критериев верификации;
- выполнение требуемых действий по верификации;
- определение и регистрация дефектов;
- предоставление результатов верификации заинтересованным сторонам.

Виды деятельности и задачи верификации, включая соответствующие методы, технические приемы и инструментарий для выполнения задач, выбираются в зависимости от конечных целей действий в течение жизненного цикла Модуля.

План верификации содержит действия по верификации, необходимые задачи по верификации для каждого действия, связанные с ними ресурсы, ответственность и графики проведения работ. Процедуры верификации определяются действующими регламентами, включающими проверку Модуля по контрольным спискам (чек-листам) после сборки новой версии.

2.4.2.4. Процесс устранения неисправностей

Цель процесса устранения неисправностей (решения проблем) заключается в обеспечении гарантии того, что все выявленные проблемы в работе Модуля идентифицируются, анализируются, контролируются и подвергаются менеджменту для осуществления их решения.

Основными задачами процесса являются:

- разработка стратегии менеджмента проблем;
- регистрация, идентификация и классификация проблем;
- анализ и оценка проблем для определения приемлемого решения (решений);
- решение проблем;
- отслеживание проблем вплоть до их закрытия (завершения решения).

Процесс устранения неисправностей (решения проблем) является циклическим. Обнаруженные в других процессах проблемы вводятся в процесс устранения неисправностей в качестве исходных данных. Каждая проблема классифицируется по категории и приоритету для облегчения анализа тенденций и эффективного использования ресурсов при решении проблем.

По выявленным проблемам инициируются необходимые действия в соответствии с категорией и приоритетом. Заинтересованные стороны информируются о существовании проблем и реализуемых мероприятиях по их решению (выполняется при необходимости, посредством коммуникаций через службу технической поддержки организации-пользователя и/или организации-разработчика).

По накопленным сведениям о проблемах проводится регулярный (ежеквартально) анализ тенденций. Устанавливаются и анализируются причины проблем, которые далее, если возможно, устраняются. Состояние по каждой зарегистрированной проблеме отслеживается и отражается в регулярных внутренних отчетах.

Процесс устранения неисправностей начинается в одном из следующих случаев:

- обнаружена ошибка в процессе тестирования Модуля;
- поступило обращение конечного пользователя Модуля.

Во всех указанных случаях обращение регистрируется в системе Yandex Tracker, в которой далее проходит процедуры классификации, определения приоритетов и отслеживания исполнения.

Обработка обращений выполняется в зависимости от их типа и приоритета, с учетом текущих планов работ. Типы задач, создаваемых в системе организации-разработчика:

- ошибка;
- разработка (реализация новой функциональности);
- поддержка (решение проблем с существующей функциональностью);
- дизайн;
- требования;
- обращение;
- интеграция.

В случае обнаружения ошибки на этапе тестирования, если тестирование проводилось в рамках работы над уже существующей задачей в системе управления проектами и задачами, информация об ошибке и условиях ее воспроизведения указывается в комментариях к этой задаче. В остальных случаях в системе Yandex Tracker создается новая задача с описанием ошибки и условий ее воспроизведения. Если ошибка возникла в процессе работы над существующей задачей, разработчик (программист) сразу приступает к ее устранению. В случае создания новой задачи, в зависимости от критичности выявленной ошибки, руководителем группы разработки определяется, когда задача должна быть взята в работу.

С обращениями конечных пользователей работают инженеры службы технической поддержки.

Если проблема или ошибка требует вмешательства разработчиков, то инженер службы технической поддержки описывает проблему или ошибку по исходным данным, полученным от конечных пользователей. В системе Yandex Tracker создается задача, которая далее передается программистам в группу разработки и тестирования. Затем проблема анализируется программистами, после чего они вносят необходимые изменения в исходный код Модуля и (совместно с инженерами по тестированию) проводят верификацию изменений.

2.4.2.5. Процесс совершенствования

Цель процесса совершенствования заключается в обеспечении постоянного улучшения потребительских качеств Модуля, включая совершенствование функциональных характеристик, а также нефункциональных свойств Модуля.

Основными задачами процесса являются:

- разработка стратегии совершенствования Модуля;
- регистрация, идентификация и классификация требований по совершенствованию Модуля;
- анализ и оценка зарегистрированных требований для определения приемлемого пути их реализации в рамках совершенствования Модуля;
- реализация требований и отслеживание их вплоть до завершения реализации.

Процесс совершенствования является циклическим. Входными данными для процесса являются требования по совершенствованию, формируемые на основании потребностей пользователей, а также на основании траекторий развития продукта и рынка в целом. Каждое зафиксированное требование классифицируется по категории и приоритету для облегчения анализа тенденций и эффективного использования ресурсов при реализации задач совершенствования.

По зафиксированным требованиям инициируются необходимые действия в соответствии с категорией и приоритетом. При реализации требований совершенствования выполняются процессы проектирования и конструирования, программирования и комплексирования аналогично реализации соответствующих процессов на стадии «Разработка» (процессы подробно описаны в пункте 2.2.2).

Требования по совершенствованию Модуля формируются на основе следующих данных:

- обращения конечных пользователей в организацию-разработчика;
- обращения от пользователей организации-разработчика;
- выявленные ошибки (в рамках процесса устранения неисправностей);
- тенденции развития потребительских предпочтений, продукта и рынка в целом.

Требования по совершенствованию Модуля накапливаются в отдельном разделе в системе Yandex Tracker. По накопленным требованиям проводится регулярный (ежеквартально) анализ, на основании которого определяются требования к реализации, а затем начинается очередной цикл в рамках процесса совершенствования. Состояние реализации по каждому требованию совершенствования отслеживается и отражается в регулярных внутренних отчетах.

Каждое требование совершенствования проходит процедуры классификации, определения приоритетов и отслеживания исполнения в системе Yandex Tracker.

Реализация требований совершенствования выполняется в зависимости от их типа и приоритета, с учетом текущих планов работ. Типы требований совершенствования, создаваемых в системе организации-разработчика:

- обращение;
- превентивная разработка (реализация новой функциональности);
- оптимизация;
- улучшение дизайна;
- улучшение пользовательского опыта;
- улучшение интеграции.

В случае реализации требования в рамках ранее обнаруженной ошибки на этапе тестирования, если тестирование проводилось в рамках работы над уже существующей задачей в системе управления проектами и задачами, информация об ошибке и условиях ее воспроизведения указывается в комментариях к этой задаче. В остальных случаях в системе Yandex Tracker создается новая задача с описанием требования.

Поступающие обращения также анализируются и на их основании формируются (при необходимости) требования по совершенствованию Модуля в форме задачи в системе Yandex Tracker. Указанная задача передается программистам в группу разработки и тестирования. Затем (в соответствии с текущими планами работ и положениями релизной политики) требование анализируется и реализуется программистами, после чего проводится верификация изменений.

2.4.2.6. Процесс технической поддержки пользователей

На стадии «Поддержка» инженеры службы технической поддержки осуществляют техническую поддержку пользователей Модуля. При этом все обращения фиксируются в системе управления задачами (Yandex Tracker) и отслеживаются в ней.

Техническая поддержка Модуля организована в форме приема, регистрации и обработки заявок. Обратиться к инженеру службы технической поддержки Модуля можно по электронной почте platform@big3.ru.

Режим работы службы технической поддержки (по московскому времени):
пн.-пт. 9:00-18:00.

2.4.3. Методы и средства

Техническая поддержка Модуля организована в форме приема и обработки заявок.

Техническая поддержка Модуля организована с учетом категории и критичности проблем. В службу технической поддержки по организованным каналам поступают обращения, которые обрабатываются в порядке их поступления. Вне очереди могут обрабатываться обращения с высоким уровнем критичности, требующие экстренного вмешательства или консультации инженеров службы технической поддержки. К таким обращениям могут быть отнесены, например, вопросы восстановления работоспособности Модуля.

Время решения проблемы, указанной в обращении, может зависеть от критичности обращения, сложности решаемой проблемы и необходимости передачи вопроса в группу разработки и/или в группу тестирования.

При обращении по вопросам поддержки у авторов обращений имеется возможность сообщить о проблемах с прикреплением файлов (изображений, видеофайлов), демонстрирующих суть проблемы.

Для пожеланий и предложений по работе Модуля можно воспользоваться электронной почтой: platform@big3.ru.

Система управления задачами Yandex Tracker – единая с группой разработчиков система, которая позволяет формировать заявки. Заявки классифицируются по категориям (в зависимости от сути обращения), продуктам, функционалу в рамках указанного продукта.

В зависимости от процесса обработки заявке присваивается один из перечисленных ниже статус:

- «Решить»;
- «Протестировано»;
- «Переоткрыть»;
- «Закрыть».

Сведения о взаимодействии с пользователями посредством обратной связи через инженеров службы технической поддержки позволяют получить информацию об их обращениях и проблемах для дальнейшего анализа. Такой подход взаимодействия позволяет учитывать предпочтения и рекомендации пользователей, а также улучшить качество продукта и обслуживания.

Режим работы инженеров службы технической поддержки Модуля: пн.-пт. 9:00-18:00 (по московскому времени).

Процессы поддержки осуществляются по местонахождению организации-разработчика.

2.4.4. Персонал

В ходе реализации процессов стадии «Поддержка» участвует персонал, состав, роли и количество которого, приведены в таблице 7.

Таблица 7 – Сведения о персонале, участвующем в процессах стадии «Поддержка»

Роль	Кол-во, чел.
Менеджер проекта	1
Программист	4
Инженер по тестированию	2
Руководитель группы технической поддержки	1
Инженер службы технической поддержки	2

Требования к квалификации остальных участников процесса стадии «Поддержка» приведены в таблице 4.

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

[illegible]