

**УТВЕРЖДЕН**  
52492702.62001-01 98 01-ЛУ

# **ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ** **Smartfood**

Описание жизненного цикла

52492702.62001-01 98 01

Листов 20

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

2024

## АННОТАЦИЯ

Документ содержит описание жизненного цикла (ЖЦ) программного обеспечения Smartofood, сведения о процессах разработки и совершенствования программного обеспечения, процессах устранения неисправностей, выявленных в ходе эксплуатации программного обеспечения, а также информацию о персонале, необходимом для обеспечения его разработки, совершенствования и поддержки.

Документ разработан с учетом положений следующих нормативных документов:

- ГОСТ Р 57098–2016 «Системная и программная инженерия. Управление жизненным циклом. Руководство для описания процесса»;
- ГОСТ Р 57193–2016 «Системная и программная инженерия. Процессы жизненного цикла систем»;
- ГОСТ Р ИСО/МЭК 12207–2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств»;
- ГОСТ Р ИСО/МЭК 14764–2002 «Информационная технология. Сопровождение программных средств»;
- ГОСТ 19.105–78 «Единая система программной документации. Общие требования к программным документам».

## СОДЕРЖАНИЕ

1. Общие сведения .....	4
1.1. Наименование программы.....	4
1.2. Назначение программы.....	4
1.2.1. Функциональное назначение .....	4
1.2.2. Эксплуатационное назначение .....	4
1.2.3. Особенности применения .....	4
2. Жизненный цикл .....	5
2.1. Стадии и процессы жизненного цикла.....	5
2.2. Стадия «Разработка» .....	7
2.2.1. Общие сведения .....	7
2.2.2. Процессы разработки .....	8
2.2.3. Методы и средства.....	11
2.2.4. Персонал .....	14
2.3. Стадия «Развертывание» .....	14
2.4. Стадия «Поддержка» (сопровождение) .....	15
2.4.1. Общие сведения .....	15
2.4.2. Процесс устранения неисправностей .....	15
2.4.3. Процесс совершенствования .....	17
2.4.4. Техническая поддержка пользователей.....	18
2.4.5. Методы и средства.....	19
2.4.6. Персонал .....	20

## **1. ОБЩИЕ СВЕДЕНИЯ**

### **1.1. Наименование программы**

Полное наименование программы: Smartofood.

В рамках настоящего документа употребляется также обозначение «ПО».

Обозначение программы: 52492702.62001-01.

Smartofood – российское программное обеспечение. Организация-разработчик: Акционерное общество «Смартобрендс» (АО «Смартобрендс»).

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

Организация-правообладатель: Акционерное общество «Смартобрендс» (АО «Смартобрендс»).

### **1.2. Назначение программы**

#### **1.2.1. Функциональное назначение**

Программное обеспечение Smartofood предназначено для автоматизации процессов взаимодействия предприятий общественного питания со своими клиентами в части приема и обработки заказов.

#### **1.2.2. Эксплуатационное назначение**

Программное обеспечение Smartofood реализовано в виде веб-приложения и ориентировано на распространение по модели SaaS.

#### **1.2.3. Особенности применения**

ПО может использоваться в сфере общественного питания.

Smartofood обеспечивает функционирование в многопользовательском режиме.

Серверная часть ПО разворачивается на стороне организации-разработчика и функционирует под управлением операционной системы Ubuntu.

Клиентская часть веб-приложения выполняется на стационарной или переносной ПЭВМ посредством веб-браузера.

В качестве системы управления базами данных (СУБД) применяются MySQL Community Edition и Redis Community Edition.

Пользователями ПО являются сотрудники организаций общественного питания и клиенты таких организаций.

## 2. ЖИЗНЕННЫЙ ЦИКЛ

### 2.1. Стадии и процессы жизненного цикла

Согласно ГОСТ Р 57193–2016 «Системная и программная инженерия. Процессы жизненного цикла систем» и ГОСТ Р ИСО/МЭК 12207–2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств» полный жизненный цикл ПО включает следующие стадии жизненного цикла программного обеспечения:

- замысел;
- разработка;
- развертывание;
- эксплуатация;
- поддержка;
- прекращение эксплуатации.

Каждая стадия жизненного цикла ПО 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 ПО:

- «разработка»;
- «развертывание»;
- «поддержка».

Указанные стадии жизненного цикла ПО направлены на достижение следующих целей:

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

Стадии «замысел» и «прекращение эксплуатации» в настоящем документе не рассматриваются, поскольку каждая из них реализуется одновременно, без разделения на процессы.

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

Описание процессов представлено с учетом положений 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 (описывают 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 ПО стадии могут повторяться и осуществляться параллельно с учетом результатов выполнения процессов на каждой из них.

Процессы ЖЦ ПО обеспечиваются специалистами организации-разработчика, выполняющих следующие роли:

- менеджер проекта;
- руководитель группы разработки;
- программист;
- тестировщик;
- специалист службы технической поддержки.

Высокий уровень качества ПО 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 ПО («code review», unit-тесты, ручное и автоматизированное тестирование сборок, тестирование кандидат-релизов совместно с выделенными группами пользователей);

- сбор и анализ анонимной статистики использования ПО пользователями;
- работа службы технической поддержки в рамках взаимодействия с пользователями ПО.

Основными инструментами автоматизации для поддержания жизненного цикла ПО являются:

- система управления проектами («Битрикс24»);
- система контроля версий (Git);
- средство автоматизированной сборки серверной части (GitLab);
- средство автоматизированной сборки клиентской части (GitLab).

Система управления проектами (задачами) предназначена для:

- организации совместной работы сотрудников организации-разработчика над задачами разработки и исправления ошибок;

- управления задачами по добавлению новой функциональности и исправлению ошибок.

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.

Средства автоматизированной сборки обеспечивают сборку версий частей ПО в заданном окружении по требованию или по расписанию.

## **2.2. Стадия «Разработка»**

### **2.2.1. Общие сведения**

Разработка ПО (размещение инфраструктуры разработки) осуществляется на территории Российской Федерации в облачном сервисе «Яндекс.Облако» по договору № 12234081/24 от 22.04.2024, на серверах, размещенных в дата-центрах по адресам:

- 600902, Владимирская Область, г. о. Город Владимир, г. Владимир, мкр. Энергетик, ул. Поисковая, д. 1 к. 2;

- 391434, Рязанская обл., г. Сасово, ул. Пушкина, д. 21;

- 141004, Московская обл., г. Мытищи, ул. Силикатная, д 19;

- 248926, Калужская обл., г. о. Город Калуга, г. Калуга, пр-д. 1-й Автомобильный, д. 8.

В рамках стадии «Разработка» выполняется разработка и совершенствование ПО, в том числе добавление новой функциональности, подключение дополнительных сервисов и выпуск версий. Каждая существенная доработка ПО выполняется как самостоятельный проект в соответствии с формализованной методикой.

### 2.2.2. Процессы разработки

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

- определение и анализ требований;
- проектирование и конструирование;
- программирование и комплексирование;
- тестирование.

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 положений 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 1.

Таблица 1 – Описание процессов стадии «Разработка» для ПО

<b>Процесс «Определение и анализ требований»</b>	
Цели	Выявление требований, которые должны быть учтены при разработке (совершенствовании) ПО. Формализация совокупности технических требований (функциональных и нефункциональных), которые должны быть учтены и реализованы при разработке (совершенствовании) ПО
Выходы	Определена совокупность функциональных и нефункциональных требований, обеспечивающих удовлетворение потребностей пользователей. Требования внесены в систему управления проектами. Сформированы описания «user story» и «use cases» для групп требований. Определена приоритетность реализации требований
Роли участников	Менеджер проекта, руководитель группы разработки, программисты, специалисты службы технической поддержки



Применяемые средства автоматизации	«Битрикс24» (формирование задач на исполнение требований, исправление ошибок)
Координатор	Менеджер проекта
Деятельность	Спецификация требований
Задачи	<p>Сбор требований от пользователей.</p> <p>Анализ накопленных сведений, получаемых от службы технической поддержки.</p> <p>Анализ планируемого применения ПО и планируемых траекторий развития продукта и рынка в целом.</p> <p>Идентификация, оценка и регистрация требований.</p> <p>Определение требуемых характеристик и условий использования ПО.</p> <p>Определение ограничений для системных решений.</p> <p>Формирование основы для ведения переговоров с заказчиками (пользователями ПО) и заключения контрактов.</p> <p>Разработка спецификации требований, которая описывает:</p> <ul style="list-style-type: none"> <li>а) функции и возможности ПО;</li> <li>б) требования деловой среды;</li> <li>в) требования по безопасности, защищенности, эргономике, интерфейсам, рабочим операциям и сопровождению;</li> <li>г) проектные ограничения и квалификационные требования.</li> </ul> <p>Анализ требований:</p> <ul style="list-style-type: none"> <li>а) анализ системных требований на предмет реализуемости, корректности и тестируемости;</li> <li>б) анализ воздействия системных требований на среду применения;</li> <li>в) упорядочение требований (требования расставляются по приоритетам, утверждаются и обновляются);</li> <li>г) оценка затрат, рисков, приоритетности реализации.</li> </ul>
<b>Процесс «Проектирование и конструирование»</b>	
Цели	<p>Определение особенностей и общих механизмов реализации требований в составе ПО.</p> <p>Разработка программно-технической архитектуры решения, позволяющей реализовать необходимые требования</p>

Выходы	<p>Определены архитектурно-технические решения для разработки (совершенствования).</p> <p>Разработаны оперативные постановки задач (частные технические задания) на разработку отдельных функций и модулей ПО.</p> <p>Требования распределены по модулям ПО.</p> <p>Определены интерфейсы взаимодействия (внутренние и внешние)</p>
Роли участников	Руководитель группы разработки, программисты
Применяемые средства автоматизации	Git (учет и отслеживание частных технических заданий, описаний архитектуры и т.п.)
Координатор	Руководитель группы разработки
Деятельность	Проектирование, документирование и согласование архитектурно-технических решений, которые необходимо реализовать
Задачи	<p>Разработка архитектурного проекта, в соответствии с которым выполняется идентификация верхнего уровня архитектуры и элементов ПО и удовлетворяются заданные требования.</p> <p>Распределение требований по элементам ПО.</p> <p>Определение внутренних и внешних интерфейсов.</p> <p>Верификация между системными требованиями и архитектурой ПО.</p> <p>Документирование и согласование архитектурных решений.</p> <p>Разработка частных постановок задач</p>
<b>Процесс «Программирование и комплексирование»</b>	
Цели	Реализация и сборка ПО (новой версии ПО, в случае совершенствования)
Выходы	Функционирующая версия ПО, готовая к прохождению процедур внутреннего и внешнего тестирования
Роли участников	Руководитель группы разработки, программисты, тестировщики
Применяемые средства автоматизации	<p>«Битрикс24» (учет и отслеживание реализации требований и задач).</p> <p>Система контроля версий Git.</p> <p>Средство автоматизированной сборки GitLab</p>
Координатор	Руководитель группы разработки
Деятельность	Реализация требований в соответствии с постановками задач

Задачи	<p>Реализация требований в соответствии с задачами из «Битрикс24» (разработка отдельных модулей и загрузка их в систему контроля версий).</p> <p>Проведение процедур «code review», выполнение unit-тестов.</p> <p>Слияние ветвей проекта (при необходимости).</p> <p>Разработка конфигурации сборки.</p> <p>Автоматизированная сборка ПО.</p> <p>Верификация доработанной версии ПО. Проведение регрессионных тестов.</p> <p>Разработка сценариев тестирования реализованных требований (включая тестовые примеры, входные и выходные данные и т.п.).</p> <p>Разработка документации.</p>
<b>Процесс «Тестирование»</b>	
Цели	Обеспечение готовности версии ПО к релизу
Выходы	Функционирующая версия ПО, готовая к релизу
Роли участников	Руководитель группы разработки, тестировщики, специалисты службы технической поддержки
Применяемые средства автоматизации	Git
Координатор	Менеджер проекта
Деятельность	<p>Проверка функционирования ПО по контрольным спискам (чек-листам).</p> <p>Проверка покрытия требований</p>
Задачи	<p>Проверка функционирования ПО по контрольным спискам (чек-листам).</p> <p>Протоколирование результатов тестирования.</p> <p>Фиксация выявленных ошибок в Git.</p> <p>Классификация выявленных ошибок и выработка решений по их устранению (при необходимости – проведение повторного тестирования)</p>

На выходе стадии «Разработка» ПО представляет собой совокупность исполняемых файлов, контейнеров и связанных ресурсов, которые поступают на вход стадии «Развертывание».

### 2.2.3. Методы и средства

Программное обеспечение разработано с использованием следующих языков программирования:

- серверная часть: PHP;
- клиентская часть: JavaScript.

В качестве средств разработки программного обеспечения применяются:

- среда разработки: Visual Studio Code Community Edition (версия 1.60.0);
- система контроля версий: Git (версия 2.25.1);
- редактор кода: Visual Studio Code Community Edition (версия 1.60.0);
- контейнеризатор Docker 26.0.0 (версия 26.0.0);
- система контроля версий: Git (версия 2.25.1).

В серверной части ПО разработано с применением следующих основных библиотек:

- yoomoney/yookassa-sdk-php (версия 2.9.1);
- sentry/sentry (версия 3.11);
- intervention/image (версия 2.5);
- spatie/image-optimizer (версия 1.6);
- symfony/process (версия 5.0);
- psr/log (версия 1.0);
- psr/cache (версия 1.0);
- bacon/bacon-qr-code (версия 2.0);
- ramsey/uuid (версия 4.2.3);
- jaybizzle/crawler-detect (версия 1.2);
- symfony/filesystem (версия 5.0);
- dragonmantank/cron-expression (версия 3.3);
- mikehaertl/php-shellcommand (версия 1.6);
- predis/predis (версия 1.1);
- behat/transliterator (версия 1.5);
- stripe/stripe-php (версия 9.4);
- aws/aws-sdk-php (версия 3.245);
- twilio/sdk (версия 7.9);
- kreait/firebase-php (версия 6.9).

В клиентской части ПО разработано с применением следующих библиотек:

- bignumber.js (версия 9.0.0);
- gorify (версия 0.2.1);
- IntlMessageFormat (версия 10.0.0);
- Ion.RangeSlider, (версия 2.3.0);
- jQuery JavaScript Library (версия 3.6.3);
- jQuery Mask Plugin (версия 1.14.15);

- jQuery UI (версия 1.12.1);
- libphonenumber-js (версия 1.8.17);
- metismenu (версия 3.0.4);
- moment.js (версия 2.14.1);
- Quill Editor (версия 1.3.6);
- RequireJS (версия 2.3.6);
- Select2 (версия 4.0.6-rc.1);
- SimpleBar.js (версия 4.1.0);
- SweetAlert2 (версия 10.0.0);
- Swiper (версия 8.4.7).

В качестве систем управления базами данных (СУБД) применяются MySQL Community Edition (версия 5.7) и Redis Community Edition (версия 6.2).

После сборки и установки ПО проводится автоматическое и ручное тестирование.

Тестирование выполняется с использованием таблиц основных сценариев. Используются следующие стадии обработки ошибки:

- поиск;
- воспроизведение;
- описание условий;
- передача данных об ошибке программистам.

В рамках автоматического и ручного тестирования осуществляется проход по всем основным сценариям, доступным пользователям.

Проверки осуществляются в следующем объеме:

- совместимость ПО с программным окружением;
- проверка выполнения функции входа в ПО (аутентификация);
- проверка отклика пользовательского интерфейса;
- проверка переключения по разделам в пользовательском интерфейсе;
- набор стандартных контрольных проверок:
  - проверка длительности выполнения операций по открытию форм пользовательского интерфейса;
  - проверка длительности выполнения поисковых операций.

Результаты тестирования версии (сборки) документируются в протоколах тестирования.

Выявленные в ходе проверки ошибки классифицируются по следующим типам:

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

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

К типовым критическим ошибкам относятся следующие ошибки:

- ошибки, блокирующие действия пользователя и мешающие пользоваться ПО;
- нарушения дизайна или функционирования пользовательского интерфейса.

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

#### 2.2.4. Персонал

В ходе реализации процессов ЖЦ стадии «Разработка» участвует персонал, состав, роли и требования к квалификации которого указаны в таблице 2.

Таблица 2 – Требования по количеству и квалификации персонала, участвующего в процессах стадии «Разработка»

Роль	Кол-во, чел.	Квалификация
Менеджер проекта	1	Релевантный опыт руководства проектами и знания в области информационных технологий. Опыт работы на аналогичной позиции не менее 3 лет
Руководитель группы разработки	1	Релевантный опыт руководства группами разработчиков. Знание языков программирования PHP, JavaScript. Понимание принципов CI/CD
Программист	2	Релевантный опыт разработки веб-приложений (серверная и клиентская части). Обязательно: знание языков программирования: PHP, JavaScript
Тестировщик	2	Релевантный опыт ручного и автоматизированного тестирования. Понимание принципов CI/CD
Специалист службы технической поддержки	2	Релевантный опыт работы в службе технической поддержки. Навыки администрирования операционных систем Ubuntu.

#### 2.3. Стадия «Развертывание»

Тиражирование и распространение ПО осуществляется путем установки и настройки серверной части ПО на сервер организации-разработчика. Пользователи получают доступ к ПО в

виде веб-приложения, при помощи стационарной или переносной ПЭВМ посредством любого веб-браузера актуальной версии.

В рамках стадии «Развертывание» осуществляется сборка и развертывание релиза ПО в рамках серверной инфраструктуры организации-разработчика.

## **2.4. Стадия «Поддержка» (сопровождение)**

### **2.4.1. Общие сведения**

Поддержка (сопровождение) ПО осуществляется по адресу: 625007, Тюменская область, г. Тюмень, ул. Валерии Гнаровской, д. 8, оф. 109.

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

Процессы стадии «Поддержка» (сопровождение) состоят из работ и задач, реализуемых персоналом сопровождения (служба технической поддержки организации-разработчика) с привлечением (при необходимости) персонала, задействованного при реализации процессов стадии «Разработка».

В данном документе рассмотрены определенные с учетом положений ГОСТ Р ИСО/МЭК 12207–2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств» и реализуемые на стадии «Поддержка» следующие процессы:

- устранение неисправностей;
- совершенствование;
- техническая поддержка пользователей.

### **2.4.2. Процесс устранения неисправностей**

Цель процесса устранения неисправностей (решения проблем) заключается в обеспечении гарантии того, что все выявленные проблемы в работе ПО 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-разработчика).

По накопленным сведениям о проблемах проводится регулярный 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 ПО;
- поступило обращение от пользователя в службу технической поддержки организации-разработчика.

Во всех указанных случаях обращение регистрируется в системе «Битрикс24», далее проходит процедуры классификации, определения приоритетности 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, в комментариях к этой задаче указывается информация об ошибке и условиях ее воспроизведения. В остальных случаях в системе «Битрикс24» создается новая задача с описанием ошибки и условий ее воспроизведения. Если ошибка возникла в процессе работы над существующей задачей,



программисты сразу приступают к ее устранению. В случае создания новой задачи, срок принятия задачи в работу определяется в зависимости от критичности выявленной ошибки.

С обращениями конечных пользователей работают специалисты службы технической поддержки организации-разработчика.

Тестировщики организации-разработчика воспроизводят проблему или ошибку по данным, полученным специалистами технической поддержки от пользователя. Далее описываются шаги по воспроизведению ошибки. В результате в системе «Битрикс24» создается задача, которая далее передается программистам. Затем проблема анализируется программистами, после чего они вносят необходимые изменения в исходные коды ПО и проводят верификацию изменений (совместно со специалистами службы технической поддержки).

### **2.4.3. Процесс совершенствования**

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

Основными задачами процесса являются:

- разработка стратегии совершенствования ПО;
- регистрация, идентификация и классификация требований по совершенствованию ПО;
- анализ и оценка зарегистрированных требований для определения приемлемого пути их реализации в рамках совершенствования ПО;
- 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 ПО формируются на основе следующих данных:

- обращения пользователей в службу технической поддержки организации-разработчика;
- выявленные ошибки (в рамках процесса устранения неисправностей);

- тенденции развития потребительских предпочтений, продукта и рынка в целом.

По накопленным требованиям проводится регулярный анализ, на основании которого определяются требования к реализации, далее 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 «Битрикс24».

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, в комментариях к этой задаче указывается информация об ошибке и условиях ее воспроизведения. В остальных случаях в системе «Битрикс24» создается новая задача с описанием требования.

С обращениями пользователей работают специалисты службы технической поддержки организации-разработчика. Они анализируют обращение и, при необходимости, формируют на его основе требование по совершенствованию ПО в форме задачи в системе «Битрикс24». Указанная задача передается программистам и тестировщикам. Далее, в соответствии с текущими планами работ и положениями релизной политики, требование анализируется и реализуется программистами, после чего проводится верификация изменений.

#### **2.4.4. Техническая поддержка пользователей**

На стадии «Поддержка» служба технической поддержки организации-разработчика осуществляет техническую поддержку пользователей. В случае подтверждения наличия ошибки, по поводу которой поступило обращение, в системе управления проектами «Битрикс24» формируется задача по совершенствованию ПО в соответствии с требованиями в заявке.

Техническая поддержка ПО организована в форме приема, регистрации и обработки заявок. Обратиться в службу технической поддержки ПО можно по электронной почте: [mail@smartofood.ru](mailto:mail@smartofood.ru) и по телефону: +7 (962) 931-50-13.

Режим работы службы технической поддержки: пн. – пт. 10:00–19:00 (по московскому времени 08:00–17:00, GMT +5).

#### **2.4.5. Методы и средства**

Техническая поддержка ПО организована в форме приема и обработки заявок 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 специалистов службы технической поддержки. К таким обращениям могут быть отнесены, например, вопросы восстановления работоспособности ПО.

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

Специалисты технической поддержки осуществляют прием обращений и классифицируют их по следующим трем типам:

- инцидент;
- консультация;
- оценка.

Специалисты службы технической поддержки осуществляют поддержку в связи с проблемами следующих видов:

- проблемы, связанные с использованием ПО, не требующие вмешательства технических специалистов;
- проблемы, которые не требуют вмешательства в коды ПО;
- проблемы, связанные с кодами и/или программным интерфейсом взаимодействия (API).

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 ПО можно воспользоваться электронной почтой: [mail@smartofood.ru](mailto:mail@smartofood.ru).

Заявки типа «оценка» направляются менеджеру проекта.

Заявки типов «инцидент» и «консультация» обрабатываются специалистами службы технической поддержки.

После проведения тестирования по заявке «инцидент» и подтверждения наличия ошибки, создается задача в системе «Битрикс24».

Сведения о взаимодействии с пользователями посредством обратной связи через службу 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 службы поддержки: пн. – пт. 10:00–19:00 (по московскому времени 08:00–17:00, GMT +5).

Фактический адрес, по которому осуществляется процесс поддержки: 625007, Тюменская область, г. Тюмень, ул. Валерии Гнаровской, д. 8, оф. 109.

#### 2.4.6. Персонал

В таблице 3 приведены сведения о персонале, участвующем в процессах стадии «Поддержка». Сведения о персонале включают в себя роли и количество задействованных сотрудников.

Таблица 3 – Сведения о персонале, участвующем в процессах стадии «Поддержка»

Роль	Количество, чел.
Специалист службы технической поддержки	2
Программист	2
Тестировщик	2

Специалист службы технической поддержки должен иметь навыки общения с пользователями, а также знать функциональные характеристики ПО, включая перечень типовых возможных проблем, процессы ЖЦ и роли других сотрудников организации-разработчика в этих процессах. Требования к квалификации остальных участников процесса стадии «Поддержка» совпадают с требованиями к квалификации участников процесса стадии «Разработка» и приведены в таблице 2.