# Жизненный цикл и поддержка Архиватора

Документ описывает, как идентифицируют сборки Архиватора, проверяют изменения, выпускают пакеты и воспроизводят ошибки. Здесь же приведён контакт для обращений.

---

## Разработка и проверка изменений

Программа использует общее ядро обработки архивов на Go. Его вызывают настольные интерфейсы и мобильные оболочки. Для браузера ядро собирается в WebAssembly и работает в отдельном Web Worker.

### Учёт изменений

Исходные тексты, сценарии сборки и тесты находятся под управлением Git. Настольный выпуск идентифицируют номером версии и редакцией исходного кода. Результаты технических проверок и исследований ведутся вместе с проектом.

### Проверка ядра

Модульные и интеграционные тесты охватывают чтение и запись форматов, проверку целостности и обработку ошибок. Тестовые архивы позволяют повторить сценарий на фиксированных входных данных.

### Проверка интерфейсов

Для браузерной версии предусмотрены сквозные проверки открытия архива, предпросмотра, извлечения отдельной записи и получения выбранных записей в ZIP. Содержимое результата можно сопоставить с исходными байтами или выводом независимого распаковщика.

## Устранение неисправностей и развитие

1. **Воспроизведение.** Для локализации ошибки нужны версия, платформа, выполненная операция и ожидаемый результат. Для сбоя одного формата полезен минимальный тестовый архив, в котором повторяется ошибка.
2. **Локализация.** Когда операция доступна в нескольких интерфейсах, её можно повторить в CLI или в браузере. Проверка тех же данных в другом интерфейсе помогает определить, связан ли сбой с движком, отображением или работой с файлами операционной системы.
3. **Исправление и проверка.** Изменение можно проверить на воспроизводящем примере и на смежных сценариях. Для повторяющихся дефектов применяются регрессионные тесты. Для изменения распаковки проверяют имена, состав и данные сохранённых записей.
4. **Развитие.** Изменения форматов, интерфейсов и производительности проходят через тот же исходный код, тесты и сборку. Зафиксированную редакцию исходников можно связать с подготовленным выпуском.

## Выпуск и доставка обновлений

Сценарии сборки формируют пакеты под конкретные системы и архитектуры. Для выпуска на сайте формируется список SHA-256. Браузерная поставка включает страницу, рабочий поток и модуль WebAssembly, которые загружаются с сайта.

Пользователь может посмотреть номер настольной сборки в окне «О программе» или командой `archiver version`. На Android версию можно узнать в системных сведениях о приложении. Сравните свою версию с пометкой у загрузки и установите подходящий пакет. Проверка контрольной суммы и порядок обновления описаны в инструкции по эксплуатации.

# Компетенции для сопровождения

Ниже перечислены технические задачи и знания, которые требуются для их выполнения.

### Анализ обращений и тестирование

Умение воспроизвести операцию в указанной ОС, работать с архивными форматами, собрать минимальный пример и сверить результат с ожидаемыми файлами.

### Общее ядро и настольные приложения

Разработка на Go, структуры архивов, файловые системы, потоки ввода и вывода. Для графического интерфейса требуются знания Gio и механизмов интеграции с файловым менеджером ОС.

### Android и веб-версия

Для Android нужны знания Kotlin, Android SDK и системного доступа к документам. Для веб-версии — JavaScript, программные интерфейсы браузера, WebAssembly и Web Workers.

### Подготовка выпуска

Сборка под целевые системы и архитектуры, упаковка, проверка подписей и контрольных сумм, публикация совместимых файлов и пользовательских инструкций.

# Как обратиться по работе программы

Для связи используйте общий адрес НейроГейт support@neuralgate.ru. Укажите «Архиватор» и краткое описание проблемы в теме. Контакты НейроГейт.

- Укажите операционную систему и архитектуру, версию Архиватора или название и версию браузера.
- Опишите последовательность действий, ожидаемый и фактический результат. Добавьте точный текст ошибки и назовите формат и примерный размер архива.
- Если нужен пример, подготовьте маленький архив с искусственными данными. Для предложения об улучшении опишите задачу и результат, которого вы хотите добиться.