

Программа для ЭВМ «Атом.Проект: Базовый релиз»

**Описание процессов, обеспечивающих поддержание жизненного цикла**

Листов: 21

Дата: 18.09.2023

Версия: 1.0

АННОТАЦИЯ

В настоящем документе приведено описание процессов, обеспечивающих поддержание жизненного цикла программы для ЭВМ «Атом.Проект: Базовый релиз» (далее – Программа).

В разделе «Общие сведения» приведены общие сведения о Программе, её назначении и функциональных характеристиках.

В разделе «Роли исполнителей» приведены роли, принимающие участие в процессах жизненного цикла Программы, с указанием их обязанностей.

В разделе «Стадии жизненного цикла Программы» описана модель жизненного цикла Программы, указаны цели, процессы, исполнители и результаты каждой из стадий.

Описание жизненного цикла приведено в соответствии с ГОСТ Р ИСО/МЭК 12207‑2010.

Документ подготовлен в соответствии с ГОСТ 19.103‑77 – в части наименования и обозначения, ГОСТ 19.106‑78 – в части оформления, ГОСТ 19.508‑79 – в части структуры и содержания.

СОДЕРЖАНИЕ

[1. Общие сведения 4](#_Toc145951482)

[1.1. Назначение Программы 4](#_Toc145951483)

[1.2. Возможности Программы 4](#_Toc145951484)

[1.3. Состав Программы 5](#_Toc145951485)

[2. Роли исполнителей 6](#_Toc145951486)

[3. Стадии жизненного цикла Программы 9](#_Toc145951487)

[3.1. Перечень стадий жизненного цикла Программы 9](#_Toc145951488)

[3.2. Стадия сбора и анализа потребностей 9](#_Toc145951489)

[3.3. Стадия проектирования архитектуры и разработки 10](#_Toc145951490)

[3.4. Стадия приёмки 14](#_Toc145951491)

[3.5. Стадия применения и сопровождения 16](#_Toc145951492)

[3.6. Стадия изъятия 19](#_Toc145951493)

[Перечень сокращений и условных обозначений 22](#_Toc145951494)

1. Общие сведения
   1. Назначение Программы

Программа для ЭВМ «Атом.Проект: Базовый релиз» (далее – Программа) — это программа, предназначенная для работы с задачами, создания графиков, дашбордов.

* 1. Состав Программы

Программа включает в себя следующие основные компоненты:

1. Ядро Программы – обеспечивает подключение к системам управления проектами, хранение пользовательских настроек графиков и дашбордов, отправку запросов в системы управления проектами, сбор и обработку отчетных данных, взаимодействие с пользовательским интерфесом Программы.
2. Интерфейс пользователя – веб-интерфейс, реализованный в виде одностраничного приложения (SPA), предназначенный для управления и мониторинга процессов конфигурирования рабочих станций.

С целью обеспечения безопасности информации при её обработке Программой используются следующие технологии и инструменты:

* HTTPS при обмене данными между компонентами Программы;
* аутентификация / авторизация в Программе на основе ролей;

1. Роли исполнителей
2. **Функциональный заказчик**

В обязанности функционального заказчика входят:

* формирование бизнес-требований к Программе;
* контроль соответствия результата заданию.

1. **Руководитель проекта**

В обязанности руководителя проекта входят:

* управление процессом разработки Программы;
* планирование и согласование сроков и ресурсов;
* контроль соответствия работ согласованному с функциональным заказчиком плану проекта;
* формирование и предоставление функциональному заказчику промежуточных и итоговых отчётов о ходе разработки Программы;
* принятие решения о выпуске релиза Программы.

1. **Ведущий программист**

В обязанности ведущего программиста входят:

* принятие ключевых решений по архитектуре Программы;
* подготовка состава задач на реализацию изменений в исходных кодах;
* определение требований к исполнителям;
* определение требований к технологиям производства и инструментам разработки;
* разработка и внесение изменений в проектную документацию.

1. **Программист**

В обязанности программиста входят:

* реализация функциональности подсистем или модулей Программы согласно заданию Ведущего программиста и в соответствии с планом, подготовленным руководителем проекта;
* первичный контроль качества и работоспособности разрабатываемых компонентов Программы;
* подготовка технического описания реализации функциональных возможностей разрабатываемых компонентов Программы;
* устранение ошибок ПО и исправление дефектов, обнаруженных в процессе тестирования;
* предоставление отчётов о состоянии выполнения заданий.

1. **Инженер тестирования**

В обязанности инженера тестирования входят:

* проверка работоспособности Программы в соответствии с техническими условиями;
* подготовка стендов для проверки работоспособности Программы в соответствии с набором сценариев использования;
* подготовка отчётов о составе и опасности обнаруженных в ПО дефектов;
* воспроизведение на стендах проблем, обнаруженных инженером технической поддержки;
* планирование сроков и ресурсов на выполнение тестирования ПО;
* предоставление отчётов о результатах тестирования Программы.

1. **Инженер поддержки процесса и инфраструктуры разработки ПО**

В обязанности инженера поддержки процесса и инфраструктуры разработки ПО входят:

* поддержка и развитие сборочной системы (сборка исполняемых модулей и библиотек, запуск различных утилит для подготовки дистрибутива продукта);
* развитие утилит контроля регламента разработки (качество кода, контроль переводов ресурсов, контроль информации о версии, контроль настроек проектов и т.п.);
* стандартизация окружения разработки;
* настройка инфраструктуры с учётом особенностей разрабатываемого ПО;
* поддержка разработки, основанной на отдельных ветках под каждую новую функцию продукта;
* создание шаблонов сборки для веток;
* выполнение регламента работы с итерационными ветками разработки.

1. **Аналитик**

В обязанности аналитика входят:

* согласование состава и характера изменений, вносимых в Программу, с функциональным заказчиком;
* формирование системных требований к продукту;
* контроль соответствия результата системным и бизнес-требованиям;
* принятие участия в решении запросов, переданных во вторую линию поддержки.

1. **Инженер технической поддержки**

В обязанности инженера технической поддержки входят:

* приём и маршрутизация заявок от пользователей Программы;
* консультация пользователей по возникающим вопросам, связанным с Программой;
* управление критическими инцидентами и оповещение пользователей;
* сопровождение и участие в обновлении установленного ПО;
* диагностика и устранение неисправностей;
* закрытие заявок на первом уровне технической поддержки.

1. **Технический писатель**

В обязанности технического писателя входят:

* разработка технической и пользовательской документации на Программу;
* документирование изменений, производимых в Программе;
* контроль документов проекта на соответствие принятому стандарту документирования.

1. Стадии жизненного цикла Программы
   1. Перечень стадий жизненного цикла Программы

Для Программы выбрана эволюционная стратегия разработки, представляющая собой многократный проход этапов разработки. Данная стратегия основана на частичном определении требований к разрабатываемому программному средству в начале процесса разработки. Требования постепенно уточняются в последовательных циклах разработки. Результат каждого цикла разработки представляет собой очередную поставляемую версию программного средства.

В процессе развития Программы используется спиральная модель жизненного цикла и релизный подход при обновлении.

Модель жизненного цикла Программы включает в себя следующие стадии:

1. стадия сбора и анализа потребностей;
2. стадия проектирования архитектуры и разработки;
3. стадия приёмки;
4. стадия применения и сопровождения;
5. стадия изъятия.
   1. Стадия сбора и анализа потребностей

В рамках данной стадии осуществляется взаимодействие с функциональным заказчиком, направленное на выявление системных и функциональных требований к Программе, потребностей в изменении Программы, разработке дополнительных подсистем, компонентов.

Схема стадии сбора и анализа потребностей приведена на рисунке Рисунок 1.



Рисунок 1 – Схема стадии сбора и анализа потребностей

На данной стадии применяются следующие технические процессы:

* процесс определения требований;
* процесс анализа требований.

Роли исполнителей, действующих на данной стадии:

* функциональный заказчик;
* аналитик;
* руководитель проекта;
* ведущий программист;
* технический писатель.

Результатом данной стадии является частное техническое задание на проектирование и разработку подсистемы (компонента Программы).

* 1. Стадия проектирования архитектуры и разработки

В рамках данной стадии:

* определяется архитектура Программы и её элементов;
* разрабатываются проектные решения;
* производятся, комплексируются, испытываются и оцениваются технические и программные средства и интерфейсы;
* разрабатывается программная документация.

На данной стадии применяются следующие технические процессы:

* процесс проектирования архитектуры;
* процесс реализации;
* процесс комплексирования Программы;
* процесс квалификационного тестирования Программы;
* процесс менеджмента документации программных средств.

Роли исполнителей, действующих на данной стадии:

* руководитель проекта;
* ведущий программист;
* программист;
* инженер тестирования;
* инженер поддержки процесса и инфраструктуры разработки ПО;
* технический писатель.

На данной стадии применяются следующие технические процессы:

Схема стадии проектирования архитектуры и разработки приведена на рисунке Рисунок 2.



Рисунок 2 – Схема стадии проектирования архитектуры и разработки

Комплексирование (сборка) Программы осуществляется из исходного кода. Исходный код хранится в системе управления версиями. Для формирования сборки Программы разработаны специализированные автоматизированные процедуры.

В результате работы инструментов сборки на выходе получаются:

* готовые к установке приложения/компоненты Программы;
* набор скриптов для модификации модели данных Программы.

На данной стадии осуществляется тестирование Программы в различных контурах:

* контур разработки – модульное тестирование;
* контур тестирования – интеграционное и системное тестирование;
* контур регрессионного тестирования – регрессионное тестирование;
* контур нагрузочного тестирования – нагрузочное тестирование.

Применяемые виды тестирования:

1. Модульное тестирование – изолированные испытания отдельных модулей Программы. Объектом испытания может служить отдельная функция, метод, процедура, модуль или программный объект. Если модуль взаимодействует с внешними системами, для тестирования применяются специализированные программные модули-«заглушки», имитирующие внешние системы.
2. Интеграционное тестирование – испытания, при которых модули Программы объединяются логически и тестируются как группа. Целью тестирования является выявление проблем взаимодействия отдельных компонентов Программы или взаимодействия Программы с внешними системами.
3. Регрессионное тестирование – испытания ранее протестированной Программы, позволяющие убедиться, что внесенные изменения не повлекли за собой появления дефектов в той части программы, которая не менялась.
4. Системное тестирование – испытания, направленные на подтверждение корректности поведения Программы в целом, подразумевающее сквозную проверку взаимодействия всех компонентов Программы и взаимодействия Программы с внешними системами.
5. Нагрузочное тестирование – испытание Программы в условиях прогнозируемой нормальной нагрузки и стрессовой нагрузки. Под величиной нагрузки понимается количество пользовательских запросов к Программе, которое она должна успевать обрабатывать, не превышая определенное исходными требованиями время отклика.
6. Юзабилити-тестирование (проверка эргономичности) – исследование, выполняемое с целью определения, удобен ли пользовательский интерфейс Программы для его предполагаемого применения. Проверка эргономичности – метод оценки удобства продукта в использовании, основанный на привлечении пользователей в качестве испытателей и суммировании полученных от них выводов.

Решения о переводе Программы на следующую фазу разработки или следующую стадию жизненного цикла принимаются в зависимости от результатов тестирования в соответствующих контурах:

* тестирование в контуре разработки – решение о возможности установки сборки в тестовый контур;
* тестирование в контуре тестирования – решение о возможности включения изменения в сборку для регрессионного тестирования и нагрузочного тестирования;
* тестирование в контурах регрессионного и нагрузочного тестирования – решение о начале квалификационного тестирования Программы (стадия приёмки).

Стадия завершается готовностью релиз‑кандидата Программы к предварительным испытаниям, готовностью программной документации в следующем составе:

1. «Формуляр»
2. «Описание применения»
3. «Описание языка»
4. «Руководство программиста»
5. «Руководство системного администратора»
6. «Руководство оператора»
7. «Руководство по техническому обслуживанию»

Результатом стадии разработки также могут являться требования, решения, оценки и прочие аналитические данные, предназначенные для использования на последующих стадиях.

* 1. Стадия приёмки

Стадия приёмки заключается в проведении испытаний Программы в единственном экземпляре или серийно (Рисунок 3). Стадия может включать в себя процессы улучшения и реконфигурации программного продукта.



Рисунок 3 – Схема стадии приёмки

В рамках данной стадии:

* разрабатывается и передаётся на утверждение функциональному заказчику Программа и методика испытаний Программы (ПМИ);
* подготовка объекта автоматизации, установка и настройка Программы;
* проводится приёмка Программы, включающая:

1. **Предварительные испытания:**

В ходе предварительных испытаний проверяется работоспособность и соответствие техническому заданию, устраняются выявленные неисправности и недостатки. Испытания должны включать системное тестирование и нагрузочное тестирование. После проведения предварительных испытаний Программа передается в опытную эксплуатацию.

1. **Опытную эксплуатацию:**

В ходе опытной эксплуатации выполняется работа с программным продуктом по назначению, собираются статистические данные о характеристиках и результатах функционирования, в том числе проводится юзабилити-тестирование. При необходимости осуществляется доработка программного продукта.

1. **Приёмочные испытания:**

Испытания проводятся после завершения опытной эксплуатации. По результатам приемочных испытаний принимается решение о выпуске релиза Программы (начале серийной эксплуатации Программы).

На данной стадии применяются следующие технические процессы:

* процесс инсталляции программных средств;
* процесс поддержки приёмки программных средств.

Роли исполнителей, действующих на данной стадии:

* функциональный заказчик
* руководитель проекта;
* ведущий программист;
* инженер тестирования;
* инженер поддержки процесса и инфраструктуры разработки ПО.

Результатом стадии является решение о выпуске релиза Программы (переход на стадию применения и сопровождения).

* 1. Стадия применения и сопровождения

Началом стадии применения Программы служит её установка и передача для применения по назначению в соответствии с моделью процесса поставки по ГОСТ Р ИСО/МЭК 12207‑2010. В рамках настоящего документа процесс поставки Программы не рассматривается.

Сопровождение Программы состоит в обеспечении техническим обслуживанием и сопровождением и другими видами поддержки функционирования и использования Программы в соответствии с согласованными условиями поставки Программы приобретающей стороне (Рисунок 4).



Рисунок 4 – Схема стадии применения и сопровождения

В рамках данной стадии:

* осуществляется настройка Программы в предназначенной для неё среде;
* осуществляется применение Программы в предназначенной для неё среде;
* обеспечивается содействие и консультации пользователям Программы;
* осуществляется контроль характеристик Программы, а также сбор и анализ данных об отклонениях, недостатках и отказах.
* определяются новые требования к Программе, её подсистемам и компонентам;
* производится анализ собранных данных об отклонениях, недостатках и отказах, анализ, уточнение требований приобретающей стороны и формирование решений;
* производятся модификации Программы;
* по мере необходимости обновляется связанная с изменениями системная и программная документация;
* обновленные компоненты Программы помещаются в среду приобретающей стороны;
* сведения о модификации Программы доводятся до всех затронутых обновлениями сторон.

На данной стадии применяются следующие технические процессы:

* процесс функционирования программных средств.
* процесс сопровождения программных средств.

Роли исполнителей, действующих на данной стадии:

* приобретающая сторона;
* поставщик;
* аналитик;
* ведущий программист;
* программист;
* инженер технической поддержки;
* технический писатель.

Техническая поддержка пользователей осуществляется через Центр поддержки пользователей. Все обращения в Центр поддержки пользователей регистрируются и классифицируются по типам (Таблица 1).

Таблица 1 – Типы обращений

|  |  |  |
| --- | --- | --- |
| № | Тип | Формат текста описания заявки |
| 1 | Инцидент (ошибка ПО, проблема смежных сервисов) | Несоответствие работы Программы утвержденным техническим заданиям и эксплуатационной документации, вывод сообщений об ошибках. |
| 2 | Запрос на изменение | Замечание/предложение по работе Программы (запрос на изменение/расширение функциональности) |
| 3 | Консультация | Просьба пользователя о пояснении работы с Программой |

Устранение ошибок в работе Программы производится путём изменения настроек среды выполнения Программы, модификации программного кода Программы без выпуска новой сборки либо с выпуском новой сборки, которая помещается в ИТ‑инфраструктуру приобретающей стороны. При необходимости корректируется документация на Программу.

На основании анализа обращений пользователей Программы, собранных данных о недостатках и отказах Программы функциональный заказчик может самостоятельно принять решение о модернизации Программы. Модернизация подразумевает развитие функциональных возможностей Программы, повышение удобства использования и администрирования. После модернизации производится выпуск нового релиза Программы (Рисунок 5). Для выполнения модернизации Программы инициируется переход на стадию проектирования архитектуры и разработки.

В случае заинтересованности в расширении функциональности Программы приобретающая сторона направляет поставщику новые требования к Программе. Требования анализируются поставщиком, далее согласовываются условия оказания услуг по доработке Программы, сроки и стоимость работ. Для выполнения доработки Программы инициируется переход на стадию проектирования архитектуры и разработки.

В случае принятия приобретающей стороной решения о завершении эксплуатации Программы инициируется переход на стадию изъятия.



Рисунок 5 – Схема процесса выпуска версий

* 1. Стадия изъятия

В соответствии с ГОСТ Р ИСО/МЭК 12207-2010 модель процесса прекращения применения программных средств предусматривает прекращение деятельности организации по поддержке функционирования и сопровождения или деактивирует, демонтирует и удаляет поврежденные программные продукты, отправляя их в финальное состояние и возвращая окружающую среду в приемлемые условия.

Прекращение применения Программы (Рисунок 6) означает деактивацию и удаление Программы из среды конкретного покупателя (приобретающей стороны).

Деятельность организации (подразделения) по поддержке функционирования и сопровождению может быть прекращена по решению обладателя исключительного права на программу для ЭВМ «Атом.Проект: Базовый релиз».



Рисунок 6 – Схема стадии изъятия

Причиной перевода Программы в данную стадию может служить замещение новой программой, катастрофический отказ, неэффективность дальнейшего применения.

На данной стадии применяются следующие технические процессы:

* процесс прекращения применения программных средств.

Роли исполнителей, действующих на данной стадии:

* приобретающая сторона;
* поставщик;
* функциональный заказчик;
* аналитик;
* инженер технической поддержки.

При планировании процесса определяется и документируется стратегия прекращения применения Программы. Разрабатывается и план прекращения активной поддержки работ, связанных с Программой. Запланированные действия включают в себя участие пользователей Программы.

План включает в себя:

* порядок прекращения полной или частичной поддержки через определенный период времени;
* порядок архивирования программного продукта и связанной с ним документации;
* определение сторон, ответственных за любые оставшиеся на будущее вопросы поддержки;
* регламент перехода к новому программному продукту (при необходимости);
* порядок доступа к копиям архива данных;
* порядок деактивации и удаления Программы из инфраструктуры приобретающей стороны.

Прекращение применения Программы осуществляется в соответствии с разработанным и утвержденным планом. Все заинтересованные стороны оповещаются о планах и действиях по выводу Программы из эксплуатации.

Вся связанная документация по разработке, журналы и коды помещаются в архивы. Используемые данные или данные, связанные с прекращением применения Программы, должны быть доступны в соответствии с требованиями законодательства, локальных нормативных актов, соглашений и условий поставки.

Перечень сокращений и условных обозначений

|  |  |
| --- | --- |
| АРМ | Автоматизированное рабочее место |
| БД | База данных |
| ИТ | Информационные технологии |
| ЛВМ | Локальная виртуальная машина |
| ОС | Операционная система |
| ПО | Программное обеспечение |
| Поставщик | Организация или лицо, которое вступает в соглашение с приобретающей стороной на поставку продукта или услуги |
| Приобретающая сторона | Правообладатель, который приобретает или получает продукт или услугу от поставщика |
| Разработчик | Организация, которая выполняет разработку задач (в том числе анализ требований, проектирование, приемочные испытания) в процессе жизненного цикла |
| Программа | «Атом.Проект: Базовый релиз» |
| СУБД | Система управления базами данных |