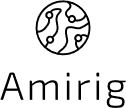
**Руководство**

**по процессам разработки и сопровождению программного продукта Amirig**



|  |  |  |  |
| --- | --- | --- | --- |
| **Разработано** | **Редакция** | **Согласовано** | Версия – v.1.0 |
| Горшков В.А. | Сирачков Ф.А. | Маслянинов В.П. | Дата выпуска – 03.09.2020 |
|  |  |  |  |

Оглавление

[1. Общее 3](#_Toc59724366)

[2. Модель жизненного цикла ПО Amirig 3](#_Toc59724367)

[2.1. Модель на основе разработки прототипа 4](#_Toc59724369)

[2.2. Структура процессов жизненного цикла ПО Amirig 4](#_Toc59724370)

[3. Структура документов, этапов (основные процессы) 5](#_Toc59724371)

[3.1. Разработка ПО 5](#_Toc59724372)

[3.2. Внедрение 6](#_Toc59724373)

[4. Вспомогательные (поддерживающие) процессы 6](#_Toc59724374)

[5. Организационные процессы 6](#_Toc59724375)

[ПРИЛОЖЕНИЕ А. Обеспечение процесса управления по ГОСТ Р ИСО/МЭК 12207. 8](#_Toc59724376)

[ПРИЛОЖЕНИЕ B. Схема использования руководства. 10](#_Toc59724377)

[ПРИЛОЖЕНИЕ С. Процессы жизненного цикла. 11](#_Toc59724378)

[ПРИЛОЖЕНИЕ D. Стадии разработки. 12](#_Toc59724379)

[ПРИЛОЖЕНИЕ E. Периодичность. Организация рабочего процесса. 14](#_Toc59724381)

# 1. Общее

В качестве основы правил и требований, отражённых в данном документе, использовались рекомендации, отражающие понятие модели жизненного цикла, включающего в себя комплекс нормативных и методических документов, регламентирующих процессы создания программного продукта, с ориентацией на Российские и международные стандарты:

* Международный стандарт ISO/IEC 12207: 1995 «Information Technology — Software Life Cycle Processes» (ISO — International Organization for Standardization — Международная организация по стандартизации; IEC — International Electrotechnical Commission — Международная комиссия по электротехнике; название стандарта «Информационные технологии — Процессы жизненного цикла программ»).
* ГОСТ 19.102-77 – Единая система программной документации - стадии разработки (Unified system for program documentation. Development stages).
* ГОСТ Р ИСО/МЭК 12207 – жизненный цикл программного средства от возникновения идеи его создания до снятия с эксплуатации.
* ГОСТ 34.601-90 – стадии и этапы создания автоматизированной системы.
* Библиотека ITIL – (Office of Government Commerce – OGC).
* Стандарты серии ISO-9000 и общие схемы обеспечения качества (Total Quality frameworks), предлагаемые Европейской организацией Управления Качеством (European Foundation of Quality Management – EFQM).

# 2. Модель жизненного цикла ПО Amirig

Модель жизненного цикла создания программного продукта Amirig — это рабочих документов и правил, определяющих последовательность выполнения работ и взаимосвязь процессов, действий и задач на протяжении всего жизненного цикла.

Модель жизненного цикла ПО Amirig включает в себя:

1. **Стадии** – часть процесса создания программного продукта Amirig, ограниченная определёнными временными рамками и заканчивающаяся выпуском конкретного продукта (моделей, программных компонентов, документации), определяемого заданными для данной стадии требованиями.
2. **Результаты выполнения** работ на каждой стадии.
3. **Ключевые события** — точки завершения работ и принятия решений.

В качестве модели кодирования и устранения ошибок принята «V модель», обеспечивающая разработку ПО через тестирование, что является одной из основных практик экстремального программирования.

# 

# 2.1. Модель на основе разработки прототипа

Модель на основе разработки прототипа основывается на разработки прототипов и прототипирования продукта на ранних стадиях жизненного цикла программного обеспечения:

1. Прояснить не ясные требования (прототип UI)
2. Выбрать одно из ряда концептуальных решений (реализация сценариев)
3. Проанализировать осуществимость проекта

Классификация протопопов:

1. **Горизонтальные прототипы** — моделирует исключительно UI, не затрагивая логику обработки и базу данных.
2. **Вертикальные прототипы** — проверка архитектурных решений.
3. **Одноразовые прототипы** — для быстрой разработки.
4. **Эволюционные прототипы** — первое приближение эволюционной системы.

# 2.2. Структура процессов жизненного цикла ПО Amirig

Единая модель жизненного цикла ПО Amirig, устанавливает последовательность процессов, принятых в компании при разработке модулей программного продукта, проведение процесса тестирования, прохождение стадий выполнения работ и подготовка сопроводительной документации.

1. Формирование требований к программному продукту Amirig или его компонентам.[[1]](#footnote-1)
   1. Обследование объекта и обоснование необходимости создания ПО
   2. Формирование требований пользователя к ПО
   3. Оформление отчёта о выполнении работ и заявки на разработку ПО
2. Разработка концепции ПО
   1. Изучение объекта
   2. Проведение необходимых научно-исследовательских работ
   3. Разработка вариантов концепции ПО и выбор варианта концепции, удовлетворяющего требованиям пользователей
   4. Оформление отчёта о проделанной работе
3. Техническое задание
   1. Разработка и утверждение технического задания на создание ПО
4. Эскизный проект
   1. Разработка предварительных проектных решений ПО и его частям
   2. Разработка документации на ПО и его части
5. Технический проект
   1. Разработка проектных решений ПО и его частям
   2. Разработка документации на ПО и его части
   3. Разработка заданий на проектирование в смежных частях проекта
6. Рабочая документация
   1. Разработка рабочей документации на ПО и его части
   2. Разработка и адаптация ПО
7. Ввод в действие
   1. Подготовка объекта автоматизации
   2. Подготовка персонала
   3. Комплектация ПО поставляемыми изделиями (программными и техническими средствами, программно-техническими комплексами, информационными изделиями)
   4. Проведение тестирования
   5. Проведение опытной эксплуатации
   6. Проведение приёмочных испытаний
8. Сопровождение ПО
   1. Выполнение работ в соответствии с гарантийными обязательствами
   2. Послегарантийное обслуживание

# 3. Структура документов, этапов (основные процессы)

Требования к структуре документов процесса. Минимальный набор требований, которые должны быть отражены в итоговой документации.

# 3.1. Разработка ПО

**Техническое задание**

* постановка задачи
* выбор критериев эффективности
* проведение предварительных научно-исследовательских работ (НИР)
* разработка ТЗ

**Эскизный проект**

* отражение структуры входных и выходных данных
* уточнение методов решения; общий алгоритм
* разработка документации эскизного проекта

**Технический проект**

* уточнение структуры входных и выходных данных
* разработка алгоритмов
* формы данных
* семантика и синтаксис языка
* структура программы
* конфигурация технических средств
* план работ

**Рабочий проект**

* программирование и отладка
* разработка документов
* подготовка и проведение испытаний
* корректировка программы и документов по итогам испытаний

# 3.2. Внедрение

* передача программы и документов для сопровождения
* оформление акта

# 4. Вспомогательные (поддерживающие) процессы

**Процесс приобретения**

Охватывает действия заказчика по приобретению ПО. К этим действиям относятся:

* инициирование приобретения;
* подготовка заявочных предложений;
* подготовка и корректировка договора;
* надзор за деятельностью поставщика;
* приёмка и завершение работ.

**Процесс поставки**

Охватывает действия и задачи поставщика при снабжении заказчика ПО или услугой. К этим действиям относятся:

* инициирование поставки;
* подготовка ответа на заявочные предложения;
* подготовка договора;
* планирование;
* выполнение и контроль;
* проверка и оценка;
* поставка и завершение работ.

# 5. Организационные процессы

**Процесс сопровождения**

Охватывает действия и задачи сопровождающей организации (службы сопровождения). Данный процесс активизируется при изменениях (модификациях) ПО и соответствующей документации, вызванных возникшими проблемами или потребностями в модернизации либо адаптации ПО.

В соответствии со стандартом IEEE-90 (IEEE –Institute of Electrical and Electronics Engineers – Институт инженеров по электротехнике и электронике) под сопровождением по­нимается внесение изменений в ПО в целях исправления ошибок, повышения производительности либо адаптации к изменившимся условиям работы или требованиям.

* регистрация новых требований на элементы ПО
* журнализация решений о переходе требований в задание на доработку
* регистрация ошибок в багтрекере
* регистрация заданий исправления ошибок
* планирование работ по исправлению ошибок
* планирование работ по тестированию

**Процесс документирования (documentation process)**

Предусматривает формализованное описание информации, созданной в течение жизненного цикла ПП. Данный процесс состоит из набора действий, с помощью которых планируют, проектируют, разрабатывают, выпускают, редактируют, распространяют и сопровождают документы, необходимые для всех заинтересованных лиц, таких как руководство, технические специалисты и пользователи системы.

Процесс документирования включает в себя:

* подготовительную работу
* проектирование и разработку документации
* выпуск документации
* сопровождение

**Управление Релизами**

Релизом называется набор Конфигурационных Единиц, которые совместно тестируются и вводятся в активную рабочую среду. Главной задачей Управления Релизами является обеспечение успешного развёртывание релизов, включая интеграцию, проведение тестирования и хранение.

Управление Релизами обеспечивает гарантию того, что в использовании находятся только тестированные и корректные версии авторизованного программного и аппаратного обеспечения. Управление Релизами тесно связано с деятельностью по Управлению Конфигурациями и Управлению Изменениями. Реальное внесение изменений часто осуществляется через действия в рамках Процесса Управления Релизами.

* журнал регистрации выпуска релизов
* журнал регистрации обновления ПО на хостингах

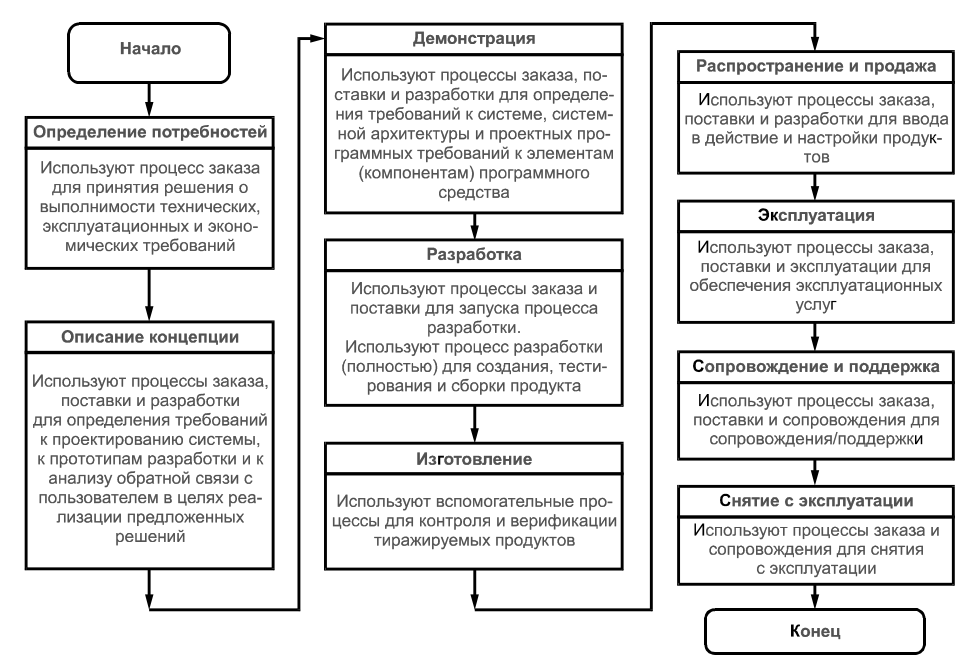
# ПРИЛОЖЕНИЕ А. Обеспечение процесса управления по ГОСТ Р ИСО/МЭК 12207.

Таблица 1 - Обеспечение работ процесса управления основными процессами (по ГОСТ Р ИСО/МЭК 12207)

| Основные процессы жизненного цикла | Работы процесса управления | | | | |
| --- | --- | --- | --- | --- | --- |
| Подготовка и определение области управления | Планирование | Выполнение и контроль | Проверка и оценка | Завершение |
| Процесс заказа |  |  |  |  |  |
| Подготовка | X |  |  |  |  |
| Использование процесса разработки |  |  | X |  |  |
| План заказа |  | X |  |  |  |
| Подготовка заявки на подряд | X | X |  |  |  |
| Подготовка и корректировка договора | X |  | X |  |  |
| Контроль и оценка изменений |  |  |  | X |  |
| Надзор за поставщиком |  |  | X | X |  |
| Приёмка и закрытие договора |  |  |  | X | X |
| Процесс поставки |  |  |  |  |  |
| Подготовка | X |  |  |  |  |
| Подготовка ответа | X |  |  |  |  |
| Подготовка договора | X |  |  | О |  |
| Планирование |  | X |  |  |  |
| Выполнение и контроль |  |  | X | О |  |
| Проверка и оценка |  |  |  | X |  |
| Поставка и закрытие договора |  |  |  |  | X |
| Процесс разработки |  |  | X |  |  |
| Подготовка процесса | X |  |  |  |  |
| Выбор модели жизненного цикла и разметка работ |  | X |  |  |  |
| Разработка и выполнение планов процесса |  | X |  |  |  |
| Оценка системных требований |  |  |  | X |  |
| Оценка системной архитектуры и требований к ней |  |  |  | X |  |
| Оценка требований к программным средствам |  |  |  | X |  |
| Проведение совместных анализов требований к программным средствам |  |  |  | X |  |
| Оценка программной архитектуры и проектов интерфейсов и баз данных |  |  |  | X |  |
| Проведение совместных анализов проекта программной архитектуры |  |  |  | X |  |
| Оценка технического программного проекта и требований к тестированию |  |  |  | X |  |
| Проведение совместных анализов технического программного проекта |  |  |  | X |  |
| Оценка машинных программ и результатов их тестирования |  |  |  | X |  |
| Разработка плана сборки |  | Х |  |  |  |
| Оценка плана сборки, проекта программ, тестов, результатов тестирования и документов пользователя |  |  |  | X |  |
| Проведение совместных анализов сборки программных средств |  |  |  | X |  |
| Оценка проекта, программ, тестов, результатов тестирования и документов пользователя |  |  |  | X |  |
| Обеспечение аудиторских проверок при квалификационных испытаниях программных средств |  |  |  |  |  |
| Оценка собранной системы |  |  |  | X |  |
| Оценка системы |  |  |  | X |  |
| Обеспечение аудиторских проверок при квалификационных испытаниях программных средств |  |  |  | X |  |
| Разработка плана ввода в действие программных средств | О | X |  |  |  |
| Обеспечение проведения заказчиком анализа готовности к приёмке и приёмочных испытаний программных средств |  |  |  | X |  |
| Процесс эксплуатации |  |  | X |  |  |
| Подготовка процесса |  | X |  |  |  |
| Процесс сопровождения |  |  |  |  |  |
| Подготовка процесса |  |  | X |  |  |
| Разработка и выполнение планов и процедур по сопровождению |  | X |  |  |  |
| Установление процедур для: получения, документирования и контроля сообщений о проблемах и заявок на изменения; обеспечения обратной связи |  | X |  |  |  |
| Анализ проблем и изменений |  |  | X |  |  |
| Анализ сообщений о проблемах или заявок на изменения |  |  |  | X |  |
| Принятие вариантов реализации изменений | X |  |  |  |  |
| Внесение изменений |  |  | X |  |  |
| Проверка и приёмка при сопровождении |  |  |  | X |  |
| Перенос |  |  | X |  |  |
| Разработка и выполнение плана переноса |  | X |  |  |  |
| Анализ последствий переноса |  |  |  | X |  |
| Снятие с эксплуатации |  |  | X |  | X |
| Разработка и выполнение плана снятия с эксплуатации |  | X |  |  |  |

# ПРИЛОЖЕНИЕ B. Схема использования руководства.

Использование ГОСТ Р ИСО/МЭК 12207 для обеспечения модели жизненного цикла ПО



# ПРИЛОЖЕНИЕ С. Процессы жизненного цикла.

Только основные процессы. Справочная таблица.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Периоды жизненного цикла | Процессы жизненного цикла | | | | |
| Заказ | Поставка | Разработка | Эксплуатация | Сопровождение |
| Определение потребностей | X |  | X |  |  |
| Исследование и определение концепции | X | X | X |  |  |
| Демонстрация и аттестация | X | X | X |  |  |
| Проектирование и разработка | X | X | X |  |  |
| Создание и производство |  |  |  |  |  |
| Распространена и продажа | X | X | X |  |  |
| Эксплуатация | X | X |  | X |  |
| Сопровождение и поддержка | X | X |  |  | X |
| Снятие с эксплуатации | X |  |  |  | X |

# ПРИЛОЖЕНИЕ D. Стадии разработки.

Настоящее требование устанавливает стадии разработки программ и программной документации. Этапы и содержание работы должно соответствовать указанным в таблице.

|  |  |  |
| --- | --- | --- |
| Стадии разработки | Этапы работ | Содержание работ |
| 1. Техническое задание | Обоснование необходимости разработки программы. | Постановка задачи.  Сбор исходных материалов.  Выбор и обоснование критериев эффективности и качества разрабатываемой программы.  Обоснование необходимости проведения научно-исследовательских работ |
| Научно-исследовательские работы | Определение структуры входных и выходных данных.  Предварительный выбор методов решения задач.  Обоснование целесообразности применения ранее разработанных программ.  Определение требований к техническим средствам.  Обоснование принципиальной возможности решения поставленной задачи. |
| Разработка и утверждение технического задания | Определение требований к программе.  Разработка технико-экономического обоснования разработки программы.  Определение стадий, этапов и сроков разработки программы и документации в неё.  Выбор языков программирования.  Определение необходимости проведения научно-исследовательских работ на последующих стадиях.  Согласование и утверждение технического задания. |
| 1. Эскизный проект | Разработка эскизного проекта | Предварительная разработка структуры входных и выходных данных.  Уточнение методов решения задачи.  Разработка общего описания алгоритма решения задачи.  Разработка технико-экономического обоснования. |
| Утверждение эскизного проекта | Разработка пояснительной записки.  Согласование и утверждение эскизного проекта. |
| 1. Технический проект | Разработка технического проекта. | Уточнение структуры входных и выходных данных.  Разработка алгоритма решения задачи.  Определение формы представления входных и выходных данных.  Определение семантики и синтаксиса языка.  Разработка структуры программы.  Окончательное определение конфигурации технических средств |
| Утверждение технического проекта | Разработка плана мероприятий по разработке и внедрению программ.  Разработка пояснительной записки.  Согласование и утверждение технического проекта. |
| 1. Рабочий проект | Разработка программы.  Разработка программной документации. | Программирование и отладка программы.  Разработка программных документов в соответствии с требованиями ГОСТ 19.101-77. |
| Испытания программы. | Разработка, согласование и утверждение программы и методики испытаний. |
| Испытания программы | Проведение предварительных государственных, межведомственных, приемо-сдаточных и других видов испытаний.  Корректировка программы и программной документации по результатам испытаний. |
| 1. Внедрение | Подготовка и передача программы | Подготовка и передача программы и программной документации для сопровождения и (или) изготовления.  Оформление и утверждение акта о передаче программы на сопровождение и (или) изготовление.  Передача программы в фонд алгоритмов и программ |

# 

# ПРИЛОЖЕНИЕ E. Периодичность. Организация рабочего процесса.

Описание процессов, обеспечивающих поддержание жизненного цикла программного обеспечения «Amirig»

**Система управления проектами, задачами и отслеживания ошибок**

Новые задачи в проекте разработки ПО Amirig возникают при необходимости добавить новую функциональность или устранить ошибки, выявленные в процессе тестирования или эксплуатации.

Работы по добавлению новой функциональности ведутся в соответствии с планом работ по развитию ПО (планом релизов). Задачи по исправлению ошибок создаются в случае выявления ошибок в ходе ручного или автоматизированного тестирования, или в процессе эксплуатации ПО.

В соответствии с планом релизов системные инженеры разрабатывают частное техническое задание на добавление новой функциональности, в которое включают список функциональных и сопутствующих им нефункциональных требований. В системе управления задачами «Bugtracker» в рамках проекта разработки ПО Amirig создаётся новая задача, к которой прикладывается частное техническое задание (список требований), необходимые нормативные документы (стандарты, регламенты и т.п.) и иные вспомогательные материалы. Задачи по исправлению ошибок создаются инженерами по тестированию и специалистами по качеству программного обеспечения по результатам ручного или автоматизированного тестирования ПО Amirig, а также инженерами службы технической поддержки по заявкам от клиентов. В описание таких задач включают сведения, необходимые для воспроизведения ошибки, к которым, как правило, относятся описание характеристики. Помимо этого, к задаче по исправлению ошибок могут прилагаться ссылки на стандарты и регламенты, описывающие ожидаемое поведение. Вновь созданная задача получает статус («Новая») и назначается руководителю группы разработки ПО.

Руководитель группы разработки ПО Amirig анализирует описание задачи с точки зрения достаточности приведённой в нем информации для начала процесса внесения изменений в исходные коды ПО. При необходимости он обращается к системным инженерам, инженерам по качеству и тестированию ПО, инженерам технической поддержки с целью уточнения описания задачи. После этого задаче назначается исполнитель из числа программистов-разработчиков и специалист по контролю качества из числа инженеров по тестированию. В случае необходимости привлечения для решения задачи сразу нескольких исполнителей, она разбивается на более мелкие задачи. Разбиение, как правило, производится таким образом, чтобы каждая подзадача сводилась к логически изолированным изменениям в исходных кодах ПО, которые могли бы быть выполнены и протестированы вне связи с другими подзадачами. Такой подход обеспечивает возможность параллельной работы с подзадачами группы из нескольких исполнителей. Если же выполнение какой-либо задачи (подзадачи) блокирует выполнение других задач, то это фиксируется с использованием соответствующих инструментов «Bugtracker».

Получив уведомление о назначении на задачу, разработчик переводит её в состояние «Развитие» с резолюцией («Работа начата») и приступает к внесению изменений в исходные тексты ПО Amirig.

В случае, если работа над задачей невозможна в данный момент времени по таким причинам, как, например, изменение в плане работы (плане релизов), наличие более приоритетных задач или наличие неудовлетворённых зависимостей, то разработчик по согласованию с руководителем группы разработки ПО и руководителем проекта переводит задачу в состояние («Пауза») с указанием конкретных причин принятия данного решения. Из данного состояния задача может быть в любой требуемый момент возвращена в работу.

В случае, если в процессе работы над задачей выявлена её неактуальность или невозможность её решения в силу каких-либо технических причин, программист-разработчик по согласованию с руководителем разработки и руководителем проекта переводит задачу в состояние («Отклонена») указанием конкретных причин принятия данного решения. При необходимости это решение может быть пересмотрено в будущем.

Завершив внесение изменений в исходные тексты ПО, разработчик инициирует создание тестовой сборки ПО. На этом этапе к работе над задачей подключается инженер по контролю качества (тестированию) ПО.

На тестовой сборке выполняется ручное тестирование функциональности, заявленной в описании задачи, если задача связана с добавлением новой функциональности, или проверяется возможность воспроизведения ошибки, если задача направлена на устранение ошибок функционирования ПО. Также на тестовой сборке, как правило, выполняется автоматизированное поверхностное тестирование с целью убедиться в корректности работы базовых функций и основных модулей ПО Amirig.

В процессе проведения тестирования специалист по качеству (тестированию) ПО в основном руководствуется спецификациями, приведёнными в описании задачи, но может также проводить дополнительные тесты для того, например, чтобы оценить влияние изменений, внесённых в исходные тексты, на смежный функционал. В процессе тестирования воспроизводятся типовые и граничные условия и оценивается корректность работы ПО с точки зрения решения поставленной задачи. В процессе проведения тестирования могут быть задействованы целевые аппаратные платформы, виртуальные машины, генераторы трафика и иные средства, определяемые методикой тестирования. Для проведения тестирования сборка ПО, разворачивается на целевой аппаратной платформе или виртуальной машине, после чего тестовое устройство (или виртуальная машина) включается в топологию, определяемую спецификацией задачи и методикой тестирования, выполняются необходимые тесты. Задача, находящаяся в состоянии («Обеспечение качества»), при необходимости может быть переведена в состояние («Пауза») или («Отклонена») с указанием конкретных причин такого решения.

В случае, если тесты выявляют некорректную работу и ошибки функционирования ПО, или ошибка, которая должна быть устранена в ходе решения задачи, снова воспроизводится на тестовой сборке, задача возвращается в разработку (переводится в состояние «Развитие») с резолюцией («Проблемы на тестовой сборке»). При необходимости инженер по тестированию добавляет в описание задачи условия и технологию воспроизведения проблемы, выявленной в ходе тестирования. Разработчик анализирует выявленную проблему и вносит в исходные тексты ПО изменения, необходимые для е устранения, после чего повторно переводит задачу в состояние «гарантия качества» с резолюцией («Тестовая сборка»). Процедуры тестирования и исправления ошибок повторяются до тех пор, пока все проблемы, возникающие на данном этапе работы, не будут устранены.

Если очередная проверка тестовой сборки не выявляет каких-либо проблем, задача возвращается разработчику с резолюцией («Тестовая сборка верифицирована»). В процессе работы над задачей и верификации тестовой сборки на основной рабочей ветке в репозиторий исходных кодов могут накапливаться изменения.

В рамках ручного тестирования в зависимости от задачи проверяется правильность работы вновь введённой функциональности или результат исправления ошибок. В основном методика верификации сборки на данном этапе совпадает с ранее рассмотренной методикой верификации тестовой сборки. Ручное тестирование направлено на проверку корректности решения поставленной задачи в рамках объединённой сборки. В свою очередь регрессионное тестирование позволяет выявить возможные проблемы, когда изменения в исходных кодах, выполненные в рамках решения задачи, влияют на стороннюю функциональность.

В случае выявления проблем в процессе тестирования инженер по качеству (тестированию) ПО фиксирует их характер и описывает методику воспроизведения, после чего возвращает задачу на доработку (переводит в состояние «Development») с резолюцией («Проблемы сборки»). Разработчик анализирует выявленные проблемы, устраняет их причины, готовит новый (исправленный) вариант сборки и отправляет его на повторное тестирование. Процесс, таким образом, повторяется до тех пор, пока все проблемы на сборке не будут устранены.

Если тестирование сборки не выявляет никаких проблем, задача возвращается разработчику (переводится в состояние «Development») с резолюцией («Объединённая сборка верифицирована»). Разработчик готовит и отправляет в главный репозиторий исходных кодов проекта запрос на принятие изменений (Pull request). При этом задача в «Bugtracker» переводится в состояние («Рецензирование») с резолюцией «Pull request».

Перед внесением изменений в главный репозиторий проекта выполняется проверка качества исходных кодов (проверка кода). Проверку качества обычно осуществляет один из наиболее опытных разработчиков или непосредственно сам руководитель группы разработки ПО. В ходе проверки прежде всего контролируется соответствие внесённых изменений принятым стандартам оформления; целесообразность применённых архитектурных решений; наличие потенциальных источников уязвимостей; достаточность, актуальность и информативность комментариев в коде.

При наличии замечаний рецензент фиксирует их в комментариях к задаче в «Bugtracker» и саму задачу возвращает разработчику с резолюцией («Необходимы изменения»). Разработчик устраняет причины замечаний и отправляет задачу на повторную проверку. Процесс повторяется, пока все замечания не будут устранены. После этого задача переводится в состояние («Выполнено») с резолюцией («Изменения в исходном коде»).

**Организация работы с системой контроля версий**

В качестве системы контроля версий используется система GitHub. Хостинг проекта осуществляется по url: https://expo.amirig.com/, которая обеспечивает веб-интерфейс для распределённого доступа к проекту, обеспечивает функциональность для проведения верификации кода.

Под управлением системы контроля версий находятся все компоненты программного обеспечения ПО Amirig, включая сторонние Open Source компоненты. Используется схема, при которой все основные модули программного обеспечения хранятся в едином репозиторий исходных кодов. В репозитории хранятся исходные коды системы сборки.

Для решения конкретной задачи, связанной с введением новой функциональности или исправлением ошибок, в соответствии с методикой организации рабочего процесса, разработчик синхронизирует локальную копию репозитория на своей рабочей машине с сервером GitHub, после чего выделяет отдельную ветку для решения задачи.

В ветку «master» попадают только те изменения, что связанны с устранением ошибок, и не попадают изменения, связанные с добавлением новой функциональности. Текущее состояние ветки «Master» помечается соответствующим номеру и наименованию релиза. В дальнейшем данный тег может использоваться для идентификации нужной сборки при организации работы технической поддержки.

**Исправление ошибок**

Процесс исправления ошибок начинается в одном из следующих случаев:

* обнаружена ошибка в процессе тестирования;
* поступило обращение пользователя в службу технической поддержки.

В случае обнаружения ошибки на этапе тестирования, если тестирование проводилось в рамках работы над уже существующей задачей, информация об ошибке и условиях её воспроизведения указывается в комментариях к этой задаче. В остальных случаях в системе управления проектами и задачами «Bugtracker» создаётся новая задача с описанием ошибки и условий её воспроизведения. Если ошибка возникла в процессе работы над существующей задачей, разработчик сразу приступает к её устранению. В случае новой задачи в зависимости от критичности выявленной ошибки определяется, когда задача должна быть взята в работу.

С обращениями пользователя работает специалист отдела технической поддержки или специалист отдела тестирования. Он воспроизводит проблему по исходным данным, полученным от клиента. При необходимости отправляет запрос клиенту на уточнение данных или осуществляет удалённое подключение к клиентскому устройству для сбора необходимой информации. Далее описываются шаги по воспроизведению ошибки и задача назначается разработчикам в отдел разработки ПО.

Проблема анализируется разработчиками, после чего они вносят необходимые изменения в исходные коды ПО и совместно со специалистами отдела тестирования проводят верификацию изменений в соответствии с методикой организации работы.

**Выпуск релиза**

Выпуск основных релизов ПО Amirig, содержащих новую функциональность, осуществляется один раз в полгода. В промежутке между основными релизами могут при необходимости выпускаться промежуточные релизы, которые, как правило, включают исправления наиболее критичных ошибок функционирования ПО.

При подготовке релиза выполняют:

* полное функциональное тестирование с использованием виртуальных машин и целевых аппаратных платформ;
* нагрузочное тестирование;
* тестирование на стабильность;
* тестирование установки.

При наличии ошибок, выявленных в процессе тестирования, в исходные коды вносятся исправления. После успешного завершения тестирования и отладки выполняется слияние изменений в ветку «Master» и производится обновление репозитория бинарных пакетов для доступа клиентов.

**Техническая поддержка**

Техническая поддержка предусматривает выпуск обновлений ПО Amirig, включая исправления критических ошибок, а также работу с обращениями пользователей. Право на получение технической поддержки может быть приобретено при покупке программного обеспечения ПО Amirig и регулируется положениями дополнительного соглашения о технической поддержке, являющимся приложением к контракту поставки ПО Amirig.

Обновления ПО Amirig выпускаются в соответствии с планом релизов.

Работа с обращениями пользователей осуществляется

* посредством выдачи рекомендаций и технических консультаций по телефону и электронной почте;
* в форме предоставления обновлений ПО Amirig.

Услуги оказываются Исполнителем в рабочие дни согласно Производственному календарю РФ с 09-00 до 18-00 по местному времени.

В стандартную техническую поддержку входят следующие работы и услуги:

* 1. Оказание помощи в решении технических проблем, связанных с эксплуатацией ПО Amirig у пользователя, в том числе, предоставление ответов на вопросы, связанные с некорректной работой ПО.
  2. Обновление ПО Amirig.
  3. Рассмотрение и регистрация предложений пользователя по развитию или модификации функций, выполняемых ПО, которые могут быть учтены и реализованы в новой версии ПО Amirig.
  4. Оказание пользователю содействия в установке и настройке ПО Amirig при изменении конфигурации операционных систем и других базовых программных продуктов или технических средств пользователя.

В расширенную техническую поддержку входят услуги по устранению существенных замечаний и/или предложений пользователя к ПО, не связанных напрямую с наличием ошибок внутри ПО Amirig.

**Персонал, необходимый для обеспечения поддержки**

Коллектив разработчиков (программисты, аналитики, специалисты по сопровождению) обладают необходимым набором знаний для работы со всеми компонентами, входящими в состав ПО Amirig.

В штате ООО «ЭЙП ТЕКНОЛОДЖИ» числится 10 сотрудников основного персонала (IT-специалисты), 3 из которых занимаются технической поддержкой и эксплуатацией.

1. далее ПО (ПО Amirig) [↑](#footnote-ref-1)