



АО "НТЦ ИТ РОСА"

**СИСТЕМА АВТОМАТИЗАЦИИ СБОРКИ И ПОДДЕРЖКИ ПАКЕТОВ,
РЕПОЗИТОРИЕВ И ДИСТРИБУТИВОВ ПРОДУКТОВ АВФ**

Описание жизненного цикла

Листов 14

2025

Инв. № подл.	
Подпись и дата	
Взам. инв. №.	
Инв. № дубл.	
Подпись и дата	

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММНОМ ОБЕСПЕЧЕНИИ.....	4
1.1 Языки программирования и технологии.....	4
1.2 Архитектура системы.....	4
1.3 Область применения ПО.....	4
2 УЧАСТНИКИ ЖИЗНЕННОГО ЦИКЛА ПО.....	6
3 МОДЕЛЬ ЖИЗНЕННОГО ЦИКЛА ПО.....	8
3.1 Выбранная модель жизненного цикла.....	8
3.2 Этапы жизненного цикла.....	8

ВВЕДЕНИЕ

Настоящий документ описывает жизненный цикл программного обеспечения "Система автоматизации сборки и поддержки пакетов, репозитория и дистрибутивов продуктов ABF" (Automatic Build Farm, далее в тексте ABF, Система).

ABF – это распределённая среда непрерывной разработки и сборки программного обеспечения, предназначенная для автоматизации процессов создания, тестирования и распространения дистрибутивов Linux и прикладных программ. Система обеспечивает полный цикл разработки, начиная от управления исходным кодом и заканчивая формированием готовых ISO-образов.

Работа с ABF может производиться через веб-интерфейс, через REST API, а также посредством консольного клиента ABF.

Настоящий документ подготовлен в соответствии с технологической инструкцией: "РОСА. Регламент формирования документации к программным продуктам" РСЮК.11001-01 90 01.

1 ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММНОМ ОБЕСПЕЧЕНИИ

Полное наименование программного обеспечения: Система автоматизации сборки и поддержки пакетов, репозитория и дистрибутивов продуктов ABF.

Актуальная версия программного продукта: 2.0.

Разработчик: АО "Научно-технический центр информационных технологий "РОСА". Россия, 124498, г. Москва, вн.тер.г. Муниципальный округ Старое Крюково, г. Зеленоград, пл. Шокина, д. 2, стр. 3 <https://rosa.ru>.

ABF – это специализированная система автоматизации сборки программного обеспечения для обеспечения полного цикла разработки, тестирования и публикации дистрибутивов и пакетов операционных систем семейства GNU/Linux.

Система представляет собой распределенную сборочную платформу с централизованным управлением, предназначенную для ведения репозитория, формирования сборочных заданий, автоматической сборки пакетов из исходных кодов, управления зависимостями, проведения тестирования и публикации результатов. ABF обеспечивает возможность построения как единичных пакетов, так и целостных дистрибутивов, включая выпуск релизов, формирование установочных образов (ISO) и сопровождение в течение всего жизненного цикла программного продукта.

1.1 Языки программирования и технологии

Система ABF разработана с использованием следующих технологий:

- языки программирования: Ruby, Bash;
- системы управления версиями: Git;
- контейнеризация: Docker;
- сборка пакетов: RPM, с использованием инструментов rpm-build, createrepo_c;
- интерфейсы взаимодействия: REST API;

1.2 Архитектура системы

Система ABF имеет модульную архитектуру, включающую следующие компоненты:

- веб-интерфейс для управления проектами и сборками;

- серверы сборки, обеспечивающие автоматическую компиляцию и тестирование пакетов;
- систему хранения артефактов сборки и репозиториев;
- интеграцию с системами контроля версий и CI/CD;

Такая архитектура обеспечивает масштабируемость, надежность и гибкость в управлении процессами разработки и сборки программного обеспечения.

1.3 Область применения ПО

Система ABF применяется для:

- автоматизации сборки пакетов и дистрибутивов Linux;
- управления жизненным циклом программных продуктов;
- обеспечения совместной работы команд разработчиков;
- интеграции с системами контроля версий и CI/CD.

Система подходит для использования как в коммерческих, так и в государственных проектах, требующих надежной и масштабируемой инфраструктуры для разработки и сопровождения программного обеспечения.

2 УЧАСТНИКИ ЖИЗНЕННОГО ЦИКЛА ПО

Жизненный цикл программного обеспечения ABF обеспечивается следующим составом участников:

1. Организация-разработчик

- Наименование: ОАО «НТЦ ИТ РОСА»
- Роль в жизненном цикле:
 - постановка задач разработки и техническое задание;
 - непосредственная реализация, сопровождение и развитие ПО;
 - проведение тестирования и выпуск релизов;
 - организация технической поддержки и публикация документации;

2. Команда архитекторов и разработчиков

- Роль:
 - проектирование архитектуры и модулей системы;
 - реализация функциональности с использованием Ruby, Bash, систем сборки RPM;
 - разработка внутренней и внешней документации;
 - обеспечение обратной совместимости между релизами;
- Особенности:
 - использование методологий DevOps, CI/CD;
 - контроль качества кода через систему обзоров и тестирование;

3. Тестировщики

- Роль:
 - проведение различных видов тестирования;
 - проверка сборок на соответствие требованиям к стабильности и безопасности;
 - верификация корректности работы компонентов при обновлениях;

4. Пользователи

- Категории пользователей:
 - внутренние команды компании-разработчика, использующие ABF для разработки дистрибутивов;
 - внешние организации, применяющие ABF как систему сборки собственных решений;
- Роль:
 - формулирование обратной связи, баг-репортов и пожеланий;
 - участие в тестировании новых релизов (бета-тестирование);

5. Сопровождающие и специалисты по эксплуатации

- Роль:
 - настройка серверной инфраструктуры и сборочных ферм;
 - мониторинг работоспособности среды ABF;

– управление резервным копированием и отказоустойчивостью.

3 МОДЕЛЬ ЖИЗНЕННОГО ЦИКЛА ПО

3.1 Выбранная модель жизненного цикла

Модель жизненного цикла отражает различные состояния изделия, начиная с момента возникновения необходимости в данном программном изделии и заканчивая моментом его полного выхода из употребления. Модель жизненного цикла — структура, содержащая процессы, действия и задачи, которые осуществляются в ходе разработки, функционирования и сопровождения программного изделия в течение всей жизни изделия, от определения требований до завершения его использования.

Модель жизненного цикла изделия, определяемая на основе государственных стандартов, включает процессы и стадии, обеспечивающие выработку требований, разработку, производство и эксплуатацию изделия. Важной особенностью является то, что жизненный цикл программного изделия — непрерывный, и результаты, получаемые на предыдущих стадиях, служат основой для последующих стадий.

Для ABF применяется инкрементальная (поэтапная) модель жизненного цикла с элементами непрерывной интеграции и доставки (CI/CD). Выбор данной модели обоснован требованиями к регулярному выпуску обновлений, поддержке гибкой архитектуры и высокой степени автоматизации.

3.2 Этапы жизненного цикла

Модель жизненного цикла базируется на перечне работ (фаз жизненного цикла). Ниже приведены основные фазы жизненного цикла изделия:

1. Инициация и постановка требований

Основными целями этапа являются:

- формулировка потребности в новом изделии;
- выбор направления и определение экономической обоснованности проектирования изделия.

Системный анализ изделия начинается с описания и анализа функционирования рассматриваемого объекта в соответствии с требованиями (целями), которые предъявляются к нему. В результате этого этапа выявляются недостатки существующего изделия, на основе которых формулируется потребность в совершенствовании изделия, и ставится задача определения экономически обоснованной необходимости (создается технико-экономическое обоснование

проекта разработки изделия). После определения этой потребности возникает проблема выбора направлений совершенствования объекта на основе выбора программно-технических средств. Результаты оформляются в виде технического задания на проект, в котором отражаются технические условия и требования к изделию, а также ограничения на ресурсы проектирования.

2. Проектирование

Данный этап предполагает:

- определения требований;
- оценки входных и выходных данных;
- установления ответственных за разработку;
- проектирование функциональной архитектуры изделия, отражающей структуру выполняемых функций;
- проектирование системной архитектуры изделия (состава обеспечивающих подсистем);
- управления изменениями проекта и разработки.

Формирование функциональной архитектуры, которая представляет собой совокупность функциональных подсистем и связей между ними, является наиболее ответственным и важным этапом с точки зрения качества всей последующей разработки изделия.

Построение системной архитектуры на основе функциональной предполагает определение элементов и модулей информационного, технического, программного обеспечения и других обеспечивающих подсистем, связей по информации и управлению между выделенными элементами.

3. Реализация

Разработка и внедрение программного обеспечения АВФ включает полный цикл создания программных компонентов, написания эксплуатационной и технической документации, а также организацию процесса внедрения и эксплуатации в инфраструктуре заказчика или разработчика.

Процесс внедрения ПО подразделяется на опытный и промышленный этапы:

- опытное внедрение предполагает проверку корректности функционирования компонентов системы, выявление и устранение ошибок на уровне отдельных элементов и взаимодействия между ними. Выполняется тестирование в условиях, приближенных к продуктивной среде, включая настройку сборочной инфраструктуры и публикацию пробных пакетов.
- промышленная эксплуатация включает развёртывание системы в полном объёме, контроль выполнения заявленных функций и соответствия требованиям, установленным на стадии технического задания и системного анализа. Результаты тестирования подтверждаются в рабочих условиях, обеспечивается регулярное обновление и сопровождение ПО.

Разработка системы АВФ включает следующие ключевые процессы:

- проектирование архитектуры системы и компонентов (веб-интерфейсов, серверов сборки, интеграции с git-репозиториями, механизмов публикации пакетов);
 - реализация прикладных и вспомогательных модулей с использованием языков Ruby, Bash и средств автоматизации CI/CD;
 - подготовка и сборка RPM-пакетов, их публикация в репозиториях дистрибутивов;
 - организация сборки дистрибутивов на основе предварительно собранных пакетов;
 - итеративное уточнение требований и приоритетов совместно с внутренними и внешними заказчиками;
 - проведение автоматизированного и ручного тестирования (юнит-, интеграционные и регрессионные тесты);
 - подготовка технической и пользовательской документации;
- Результаты этапа разработки ABF включают:

- исходные тексты всех компонентов, размещённые в системе контроля версий (Git);
- бинарные пакеты и ISO-образы, сформированные в процессе сборки;
- комплект технической документации, включая инструкции пользователя, описание архитектуры, процедуры установки и сопровождения, а также документы для оценки соответствия требованиям в области информационной безопасности.

4. Сборка и упаковка

Сборка и упаковка программного обеспечения ABF выполняются с использованием RPM-технологий и средств автоматизированной сборки.

Каждому компоненту соответствует файл спецификации .spec, содержащий инструкции для компиляции, установки и упаковки. Процесс сборки осуществляется в изолированной среде, что обеспечивает воспроизводимость и независимость результата от конфигурации сборочного узла.

Изменения в исходных файлах и спецификациях автоматически инициируют сборку на удалённых серверах через систему непрерывной интеграции (CI). В рамках этого процесса производится:

- сборка пакетов;
- проверка целостности и корректности установки;
- автоматическое формирование бинарных пакетов и репозитория.

После успешной сборки пакеты могут быть включены в состав установочных образов. Создание ISO-дистрибутивов осуществляется с использованием специализированных сборочных инструментов, обеспечивающих формирование загрузочной среды, добавление документации и локалей.

Все результаты сборки сопровождаются логами, контрольными суммами и, при необходимости, цифровой подписью. Артефакты доступны для загрузки и анализа через веб-интерфейс системы ABF.

5. Тестирование

Проведение тестирования является обязательным этапом жизненного цикла ПО ABF перед передачей новой версии потребителям и включением её в официальный репозиторий. Тестирование осуществляется ответственными специалистами в соответствии с утверждённым регламентом.

Для проведения испытаний формируется тестовый стенд, определяются задачи и цели тестирования. На этом этапе проверяется корректность работы сборочного механизма, стабильность выполнения ключевых операций и соответствие программного обеспечения предъявляемым требованиям.

В рамках тестирования и отладки выполняются следующие действия:

- сборка и развёртывание дистрибутивов;
- выполнение функционального и регрессионного тестирования;
- анализ уязвимостей и потенциальных рисков безопасности;
- устранение выявленных дефектов;
- загрузка эталонных версий дистрибутивов и исходных кодов в официальный репозиторий;
- обновление и актуализация сопроводительной документации.

По завершении работ сотрудник, ответственный за выпуск новой версии, формирует заключение о её качестве, с указанием уровня устранения ошибок и общего состояния стабильности. Заключение согласуется с:

- руководителем проекта;
- главным инженером.

6. Эксплуатация и сопровождение

Производитель гарантирует соответствие программного обеспечения ABF заявленным характеристикам при условии соблюдения пользователями установленных требований к условиям хранения, транспортирования, установки и эксплуатации, зафиксированных в эксплуатационной документации.

В период эксплуатации обеспечивается сопровождение и техническая поддержка пользователей, включая:

- помощь в установке и конфигурации программного обеспечения;
- устранение возможных сбоев и ошибок;
- консультации по вопросам использования;
- при необходимости – обучение технических специалистов заказчика.

Техническая поддержка направлена на поддержание стабильной и эффективной работы системы, своевременное выявление и устранение ошибок, а также на

адаптацию программного обеспечения к изменяющимся требованиям. В рамках поддержки выполняются следующие виды работ:

- исправление дефектов, не обнаруженных на предыдущих этапах жизненного цикла;
- оптимизация функционирования системы при различных условиях эксплуатации;
- обновление компонентов программного обеспечения (в том числе библиотек и зависимостей), с учётом развития технологий и инфраструктурных изменений.

Поддержка пользователей и администраторов осуществляется по каналам связи (телефон, электронная почта, внутренние трекары и системы обратной связи) и охватывает все ключевые аспекты жизненного цикла: установку, обновление, администрирование и эксплуатацию системы.

Важной характеристикой жизненного цикла ABF является его цикличность: системный анализ → разработка → тестирование → производство → сопровождение → повторный системный анализ. При первом прохождении этапа "Разработка" формируется базовая архитектура системы. При последующих итерациях осуществляется модификация проекта с целью актуализации, улучшения функциональности и адаптации к новым условиям применения.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

ПО	– Программное обеспечение
ТЗ	– Техническое задание
ABF	– Automated Build Farm (автоматизированная сборочная ферма)
CI/CD	– Continuous Integration / Continuous Delivery (непрерывная интеграция / непрерывная доставка)
RPM	– RPM Package Manager (система управления пакетами RPM)
API	– Application Programming Interface (программный интерфейс приложения)
ISO	– Файловый образ диска в формате ISO

Лист регистрации изменений									
Номера листов (страниц)					Всего листов в докум.	№ документа	Входящий № сопровод. докум. и дата	Подпись	Дата
Изм	измененных	замененных	новых	аннулированных					