

Контентная платформа.
Жизненный цикл
программного обеспечения

Оглавление

Оглавление	2
Введение	3
Стадии жизненного цикла	4
1.1 Стадия сбора потребностей	4
1.2 Стадия разработки	4
1.3 Стадия ввода в эксплуатацию	5
1.4 Стадия эксплуатации	6
Технические процессы	7
1.1 Процесс определения требований	7
1.1 Процесс анализа требований	8
1.2 Процесс проектирования архитектуры	9
1.3 Процесс реализации	10
1.4 Процесс сборки системы	11
1.5 Процесс тестирования	12
Версионность ПО	14
Устранение неисправностей ПО	15
Развитие	17

Введение

Этот документ описывает процессы, которые гарантируют сохранение работы программного обеспечения (ПО) на протяжении всего его жизненного цикла. В эти процессы включены устранение обнаруженных неисправностей в ПО во время его использования, улучшение программного обеспечения и информация о персонале, необходимом для обеспечения поддержки.

“Контентная платформа” – программная платформа, позволяющая управлять контентом веб-приложений, настраивать его отображение у конечных пользователей. Программная платформа позволяет работать с помощью визуального веб-интерфейса.

Программная платформа состоит из следующих компонентов:

1. Веб-интерфейс администратора
2. Веб-интерфейс конечного пользователя
3. Серверное приложение, обеспечивающее работу платформы

Стадии жизненного цикла

1.1 Стадия сбора потребностей

В рамках данной стадии идет взаимодействие с Заказчиком, выявление потребностей в изменении программного средства. Написание проектной документации.

1.2 Стадия разработки

Стадия разработки начинается с детального технического уточнения системных требований и проектных решений.

В ходе выполнения стадии проектируется архитектура системы и ее элементов. При реализации системы производятся, комплексированы, испытываются и оцениваются технические и программные средства и интерфейсы, определяются требования к средствам производства, обучения и поддержки, разрабатывается документация. После реализации системы выполняется ее сборка и тестирование исполнителем. Действия стадии осуществляются с учетом требований и целей всех сторон, принимающих участие в стадии разработки и в последующих стадиях.

Стадия завершается готовностью прототипа системы к предварительным испытаниям.

Результатом стадии разработки является программное средство (или прототип), готовая к предварительным испытаниям, вместе с технической документацией, а также требования, решения, оценки и прочие аналитические данные, предназначенные для использования на последующих стадиях.

1.3 Стадия ввода в эксплуатацию

Стадия ввода в действие заключается в изготовлении, сборке, комплексировании и проведении испытаний программного продукта, разработанного на предыдущем этапе, в единственном экземпляре или серийно. Стадия может включать в себя процессы улучшения и реконфигурации программного продукта.

Стадия включает в себя следующие этапы тестирования:

- 1) Предварительные испытания. В ходе предварительных испытаний проверяется работоспособность и соответствие техническому заданию, устраняются выявленные неисправности и недостатки. После этого программное средство передается в опытную эксплуатацию.
- 2) Опытная эксплуатация. В ходе опытной эксплуатации выполняется работа с программным продуктом по назначению, собираются статистические данные о характеристиках и результатах функционирования. При

необходимости осуществляется доработка программного продукта.

- 3) Приемочные испытания. Испытания проводятся после завершения опытной эксплуатации. По результатам приемочных испытаний программное средство передается в промышленную эксплуатацию.

Результатом стадии является ввод программного продукта в промышленную эксплуатацию (стадия эксплуатации).

1.4 Стадия эксплуатации

Началом стадии эксплуатации служит установка и передача программного продукта для применения по назначению.

Стадия может включать в себя контроль характеристик, а также сбор и анализ данных об отклонениях, недостатках и отказах, выявленных в системе поддержки. В результате таких действий могут быть инициированы стадии разработки и производства или стадия изъятия и списания.

Стадия эксплуатации включает в себя процесс выпуска версий, схема которого приведена отдельно.

Технические процессы

1.1 Процесс определения требований

Цель процесса определения требований состоит в выявлении тех требований к программному продукту, которые, если выполнены, позволят обеспечить необходимые услуги пользователям и другим правообладателям в заданной среде применения.

Задачи, которые выполняются в процессе определения требований, включают:

1. Идентификацию правообладателей, то есть тех лиц или организаций, чьи потребности и требования должны быть учтены при разработке программного продукта.
2. Идентификацию, оценку и регистрацию требований, которые выявляются и формулируются на основе взаимодействия с правообладателями и пользователями. Это позволяет собрать и систематизировать информацию о том, какие услуги и функциональность должны быть включены в программный продукт.
3. Определение требуемых характеристик и условий использования программного продукта. Здесь анализируются и уточняются основные параметры, функциональные возможности, производительность, интерфейс и другие аспекты программного продукта, которые должны быть учтены в процессе его разработки.
4. Определение ограничений для системных решений. Это включает выявление и учет технических, временных, финансовых или других ограничений, которые могут влиять на разработку программного продукта и его функциональность.

В результате этих задач формируется основа для проведения переговоров и заключения соглашений о разработке системы. Это позволяет обеспечить взаимопонимание между разработчиками и правообладателями относительно того, какие требования и условия должны быть учтены в процессе разработки программного продукта.

1.1 Процесс анализа требований

Цель анализа требований заключается в преобразовании требований правообладателей в системные технические требования, которые будут использоваться при проектировании и разработке системы.

Задачи, выполняемые в процессе анализа системных требований, включают:

1. Определение системных функциональных и нефункциональных требований, которые описывают проблему или задачу, которую необходимо решить.
2. Уточнение требований путем проведения дополнительных исследований, консультаций и обсуждений с правообладателями и другими заинтересованными сторонами.
3. Анализ системных требований на корректность и тестируемость, чтобы убедиться, что они ясно определены и могут быть проверены на соответствие.
4. Установление приоритетов, утверждение и обновление требований в соответствии с их значимостью и приоритетом для разработки системы.
5. Согласование системных требований с базовыми требованиями заказчика и достижение согласия между всеми заинтересованными сторонами.
6. Формирование и оптимизация предпочитаемого проектного решения, основываясь на системных требованиях и учете ограничений и возможностей системы.
7. Оценка затрат и рисков, связанных с реализацией системных требований, чтобы оценить ресурсы и потенциальные проблемы, которые могут возникнуть в процессе разработки.
8. Доведение системных требований до всех заинтересованных сторон и включение их в базовую линию, которая служит основой для дальнейшего проектирования и разработки системы.

1.2 Процесс проектирования архитектуры

Цель процесса проектирования архитектуры программного продукта заключается в определении того, как системные требования должны быть разделены и организованы относительно элементов программного продукта.

Задачи, выполняемые в процессе проектирования архитектуры программного продукта, включают:

1. Разработку архитектурного проекта, в результате чего определяется верхний уровень архитектуры и элементы программного продукта, которые соответствуют заданным требованиям.
2. Распределение требований по элементам программного продукта, чтобы определить, какие требования относятся к каждому элементу и как они должны быть реализованы.
3. Определение внутренних и внешних интерфейсов каждого элемента, чтобы обеспечить правильное взаимодействие между ними и с внешними системами.
4. Проверку соответствия между системными требованиями и архитектурой программного продукта, чтобы убедиться, что разработанная архитектура действительно удовлетворяет требованиям системы.
5. Отражение системных требований, конструкции, архитектурного проекта программного продукта и их взаимосвязи в базовой линии и передача этой информации всем заинтересованным сторонам.
6. Учет человеческого фактора, эргономических знаний, технических методов и средств в системном проекте, чтобы обеспечить удобство использования и эффективность программного продукта.

1.3 Процесс реализации

Основная цель процесса реализации программных средств заключается в создании специфицированных элементов программного продукта или услуги. В этом процессе происходит преобразование заданных ограничений, связанных с поведением, интерфейсом и производством, в конкретные действия, которые формируют системные компоненты в виде программных продуктов или услуг.

Основные задачи, выполняемые в процессе реализации программных средств, включают:

1. Определение стратегии реализации, что включает в себя разработку плана и установление направления для достижения поставленных целей.
2. Определение технологических ограничений, связанных с реализацией проекта, таких как выбор конкретных технологий, используемых при разработке.
3. Изготовление программных компонентов, которые соответствуют предварительно определенным требованиям и спецификациям.
4. Передача созданных программных компонентов в систему контроля версий для дальнейшего управления и отслеживания изменений.

В дополнение к этим задачам, процесс реализации программных средств включает следующие подпроцессы:

- Процесс анализа требований к программным средствам, включающий сбор, анализ и документирование требований пользователей и системы.
- Процесс проектирования архитектуры программных средств, который определяет общую структуру и организацию программного продукта.
- Процесс детального проектирования программных средств, где осуществляется разработка более подробных спецификаций компонентов и модулей.
- Процесс конструирования программных средств, включающий создание, программирование и реализацию различных модулей и компонентов.
- Процесс комплексирования программных средств, который включает интеграцию различных компонентов в единую систему.

- Процесс квалификационного тестирования программных средств, с целью проверки и подтверждения их соответствия заданным требованиям и функциональности.

Результатом успешного процесса реализации является создание программного компонента, который удовлетворяет требованиям архитектурных решений (проверяемых путем верификации) и требованиям правообладателей (подтверждаемых путем валидации).

В процессе реализации также выполняется разработка соответствующей документации, согласно процессу управления программной документацией.

1.4 Процесс сборки системы

Цель процесса сборки системы заключается в объединении различных системных элементов (включая технические и программные компоненты, ручные операции и другие системы, если необходимо) для создания программного продукта, который полностью соответствует проекту и ожиданиям заказчика, выраженным в требованиях.

Задачи, выполняемые в процессе комплексирования, включают:

1. Определение стратегии сборки программного продукта в соответствии с приоритетами системных требований. Это включает планирование последовательности и способа объединения различных компонентов системы.
2. Разработку критериев для проверки соответствия системным требованиям, которые распределяются по элементам системы, включая интерфейсы между ними. Это помогает обеспечить правильное взаимодействие и работоспособность системы в ее собранном состоянии.
3. Проверку собранного программного продукта на соответствие определенным критериям. Это включает проведение тестирования и проверку функциональности,

чтобы убедиться, что система работает правильно и соответствует системным требованиям.

4. Разработку и применение стратегии регрессионного тестирования для повторного тестирования программного продукта в случае внесения изменений. Это гарантирует, что после внесения изменений в систему ее работоспособность и соответствие требованиям остаются неизменными.
5. Выполнение сборки программного продукта, который демонстрирует полную функциональность и соответствие системному проекту. Это включает объединение всех необходимых системных элементов и их проверку в целостной системе.

1.5 Процесс тестирования

Цель процесса тестирования системы заключается в проверке каждого требования на соответствие и подтверждении готовности программного продукта к поставке.

Задачи, выполняемые в процессе тестирования программного продукта, включают:

1. Разработку критериев для оценки соответствия системным требованиям. Это включает определение критериев, по которым будет оцениваться функциональность и работоспособность программного продукта.
2. Тестирование программного продукта после сборки. В этой задаче выполняется набор тестов, которые проверяют работу программного продукта в соответствии с определенными требованиями и оценивают его производительность, надежность и другие аспекты.
3. Документирование результатов тестирования. В данной задаче регистрируются и описываются результаты проведенных тестов, включая обнаруженные дефекты, отклонения от требований и другую информацию о работе программного продукта.

Результатом тестирования программного продукта является подтверждение его соответствия установленным требованиям и готовности к передаче на следующий этап разработки или поставку. Это гарантирует, что программное средство функционирует должным образом и удовлетворяет требованиям заказчика перед тем, как будет внедрено или использовано.

Версионность ПО

Каждая версия ПО имеет свой идентификатор (номер) версии, который строится по следующему формату:

N1.N2

Где:

N1 – версия (глобальные изменения),

N2 – релиз (изменения или дополнения отдельных функций),

Например, «Контентная платформа» версия 1.0.

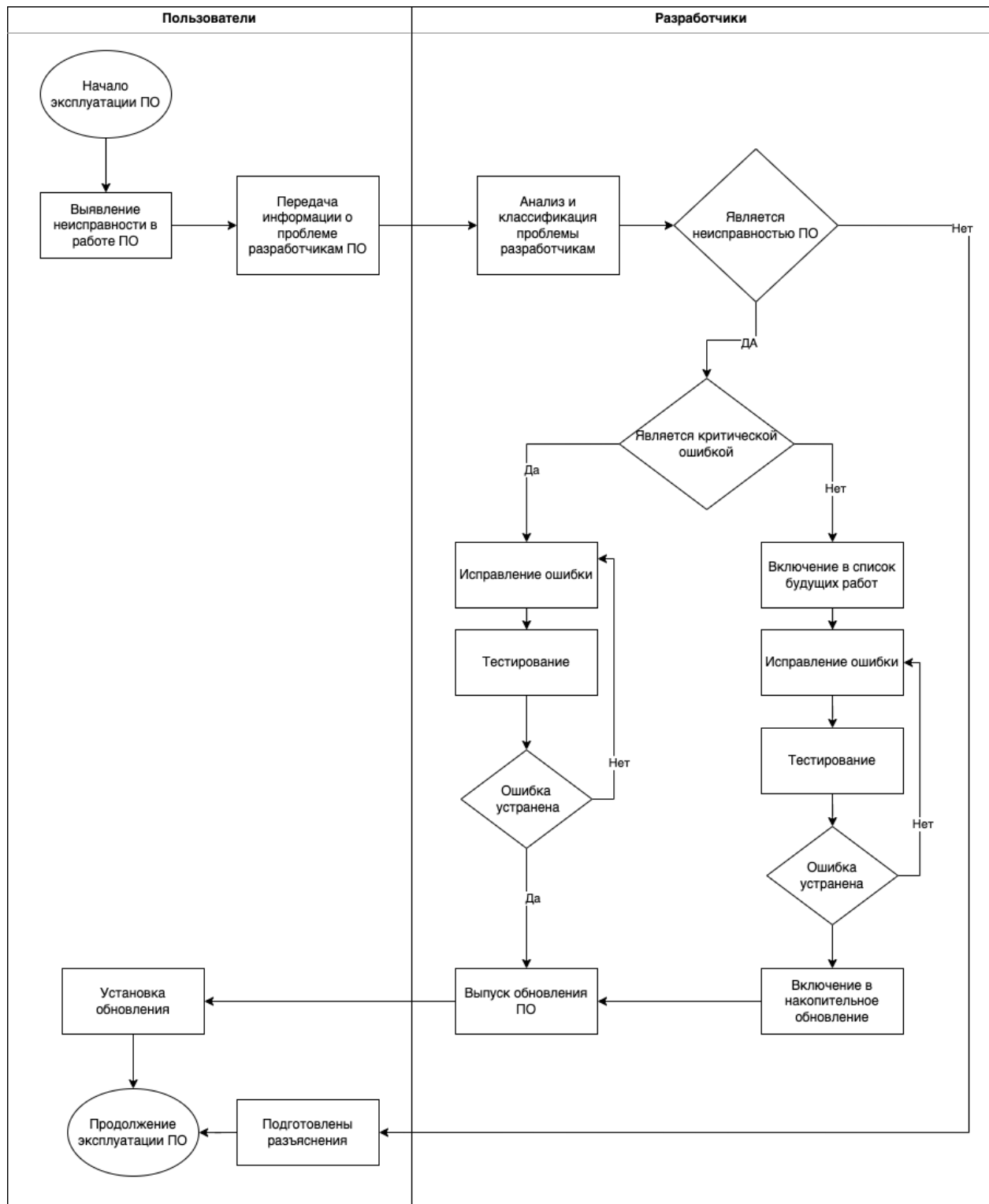
По номеру версии можно просмотреть список выполненных изменений функционала, либо список исправленных ошибок.

Новые версии ПО выходят периодически, в зависимости от накопленных изменений (исправление ошибок, новый функционал). Так же, возможны внеплановые релизы в связи с:

1. Критическими ошибками
2. Изменениями в бизнес-процессах;
3. Выпуском новых нормативных и правовых изменений направленных на работу компаний в сфере ИТ (включая решения профильных министерств, таких как МинЦифры);

В иных случаях, модификация кода программного обеспечения проводится в рамках доработок по запросам клиентов – пользователей данного программного обеспечения.

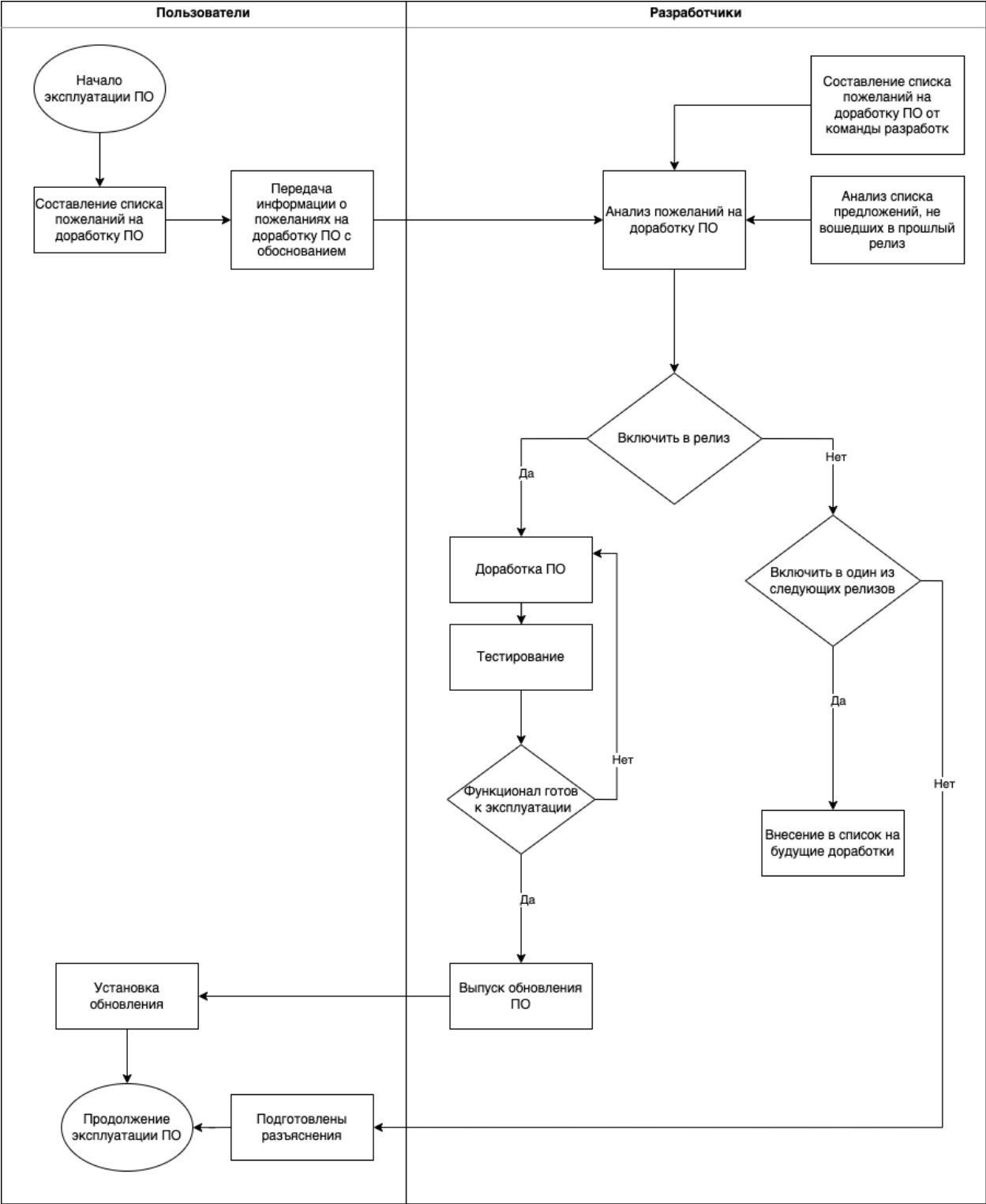
Устранение неисправностей ПО



На схеме выше изображен процесс работы с ошибками, возникающими по мере эксплуатации программного обеспечения пользователями:

При выявлении неисправности в работе ПО, пользователь сообщает разработчику ПО об ошибке, используя каналы связи, указанные в договоре между пользователем и разработчиком ПО. Проводится анализ ошибки и, в случае, если ошибка подтверждена, проводится классификация проблемы. Ошибка может классифицироваться как критическая, если она напрямую мешает работе системы и некритичная. В случае критичности ошибки, она оперативно берется в работу и обновление ПО выпускается в момент готовности исправления ошибки. Если ошибка классифицируется как некритичная, она берется в работу и выпускается в пакете со следующей версией ПО. Если ошибка не подтверждается разработчиками, оказывается консультация по работе ПО.

Развитие



На схеме выше изображен процесс работы с развитием программного продукта: Пользователь ПО сообщает разработчику ПО предложения по улучшению ПО, используя каналы связи, указанные в договоре между пользователем и разработчиком ПО. Так же, собирается информация из внутренних внешних источников, которая может повлиять на развитие ПО. Помимо этого, анализируются предложения от пользователей и внутренних источников, не вошедшие в предыдущий релиз. Проводится анализ полученной информации, расставляются приоритеты по задачам. В зависимости от важности и срочности изменений, доработки могут пойти в ближайший релиз ПО, либо уйти в список на будущие доработки.