AXIOM JDK

Axiom Trusted Images 25 LTS Процессы жизненного цикла и поддержки

Axiom Trusted Images | Октябрь 2026

Copyright © 2019-2026 Все права защищены АО "АКСИОМ" (АКСИОМ)

Программное обеспечение АКСИОМ содержит программное обеспечение с открытым исходным кодом. Дополнительная информация о коде сторонних разработчиков доступна на сайте https://axiomjdk.ru/third_party_licenses. Для дополнительной информации о том, как получить копию исходного кода, можно обратиться по адресу info@axiomjdk.ru.

ДАННАЯ ИНФОРМАЦИЯ МОЖЕТ ИЗМЕНЯТЬСЯ БЕЗ ПРЕДВАРИТЕЛЬНОГО УВЕДОМЛЕНИЯ. АКСИОМ ПРЕДОСТАВЛЯЕТ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ "КАК ЕСТЬ" БЕЗ КАКИХ-ЛИБО ГАРАНТИЙ, АКСИОМ ПРЯМО ОТКАЗЫВАЕТСЯ ОТ ВСЕХ ПОДРАЗУМЕВАЕМЫХ ГАРАНТИЙ, ВКЛЮЧАЯ, НО НЕ ОГРАНИЧИВАЯСЬ ПОДРАЗУМЕВАЕМЫМИ ГАРАНТИЯМИ ТОВАРНОЙ ПРИГОДНОСТИ И ПРИГОДНОСТИ ДЛЯ ОПРЕДЕЛЕННОЙ ЦЕЛИ.

АКСИОМ НИ ПРИ КАКИХ ОБСТОЯТЕЛЬСТВАХ НЕ НЕСЕТ ОТВЕТСТВЕННОСТИ ЗА ЛЮБЫЕ КОСВЕННЫЕ, СЛУЧАЙНЫЕ, СПЕЦИАЛЬНЫЕ, ШТРАФНЫЕ ИЛИ КОСВЕННЫЕ УБЫТКИ, ИЛИ УБЫТКИ ОТ ПОТЕРИ ПРИБЫЛИ, ДОХОДА, ДАННЫХ ИЛИ ИСПОЛЬЗОВАНИЯ ДАННЫХ, ПОНЕСЕННЫЕ ВАМИ ИЛИ ЛЮБОЙ ТРЕТЬЕЙ СТОРОНОЙ, БУДЬ ТО В РЕЗУЛЬТАТЕ ДЕЙСТВИЯ ДОГОВОРА ИЛИ ДЕЛИКТА, ДАЖЕ ЕСЛИ АКСИОМ БЫЛО ПРЕДУПРЕЖДЕНО О ВОЗМОЖНОСТИ ТАКИХ УБЫТКОВ.

Использование любого программного продукта АКСИОМ регулируется соответствующим лицензионным соглашением, которое никоим образом не изменяется условиями данного уведомления. Программные продукты и фирменные наименования: Axiom JDK, Axiom JDK Pro, Axiom Runtime Container Pro, Axiom Linux, Libercat, Libercat Certified и АКСИОМ принадлежат АКСИОМ и их использование допускается только с разрешения правообладателя.

Товарный знак Linux® используется в соответствии с сублицензией от Linux Foundation, эксклюзивного лицензиата Линуса Торвальдса, владельца знака на всемирной основе. Java и OpenJDK являются товарными знаками или зарегистрированными товарными знаками компании Oracle и/или ее аффилированных лиц. Другие торговые марки являются собственностью их соответствующих владельцев и используются только в целях идентификации.

Содержание

1. Введение 6

2. Жизненный цикл продукта Axiom Trusted Images 25 LTS 7

2.1. Проектирование и дизайн 7

2.2. Разработка и тестирование 7

2.3. Стабилизация 8

2.4. Выпуск 8

2.5. Поддержка 9

3. Сборка продукта 10

4. Описание процесса и регламентов поддержки 11

4.1. Команда 11

4.2. Регламенты работы 11

4.3. Информационные системы 11

5. Приложение A. Разработка 13

5.1. Обеспечение качества 13

5.2. Регрессионное тестирование 13
5.3. Тестирование новой функциональности 13
5.4. Тестирование нефункциональных требований 13
5.5. Исследовательское тестирование 14
5.6. Жизненный цикл дефекта 14
5.7. Разработка 14
5.8. Написание исходного кода 14
5.9. Рецензирование кода 15
5.10. Внесение изменений в систему контроля версий 15
5.11. Сборка компонент продукта 15
5.12. Документация 15

6. Приложение Б. Использующиеся информационные системы 16

6.1. Системы контроля версий 16
6.2. GitLab 16
6.3. База знаний GitLab 16
6.4. Система сборки 17
6.5. Файловые хранилища 17
6.6. Планировщик автоматических тестов 17
6.7. Система управления тестированием 17

6.8. Хранилище результатов прохождений автоматических тестов 18

7. Приложение В. Инструментальные средства разработки 19

7.1. Сводная информация по инструментальным средствам разработки 19

1. Введение

Цель документа – описание процессов, обеспечивающих поддержание жизненного цикла программного изделия, в том числе устранение неисправностей, выявленных в ходе эксплуатации программы, совершенствование программного изделия, а также информацию о персонале, необходимом для обеспечения такой поддержки.

2. Жизненный цикл продукта Axiom Trusted Images 25 LTS

Жизненный цикл проекта по выпуску новой версии программного продукта имеет несколько фаз. Далее перечисляются фазы, их характеристики и необходимые роли для исполнения.

2.1. Проектирование и дизайн

Проектирование системы проводится на основе результатов формирования требований.

Проектирование включает в себя определение архитектуры программной системы, поддержка и обеспечение ее функциональных и нефункциональных требований, внешних условий функционирования, требования к программным и информационным компонентам, а также условия эксплуатации. Определяются основные риски проекта и планы ослабления рисков.

Разрабатывается функциональная спецификация ПО, выбирается архитектура системы, определяется наиболее подходящая СУБД, проектируются структуры хранения данных, оговариваются требования к аппаратному обеспечению, определяется набор организационных мероприятий, которые необходимы для внедрения ПО, а также перечень документов, регламентирующих его использование.

В проработке участвуют программные и продуктовые менеджеры, руководитель разработки и архитектор.

2.2. Разработка и тестирование

На фазе разработки происходит уточнение дизайна реализации функциональности, разбиение задач на подзадачи, и уточняется оценка трудозатрат. Задачи распределяются между разработчиками.

На данной фазе строятся прототипы программного продукта, осуществляется физическая реализация структур данных, разрабатываются программные коды, выполняется отладочное тестирование, создается техническая документация, в которой описаны особенности реализации функционала и принятые ограничения. В результате этапа кодирования появляется рабочая версия продукта.

На протяжении фазы разработки еженедельно производится контроль выполнения планов работ и уточнение проектных рисков. При этом прогнозируется срок окончания работ на основании

текущих данных. Также актуализируется информация о текущих обнаруженных дефектах и операционных проблемах, связанных с наличием данных дефектов.

Фаза завершается по окончанию разработки всего функционала, входящего в состав продукта.

2.3. Стабилизация

На этой фазе команды разработки и тестирования занимаются исправлением и валидацией исправленных найденных дефектов в рамках текущего проекта. Определением того, что подлежит исправлению, занимаются руководители проектов.

На протяжении фазы стабилизации еженедельно производится контроль выполнения планов работ и уточнение проектных рисков. Для определения актуального состояния используется информация о дефектах.

На фазе стабилизации в системе контроля версий создается отдельная ветвь для выпускаемой версии продукта. В ней производятся только те изменения, которые направлены на исправление дефектов, подлежащих исправлению именно в данной версии продукта, и исключаются изменения, не связанные с такими исправлениями.

2.4. Выпуск

Выпуску подлежит версия продукта, собранная ранее системой сборки на основании исходного кода, содержащего изменения для всей необходимой функциональности, исправления найденных дефектов и корректно прошедшая ручное и автоматическое тестирование продукта. Сборка производится из ветки, созданной для выпускаемой версии продукта на фазе стабилизации. В системе контроля версий создается специальная пометка, позволяющая точно восстановить исходный код продукта выпускаемой версии. Описание шагов сборки приведено ниже в отдельном разделе Сборка.

При выпуске продукта производятся действия, указанные в документе «контрольный список выпуска продукта» (release checklist). Действия включают:

- Отчет по качеству выпускаемого функционала;
- Проверку и подготовку информации о выпускаемой версии продукта;
- Подготовку к началу поддержки версии продукта;
- Публикацию и фиксацию (создание чексумм (checksum) для бинарных файлов/артефактов) распространяемых артефактов версии продукта.

2.5. Поддержка

Выпущенная версия продукта находится на поддержке определенное время (по умолчанию 3 года). В течение этого периода производится исправление обнаруженных в эксплуатации дефектов. При обнаружении дефект фиксируется в системе отслеживания обращений. После исправления производится либо выпуск планового обновления, содержащего исправление дефекта, либо, при необходимости, дополнительно выпускается критическое исправление.

3. Сборка продукта

Сборка продукта производится системой сборки по запросу от заинтересованных лиц или запускается автоматически при наличии изменений в системе контроля версий или по расписанию. Артефактам сборки автоматически назначается версия. В случае успешной сборки продукт проходит первичное тестирование (так называемый smoke testing – минимальный набор тестов на явные ошибки), по успешному прохождению которого поступает на вход системе исполнения автоматических тестов.

С успешно прошедшей первичное тестирование версией продукта ассоциируются новые изменения в системе контроля версий (по сравнению с предыдущей версией продукта), и ассоциируются дефекты, исправлению которых были посвящены изменения. Такие дефекты поступают на верификацию системе исполнения автоматических тестов и инженеру по обеспечению качества, чтобы на этом этапе подтвердить, что автоматические тесты проверяют исправление конкретного дефекта.

Собранные файлы продукта хранятся в системе хранения артефактов сборки в соответствии с политикой хранения, определяемой на основании характеристик качества данной версии.

Сборка продукта производится автоматически системой по сборке «Jenkins».

При обновлении пакетов в репозитории, если эти пакеты входят в состав образа контейнера, такой образ собирается заново со свежими пакетами. Сборка производится выполнением задачи на Jenkins. Во время выполнения сборки вызываются необходимые make-файлы (makefiles, make, cmake, GNU Autotools), и на выходе создаются готовые установочные файлы.

Единым хранилищем исходных кодов проекта является локальный git-сервер.

Единым хранилищем собранных модулей программного обеспечения является репозиторий на сайте https://download.axiomjdk.ru/.

4. Описание процесса и регламентов поддержки

4.1. Команда

Техническая поддержка обеспечивается командой поддержки. Команда поддержки состоит из руководителя группы и специалистов технической поддержки/разработчиков в количестве не менее 5 человек.

4.2. Регламенты работы

Заявки на техническую поддержку принимаются по телефону и через портал поддержки клиентов.

Заявки на техническую поддержку принимаются для LTS-версий продуктов (то есть версий, имеющих долгосрочную поддержку), и обработка заявок производится в объеме, декларированном для продукта в момент времени принятия заявки.

Для срочных заявок контакт с клиентом осуществляется по телефону оператором поддержки в кратчайшие сроки для выяснения деталей нештатной ситуации и поиска возможных способов устранения. В случае, если оператор не имеет возможности устранить неисправность, заявка передается в следующую линию поддержки.

Команда поддержки следующей линии производит обработку заявок высокой технической сложности. Команда представляет способ исправления нештатной ситуации, и, в случае если ситуация возникла из-за ранее неизвестного дефекта в обслуживаемом продукте, заводит дефект в системе отслеживания обращений, после чего обработка дефекта передается в команду разработки продукта. Исправление дефекта в таком случае производится в рамках плановых выпусков пакетов поддержки продукта.

В целях поддержания высокого уровня технической поддержки, производится мониторинг достаточности ресурсов для технической поддержки.

4.3. Информационные системы

Система отслеживания обращений на портале поддержки клиентов предназначена для регистрации заявок на техническое обслуживание продуктов через форму регистрации

обращения.

5. Приложение А. Разработка

5.1. Обеспечение качества

Обеспечение качества продукта производится комплексом мер и действий, перечисленных ниже. За обеспечение качества продукта отвечают разработчики и группа по обеспечению качества.

5.2. Регрессионное тестирование

Регрессионное тестирование отвечает за обнаружение ошибок, внесенных в ранее разработанную функциональность. В основном производится автоматически. За разработку автоматических регрессионных тестов отвечает команда разработчиков.

Запуск регрессионного тестирования производится как автоматически, так и по требованию инженера по обеспечению качества.

5.3. Тестирование новой функциональности

По мере готовности новой функциональности производится передача соответствующего модуля в команду обеспечения качества. Команда разрабатывает план тестирования функциональности на основании технического описание реализации и устных рекомендаций разработчиков, ответственных за функциональность. В ходе разработки автоматических тестов инженеры по обеспечению качества исправляют обнаруженные дефекты в информационной системе управления жизненным циклом дефекта (см. «Жизненный цикл дефекта»). По завершению разработки автоматического теста он включается в набор регрессионных тестов.

Разработка автоматических тестов, производимая как при доработке регрессионных тестов, так и тестов новой функциональности, состоит из тех же действий, что и разработка исходного кода продукта (см. «Разработка»).

5.4. Тестирование нефункциональных требований

Тестированием нефункциональных требований занимается специально выделенная для этого команда. Тестирование производится автоматически. Задачей команды является постановка задач инфраструктуре тестирования (в постановку задачи входит определение конфигурации тестовых стендов, на которых производится тестирование и набор измеряемых параметров производительности), анализ результатов прохождения тестов (включая автоматический анализ)

и поддержка исходного кода автоматических тестов в состоянии, пригодном для использования.

5.5. Исследовательское тестирование

Производится вручную командой обеспечения качества и предназначено для обнаружения дефектов в тестах, не включенных в автоматический план тестирования.

5.6. Жизненный цикл дефекта

Дефект в продукте может быть обнаружен как вручную инженером по обеспечению качества, так и системой автоматического тестирования. При обнаружении заводится сущность типа «дефект» в системе управления жизненным циклом дефекта gitlab. Вручную, или же автоматически, в дефекте описываются шаги его воспроизведения. Таким образом, разработчик, отвечающий за исправление дефекта, имеет максимальную информацию об условиях его возникновения. Ответственным за исправление дефекта в большинстве случаев назначается программист, отвечающий за компонент.

Дефект может быть исправлен в рамках выпуска данного продукта или по согласованию с ответственными лицами отложен на более поздний срок.

После завершения работы по исправлению дефекта (см. «Разработка») разработчик, отвечающий за компонент, в котором был обнаружен дефект, инициирует сборку версии продукта на основании исходного кода, содержащего исправление. В случае успеха этой операции для обнаруженных автоматически дефектов задача проверки исправления дефектов передается в автоматическое тестирование. В данном случае повторяется выполнение теста, и в случае успеха прохождения, дефект фиксируется как исправленный в данной версии. Для дефектов, обнаруженных в ходе ручного испытания, задача по верификации дефекта назначается на инженера по обеспечению качества и проводится также вручную.

5.7. Разработка

К разработке относятся действия, порождающие программный продукт. За разработку программного кода и создание на его основе программного продукта отвечает отдел по разработке.

5.8. Написание исходного кода

Написание исходного кода производится разработчиком, отвечающим за выполнение соответствующей задачи по разработке или исправлению дефекта.

5.9. Рецензирование кода

В процессе рецензирования кода контролируется стандарт кодирования и качество написанного кода с точки зрения затрат на его дальнейшую поддержку. Контролируется также проверка алгоритмических проблем, проблем исполнения программного кода в многопоточном, много-процессном и распределенном окружении, проверка проблемы потребления памяти, производительности и прочие существенные аспекты разработки исходного кода. Процессы и средства рецензирования кода различаются в компонентных командах. Это обусловлено различиями в компонентах (внутренняя разработка, внутренняя разработка компонента с открытым кодом, внешний компонент с открытым кодом) и наличием особых требований по качеству проведения рецензирования в командах. Среди различий можно отметить: использование специального инструментария для проведения рецензирования кода или рецензирования, производимого в списке рассылки почты; способ определения участников рецензирования и их ролей. Для рецензирования через список рассылки используются в том числе и автоматические средства, проверяющие, что изменение успешно компилируется и, будучи проинтегрированным, не приведет к ошибкам прохождения автоматических тестовых планов компонент.

5.10. Внесение изменений в систему контроля версий

Внесение кода в систему контроля версий производится либо автором кода, либо лицом, отвечающим за поддержку кодовой базы в зависимости от компонента. Изменение в системе контроля версий сопровождается комментарием, подробно описывающим суть и необходимость исправления. Также изменение может быть ассоциировано с дефектом, исправлению которого это изменение посвящено, или задачей, в рамках выполнения которой изменение было произведено.

5.11. Сборка компонент продукта

Сборка компонента производится по запросу от лица, ответственного за компонент. Артефактам сборки автоматически назначается версия. В случае успеха сборки бинарные артефакты проходят функциональное тестирование, состав которого определяется разработчиками компонента. В случае успеха прохода автоматического функционального тестирования новая версия компонента публикуется в хранилище артефактов сборки.

5.12. Документация

Документация разрабатывается техническими писателями по задачам на разработку документации. Контролем качества занимается команда обеспечения качества.

6. Приложение Б. Использующиеся информационные системы

6.1. Системы контроля версий

Отвечают за хранение исходного кода проектов и контроль изменений. Отдельно выделены репозитории для истории изменения закрытого и открытого исходного кода.

6.2. GitLab

Хранение исходных текстов проекта осуществляется в системе управления версиями Git-репозитории, в которой также доступна инспекция исходного кода ПО.

Обеспечивается ведение нескольких веток одного проекта как одним разработчиком, так и разными, в целях поддержки нескольких версий дистрибутива, а также нескольких дистрибутивов.

Система доступна по адресу http://gitlab.int.axiomjdk.ru/. Для совершения каких-либо действий, необходимо аутентифицироваться в системе с использованием доменной аутентификации.

Git-репозиторий представляет собой каталог файловой системы, в котором находятся: файлы конфигурации репозитория; файлы журналов, хранящие операции, выполняемые над репозиторием; индекс, описывающий расположение файлов; хранилище, содержащее собственно файлы. Когда ядро обрабатывает команду изменения, оно создает в хранилище новые файлы, соответствующие новым состояниям измененных файлов. Никакие операции не изменяют содержимого уже существующих в хранилище файлов.

6.3. База знаний GitLab

Хранит описание архитектуры, принятых проектных решений, стандарты кодирования, соглашения и прочую информацию, подлежащую к распространению внутри подразделения.

Планирование работ и управленческий учет осуществляется в системе GitLab – http://gitlab.int.axiomjdk.ru. GitLab – серверное веб-приложение для управления проектами и задачами с гибкой ролевой системой доступа от компании GitLab Inc. Для совершения каких-либо действий, необходимо аутентифицироваться в системе с использованием доменной аутентификации.

В системе заводятся проект (проекты) и подпроекты, в каждом из которых ведутся свои задачи, обеспечивая базу знаний проекта. В ней могут содержаться документы, замечания, описание механизмов, структуры, варианты использования продукта и т. п.

Общие возможности GitLab:

- ведение нескольких проектов;
- система доступа, основанная на ролях;
- система отслеживания ошибок;
- система оповещения об изменениях;
- база знаний для каждого проекта;
- интеграция с системами управления версиями исходного кода.

6.4. Система сборки

Система сборки отвечает за сборку компонент и продукта, интегрирован с системами контроля версий и планировщиком автоматических тестов.

6.5. Файловые хранилища

Файловые хранилища представляют сервис по хранению файлов с требуемыми характеристиками по доступности и отказоустойчивости. Файловые хранилища разделены по назначению, в том числе - файловые хранилища для артефактов сборки, образов операционных систем, результатов испытаний и прочих артефактов.

6.6. Планировщик автоматических тестов

Отвечает за планирование выполнения автоматических тестов на массиве серверов, предназначенных для выполнения автоматических тестов. При планировании учитываются результаты последних прохождений тестовых планов и информация о версиях продукта, в которых доступны исправления для ранее заведенных дефектов. Обеспечивает автоматическое заведение дефектов при обнаружении ошибок в ходе выполнения автоматических тестов.

6.7. Система управления тестированием

Система представляет автоматизацию учета для проведения ручных тестов.

6.8. Хранилище результатов прохождений автоматических тестов

Хранилище содержит результаты прохождения автоматических тестов, предоставляет отчетность по результатам.

7. Приложение В. Инструментальные средства разработки

7.1. Сводная информация по инструментальным средствам разработки

Таблица 1. Сводная информация по инструментальным средствам производства

Средства	Продукт	Версия	Разработчик/документация
Система планирования работ и управленческого учета	GitLab	15.8.4	https://about.gitlab.com/
Система учета запросов и ошибок	GitLab	15.8.4	https://about.gitlab.com/
Система хранения документов проекта	GitLab	15.8.4	https://about.gitlab.com/
Система хранения исходных текстов проекта	Git	2.30	Некоммерческий проект с открытым исходным кодом
Средства сборки продуктов	Сборочные скрипты		Разработчик АО «АКСИОМ»

Средства	Продукт	Версия	Разработчик/документация
	Выделенный сервер сборки	Linux OEL 6, Windows 2012 R2 x64	https://microsoft.com
	Jenkins	2.289	https://www.jenkins.io/
Средства разработки	Apache Maven	3 и выше	Некоммерческий проект с открытым исходным кодом
	Gradle	2.6 и выше	Некоммерческий проект с открытым исходным кодом
	Apache Ant	1.9 и выше	Некоммерческий проект с открытым исходным кодом
Средства разработки документации	OpenOffice	4.1	Некоммерческий проект с открытым исходным кодом
Средства тестирования	Скрипты автоматизации тестирования		Разработчик АО «АКСИОМ»
	llvm	7, 9	Проект с открытым исходным кодом
	Вспомогательное ПО собственной разработки		Разработчик АО «АКСИОМ»
	Svace	3.2.0	https://www.ispras.ru/technologies/svace/

AXIOM JDK

Axiom Trusted Images 25 LTS
Процессы жизненного цикла и поддержки