

A photograph of a majestic mountain range with snow-capped peaks, bathed in the warm, golden light of a sunset or sunrise. The sky is a soft, hazy orange. The foreground shows the dark, shadowed slopes of the mountains.

Безопасная разработка цифровых продуктов: вызовы и перспективы

Основные вызовы в безопасной разработке в финтехе сегодня

Безопасность как часть ДНК
всей разработки

Open-source vs closed
source

Проверка внешних
библиотек и OSS на
уязвимости

Нехватка
квалифицированных кадров

Влияние на TTM

Низкий уровень развития
профильных сообществ

Основные вызовы в безопасной разработке в финтехе сегодня

Безопасность как часть ДНК
всей разработки

Open-source vs closed
source

Проверка внешних
библиотек и OSS на
уязвимости

Нехватка
квалифицированных кадров

Влияние на TTM

Низкий уровень развития
профильных сообществ

Больше вопросов = правильная архитектура

Какие последствия компрометации?

Работает с сетью?
С интернет?

Open source или бинарник?

Взаимодействует с пользователями?

Важная цель?

Обрабатывает внешние данные?

Доверенный разработчик?

Взаимодействует с СКЗИ?

Использует API или работает само?

Легко изолировать?

Был аудит/сертификация?

Есть подходящие средства анализа?

Обрабатывает защищаемую информацию?

Доверенный канал получения?

Уже применяется в компании?

Есть встроенные средства безопасности?

Есть требования регулятора?

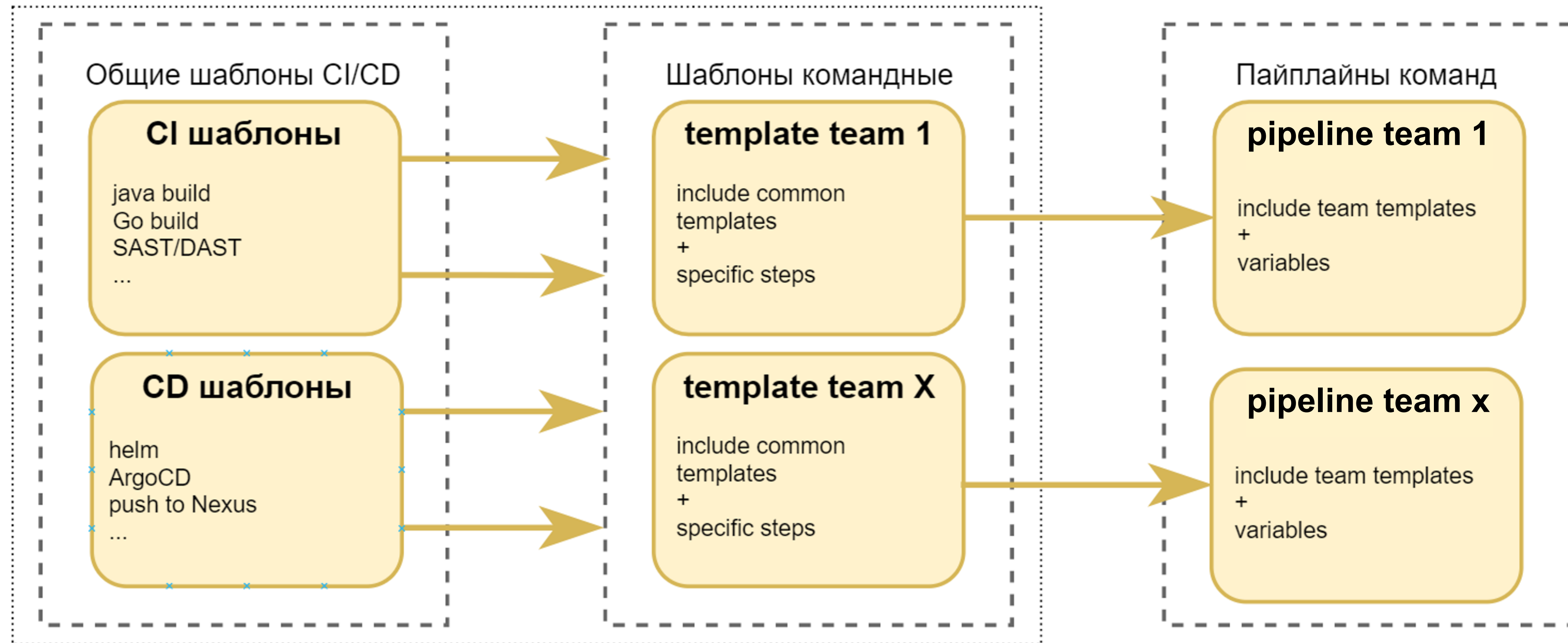
Регулярно обновляется?

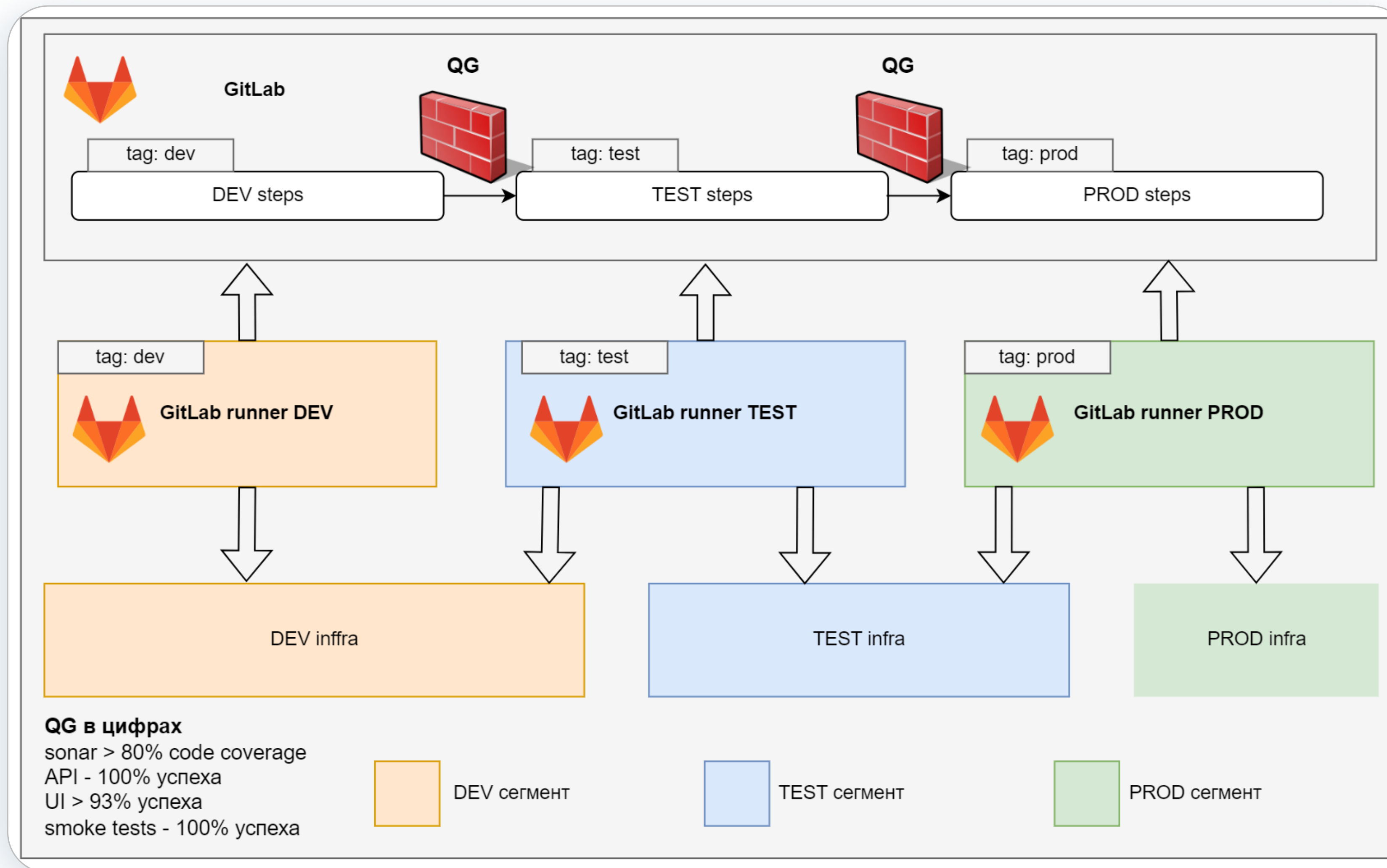
Есть аналоги или замены?

Много зависимостей?

Нельзя сделать универсальную методику

Единый репозиторий шаблонов





QGs/SGs (quality and security gates) - проверки и тесты на различных этапах конвейера. Такие как SAST, юнит, UI, API, регрессионные, контрактные, ручные, интеграционные и т.д. Несоответствие значения QGs/SGs эталонному блокирует конвейер до исправления ошибок.

Что входит в SGs?

SCM и контроль версий



1. Единый инстанс хранения исходного кода
2. Использование GitOps-ориентированных инструментов (пр. ArgoCD)
3. Запуск процессов проверки на MR
4. Использование практик автодокументирования

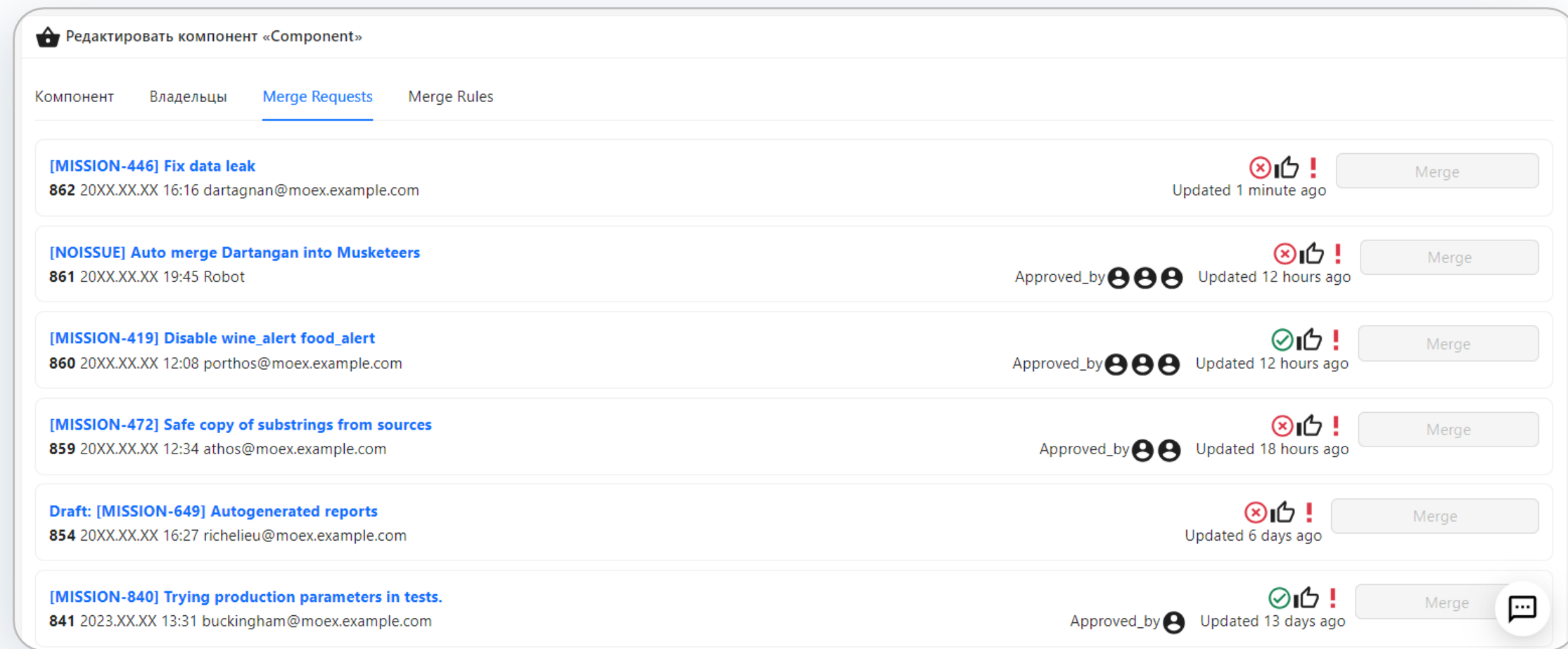
Сравнение полигонов « qa | prod »

Deployment Service Helms Resources Replicas

Введите имя ресурса... Все

Ресурс	qa	prod	Статус
service-a	Images в namespaces qa • qa-nexus.moex.example/docker/service-a:0.13.0	Images в namespaces prod • prod-nexus.moex.example/docker/service-a:0.13.0	OK
service-b	Images в namespaces qa • qa-nexus.moex.example/docker/service-b:0.28.0	Images в namespaces prod • prod-nexus.moex.example/docker/service-b:0.28.0	OK

Что входит в SGs? Сборка и CI



Robot @robot · 3 weeks ago Developer

Правило	Количество апруверов	Апрувнули	Могут апрувнуть	Правило выполнено?
Два за всех	1 / 2	porthos@moex.example.com	[athos@moex.example.com aramis@moex.example.com dartagnan@moex.example.com]	✗

1. Разделение окружений (dev/test/prod)
2. Инструменты SAST с четкими метриками прохождения для команд (~~default~~)
3. Функционал MR-approvers в DOP
4. Инструменты генерации SBOM и SCA-анализ
5. Подпись артефактов

Что входит в SGs?

Хранение артефактов и библиотек

1. Миграция артефактов только через изолированные контуры Dev -> Test -> Prod
2. Разделение инстансов хранения артефактов по уровням безопасности
3. Процесс переноса во внутренний контур внешних библиотек
4. Мониторинг внешних репозиториев

Как это выглядит в DOP

Выберите формат
npm

Выберите продукт
Platform

Цель использования
Платформенные сервисы Vue 3 последней версии

Доступ в сеть
-

Дополнительная информация, ссылки, wiki
-

Полигон использования
DEV

Источник списка пакетов
PACKAGE_LIST

Список пакетов с версиями
vue@3.3.4

Nexus of destination
nexus.example.com


Target nexus repo
npm-example-hosted

Quarantine nexus proxy repo
npm-example-hosted-proxy

менеджер пакетов
yarn

опции npm audit fix [--force]
NO_AUDIT_FIX

Создать



Как это выглядит в DOP

Статус: Progress ✎ Отменить проверку

Формат	npm	JIRA	NPM-MISSION-009	Прокси репозиторий	npm-example-hosted-proxy
Целевой нексус	nexus.example.com	Целевой репозиторий	npm-example-hosted	Источник пакетов	PACKAGE_LIST
Окружение	PROD	Описание	Продукт: Platform - Платформенные сервисы Vue 3 последней версии		

Список пакетов

✎

менеджер пакетов опции npm audit fix [--force]

Please select ▼

Please select ▼

+ Создать процесс

i Обратите внимание, что некоторые проверки проходят в несколько этапов и требуют подтверждения этапов пользователем!

1 Загрузка пакетов в карантинный репозиторий >

2 Загрузка результатов проверки

✔ 1 Completed 2023.09.06 11:19

pipeline ✔

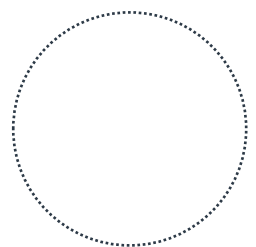
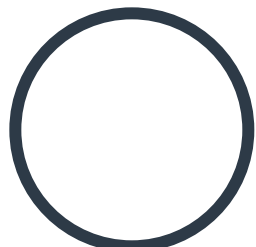
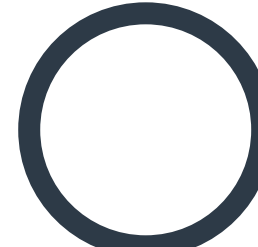
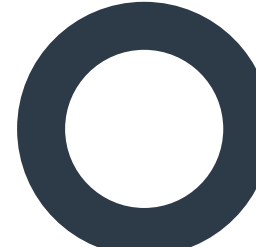
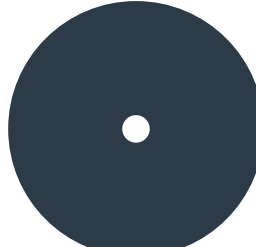
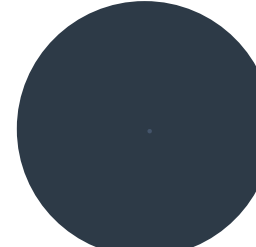
NO_AUDIT_FIXyampetrovskayaee

✎

11

Open-source тоже разный

Высокие риски

Модель	Полный open-source	Skinny	Thin	Lean	Think	Proprietary
Визуализация						
Состав	100% OSS 0% закрытое ПО	~90% OSS ~10% закрытое ПО	~70% OSS ~30% закрытое ПО	~50% OSS ~50% закрытое ПО	~10% OSS ~90% закрытое ПО	0% OSS 100% закрытое ПО
Модель доставки	On-prem	On-prem	On-prem / cloud	On-prem / cloud	On-prem / cloud	On-prem / cloud
Уровень контроля кода	Сверхмаксимальный	Максимальный	Средний	Низкий	Минимальный	Сверхминимальный



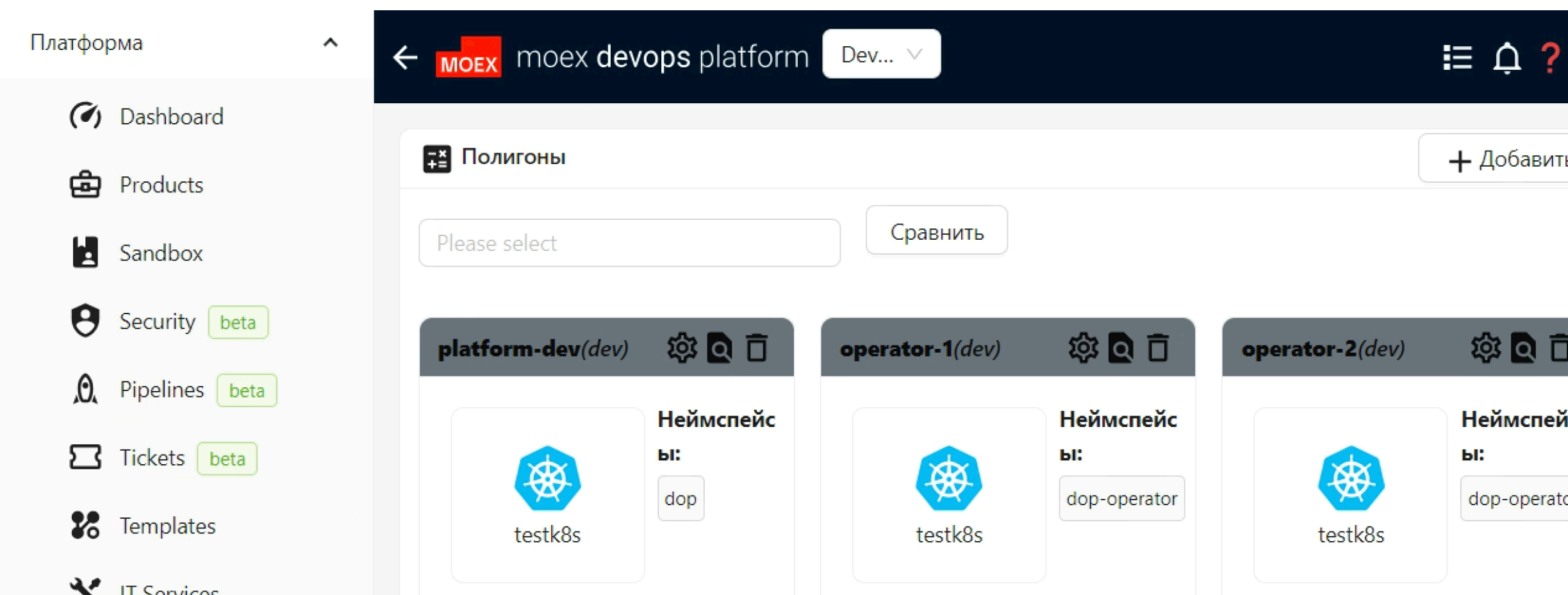
Что входит в SGs?

Тестирование и анализ

1. Система управление требованиями и тестами
2. Линтеры и SAST-ы начиная с ПК разработчика, а не только в CI/CD
3. Fuzzing тестирование
4. Тестирование изолированных окружений в песочнице
5. CDD или разработка на базе контрактов

Что входит в SGs?

Доставка, CD и мониторинг



1. Контроль целостности конфигураций и YAMЛиков
2. Управление окружениями на базе MR в deploy-maps
3. Конфигурация \neq секреты: стендо-зависимые переменные отдельно, чувствительные данные отдельно
4. Мониторинг аномалий и изменений в окружениях
5. Обеспечение полной видимости и observability всех контуров



**Довериться
конвейеру**



**Постоянно
проверять
гипотезы
относительно
devsecops**

Основные вызовы в безопасной разработке в финтехе сегодня

Безопасность как часть ДНК
всей разработки

Open-source vs closed
source

Проверка внешних
библиотек и OSS на
уязвимости

Нехватка
квалифицированных кадров

Влияние на TTM

Низкий уровень развития
профильных сообществ

Команды и культура

Внутреннее обучение и митапы

1. ИБ + ИТ = ❤️
2. Проведение аудита и анализа
3. Внутренние курсы для разработчиков и инженеров
4. Внутренние сообщества по DevOps и DevSecOps: единая точка правды
5. Выступления внешних спикеров во внутренних митапах
6. Создание понятных how-to

Команды и культура ИТ-сообщества

1. Создание сообщества FinDevSecOps: community по безопасной разработке в финсекторе
2. Создание совместных проектов и how-to
3. Унифицированная имплементация решений и практик
4. Шаринг знаний с помощью митапов



Выводы

- SGs не должны мешать команде
- Централизованный CI/CD-flow и gitops – наше все!
- Чем больше вопрос относительно дальнейшего использования open-source компонента, тем ниже риски связанные с ним
- Внедрил SG – определи метрики!
- Компетенции devsecops должны быть распределены на всех участников команды
- ИБ + ИТ = ❤️: безопасность как часть ДНК всего цикла разработки, security by design

DEVSECOPS
НАЧИНАЕТСЯ
ЗДЕСЬ