

ОПИСАНИЕ ЖИЗНЕННОГО ЦИКЛА ПО ФЕНИКС СИСТЕМЫ ТЕХНИЧЕСКОГО ДОКУМЕНТООБОРОТА

ФЕНИКС

Автор:	ООО «Цифровая Эпоха»
Дата создания:	14.07.2023
Дата обновления:	08.09.2023
Версия:	1.2

ОГЛАВЛЕНИЕ

1	ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПО	3
2	ДОКУМЕНТИРОВАНИЕ ПО	3
3	ОБУЧЕНИЕ ПЕРСОНАЛА, ТРЕБОВАНИЯ К КВАЛИФИКАЦИИ	3
4	ПОДДЕРЖКА И ВЫПУСК НОВЫХ ВЕРСИЙ.....	4
5	ПОДХОДЫ К ТЕСТИРОВАНИЮ СИСТЕМЫ	7
5.1	Автоматическое тестирование.....	7
5.2	Ручное тестирование	7
6	ФАКТИЧЕСКИЙ АДРЕС РАЗМЕЩЕНИЯ ИНФРАСТРУКТУРЫ	8
7	ФАКТИЧЕСКИЙ АДРЕС РАЗМЕЩЕНИЯ КОМАНДЫ РАЗРАБОТКИ И СЛУЖБЫ ПОДДЕРЖКИ.....	8

1 Проектирование и разработка ПО

Разработка проходила по **итерационной модели**, где всегда имеется возможность вернуться на любой предыдущий этап разработки ПО для уточнений требований и исправления компонентов.



2 Документирование ПО

Применялись следующие виды документирования:

- Документирование требований (функциональной спецификация)
- Документирование архитектурных решений (ТРП)
- Документирование системного API
- Документирование кода - методов, классов, функций и их свойств/аргументов (java doc)
- Документирование сценариев автоматического тестирования
- Документирование сценариев пользовательского взаимодействия с системой.

3 Обучение персонала, требования к квалификации

Обучение персонала проходит по программе подготовки для работы на проекте согласно выбранной программе подготовки кадров. Программы обучения определяются квалификацией персонала и опытом работы с системами технического документооборота, и имеют длительностью от одного до трех месяцев. В создании продукта участвовали разработчики, прошедшие квалификационное обучение по Documentum и Capital Projects, имеющие высшее инженерное образование и трудовой стаж от 5 лет в программировании

на языке высшего уровня Java, JavaScript. Архитекторы и аналитики со стажем работы над продуктами СТДО от 10 до 15 лет.

4 Поддержка и выпуск новых версий

Для реализации процесса поддержки, процесса выпуска новых версий ПО ФЕНИКС используются сервисы:

- Для хранения и управления репозиториями Git, а также реализации CI/CD используется Gitlab, развернутый on-premise, в облаке VK Cloud Solutions, в тенанте, принадлежащем ООО «Цифровая Эпоха»
- Для хранения и публикации docker-образов микросервисов ПО ФЕНИКС используется репозиторий Nexus, развернутый on-premise, в облаке VK Cloud Solutions, в тенанте, принадлежащем ООО «Цифровая Эпоха»
- Для непрерывного анализа и измерения качества программного кода используется решение SonarQube, развернутое on-premise, в облаке VK Cloud Solutions, в тенанте, принадлежащем ООО «Цифровая Эпоха»
- Для тестирования и проверки функционала/скриптов развертывания используются dev/test-среды, реализованные в виде Kubernetes-кластеров, развернутых on-premise, в облаке VK Cloud Solutions, в тенанте, принадлежащем ООО «Цифровая Эпоха»

Для осуществления поддержки и выпуска новых версий реализована схема публикации образов микросервисов/компонентов ПО ФЕНИКС. Схема публикации отображена на Рис. 1.

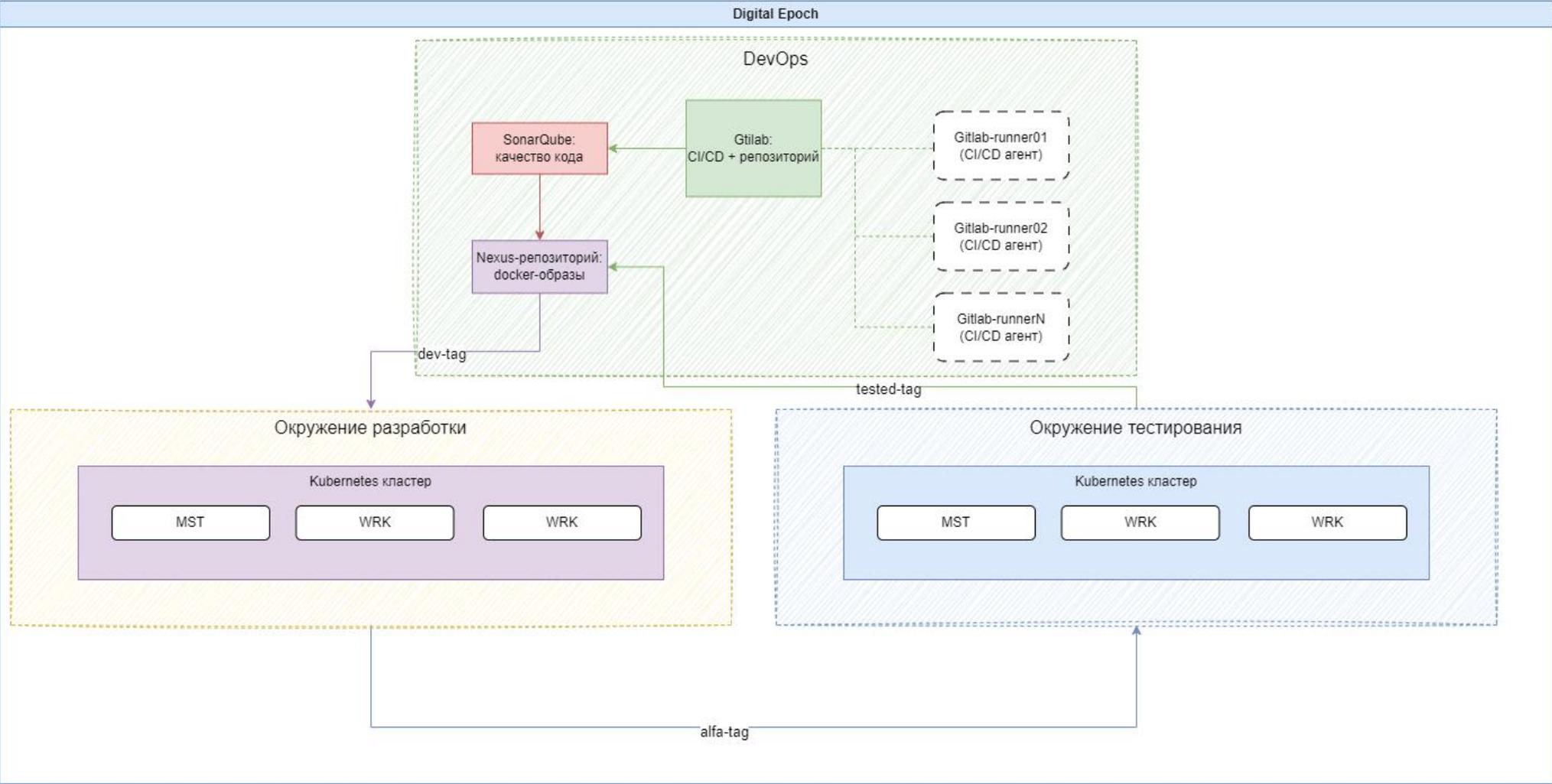


Рис. 1. Низкоуровневая схема публикации сборки микросервиса/компонента ПО Феникс

1. Каждая мажорная/минорная публикуемая версия кода микросервиса/компонента хранится в репозитории в отдельном бранче
2. Каждая мажорная/минорная публикуемая версия кода подвергается анализу и измерению качества программного кода, а также проверкам на уязвимости в коде с использованием стандартных шаблонов
3. Сборка микросервиса/компонента после успешного прохождения ci/cd-пайплайна хранится в виде docker-образа в Nexus-репозитории и доставляется на dev-окружение для первичного тестирования функционала
4. После успешного первичного тестирования сборке назначается тег "alfa". Данная сборка устанавливается на окружение тестирования, где производится полное тестирования функционала микросервиса/компонента в соответствии с программой и методикой испытаний
5. После успешного альфа-тестирования сборке назначается тег "tested"
6. Образ с тегом "tested" повторно подвергается тестированию запуска и валидации docker-контейнера из образа и после этого образу присваивается тег "stable"
7. Образ с тегом данный образ является стабильным и может быть доставлен на целевое окружение
8. В случае исправления ошибок или внесения изменений в функционал существующей мажорной версии микросервиса/компонента ПО ФЕНИКС выпускается минорная версия, которая проходит весь цикл проверок (пп 1-7).

5 Подходы к тестированию системы

5.1 Автоматическое тестирование

Автоматическое тестирование реализовано с помощью инструментария Gitlab CI/CD, Gradle и jmeter. Каждая задача тестирования в пайплайне выполняется изолированно, в отдельном, временном контейнере. Выполняемые тесты делятся на категории:

- Интеграционные и медленные тесты. Интеграционные и медленные тесты представляют собой набор самописных тестов и коротких проверок, направленных на выявление ошибок взаимодействия компонентов между собой, а также на уровне инфраструктуры. Интеграционные тесты выполняются при каждом последующем коммите в main/dev/demo-ветках репозиториях любого из микросервисов/компонент ПО ФЕНИКС. Медленные тесты выполняются по расписанию, каждый день в main/dev/demo-ветках репозиториях каждого из микросервисов/компонент ПО ФЕНИКС
- Сценарные тесты и тесты производительности. Сценарные тесты и тесты производительности представляют собой набор самописных тестов службы шлюза NGCP в формате .jmx, которые запускаются по триггеру и по расписанию в инструменте для проведения нагрузочного тестирования Jmeter. Выполнение jmeter-тестов происходит в рамках этапа "jmeter-test" cicd-пайплайна в main/dev/demo-ветках репозитория службы "gateway-service". Jmeter-тесты запускаются изолированно, в отдельном, временном docker-контейнере в рамках выполнения конкретной задачи этапа "jmeter-test" cicd-пайплайна в main/dev/demo-ветках репозитория службы "gateway-service".

5.2 Ручное тестирование

Для проверки и выявления несоответствий в интерфейсе программы производится ручное тестирование. Ручное тестирование выполняется специалистами по регулярно обновляемой, в соответствии с добавляемым/изменяемым функционалом, программе и методике испытаний.

6 Фактический адрес размещения инфраструктуры

Инфраструктура разработки ФЕНИКС включает в себя:

- Репозиторий Gitlab
- Репозиторий Nexus
- CI/CD на базе Gitlab и gitlab-runner
- Платформа для непрерывного анализа программного кода SonarQube
- Dev/test и прочие окружения для ручного тестирования и проверки функционала развернутого ПО ФЕНИКС

Инфраструктура разработки ФЕНИКС развернута в Tier 3 облаке VK Cloud Solutions (<https://mcs.mail.ru/>), в зоне доступности GZ1, в тенанте, принадлежащем ООО «Цифровая Эпоха». Фактический адрес размещения инфраструктуры зоны доступности GZ1: дата-центр Goznak. Адрес: г. Москва, проспект Мира, 105, стр. 6.

7 Фактический адрес размещения команды разработки и службы поддержки

Команда разработки и технической поддержки располагается в офисе ООО «Цифровая Эпоха» по адресу

199034, г. Санкт-Петербург,

вн.тер.г. Муниципальный округ № 7,

Линия 17-я В.О., д. 4-6 Литера Е,

Офис Помещение 1Н, офис Т-415-4