

Документ подписан простой электронной подписью
Информация о владельце: **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ**
ФИО: Порохня Андрей Алексеевич **РОССИЙСКОЙ ФЕДЕРАЦИИ**
Должность: И.о. директора института перспективной инженерии
Дата подписания: 19.06.2026 14:50:30 **Федеральное государственное автономное образовательное учреждение**
Уникальный программный ключ: **высшего образования**
990bea3aeec48c23bd8699e48288f8d1960c600 **«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

УТВЕРЖДАЮ
И.о. директора института
перспективной инженерии канд. техн.
наук, доцент Порохня А.А.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ
ПРОДВИНУТАЯ РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Направление подготовки	09.04.04 Программная инженерия
Направленность (профиль)	Разработка, мониторинг и управление жизненным циклом инженерных систем
Год начала обучения	2026
Форма обучения	очная
Реализуется в семестре	2

Введение

Назначение: ФОС предназначен для объективной оценки уровня сформированности компетенций.

ФОС является приложением к программе дисциплины «Продвинутая разработка программного обеспечения».

Разработчик Е.Н. Новикова, к.ф.-м.н., доцент, заведующий межинститутской базовой кафедрой департамента цифровых, робототехнических систем и электроники.

Проведена экспертиза ФОС.

Председатель Новикова Е.Н., межинститутской базовой кафедры департамента цифровых, робототехнических систем и электроники института перспективной инженерии

Члены комиссии:

Азаров И.В., и.о. директора департамента цифровых, робототехнических систем и электроники института перспективной инженерии

Николаев Е. И., доцент департамента цифровых, робототехнических систем и электроники института перспективной инженерии

Представитель организации-работодателя:

Руководитель группы разработки А.А. Шипулин, ООО «Стилсофт».

Экспертное заключение: ФОС по дисциплине позволяет оценить уровень сформированности компетенций. Приступить к апробации.

Срок действия ФОС определяется сроком реализации образовательной программы.

1. Описание критериев оценивания компетенции на различных этапах их формирования, описание шкал оценивания

Уровни сформированности компетенций, индикаторов	Дескрипторы			
	Минимальный уровень не достигнут (Неудовлетворительно) 2 балла	Минимальный уровень (удовлетворительно) 3 балла	Средний уровень (хорошо) 4 балла	Высокий уровень (отлично) 5 баллов
ПК-1- Способен проектировать, реализовывать, оптимизировать и поддерживать сквозные конвейеры непрерывной интеграции, поставки и развертывания (CI/CD) для программного обеспечения и встроенных систем, интегрируя практики автоматизированного тестирования, проверки безопасности, управления артефактами и конфигурациями				
ИД-1 ПК-1. Проектирует архитектуру CI/CD пайплайна с учётом требований к тестированию, безопасности и развертыванию для различных типов приложений (монолитные, микросервисные, встраиваемые системы)	Не может описать структуру CI/CD пайплайна, не понимает назначения этапов, не различает типы приложений.	Описывает базовую структуру пайплайна (build → test → deploy) для одного типа приложения, но допускает ошибки в выборе этапов.	Проектирует пайплайн для 2-3 типов приложений, учитывает этапы тестирования и безопасности, но допускает неточности в последовательности и этапов.	Самостоятельно проектирует оптимальную архитектуру CI/CD пайплайна для любых типов приложений, обосновывает выбор этапов и их последовательность, учитывая требования к качеству и безопасности.
ИД-2 ПК-1. Реализует пайплайн непрерывной интеграции, включающий этапы линтинга, автоматизированного тестирования (unit, интеграционные), статического анализа безопасности (SAST) и измерения покрытия кода	Не может настроить ни один из этапов, пайплайн отсутствует или неработоспособен.	Реализует 2 из 4 обязательных этапов (например, unit-тесты и линтинг), но SAST и покрытие кода отсутствуют или настроены неверно.	Реализует 3 из 4 этапов, все работают корректно, но есть замечания по конфигурации (неоптимальные пороги, отсутствие блокировки при падении).	Реализует все 4 этапа, настроены пороговые значения (Quality Gate), сборка автоматически блокируется при нарушении требований, пайплайн задокументирован.
ИД-3 ПК-1. Настраивает автоматическую сборку, версионирование (SemVer) и публикацию артефактов (библиотеки, Docker-образы) в бинарный	Не умеет собирать артефакты, версионирование отсутствует, публикация не настроена.	Собирает артефакт вручную или с ошибками, версионирование не автоматизировано (версия задаётся вручную).	Настроена автоматическая сборка и публикация артефакта, версионирование частично автоматизировано (например, только по тегам).	Полностью автоматизирована сборка, семантическое версионирование (на основе git tag/commit), публикация в репозиторий с разными стратегиями для

репозиторий				веток (snapshot/release)
ИД-4 ПК-1 Интегрирует управление конфигурациями и секретами в пайплайн, обеспечивая безопасную подстановку переменных окружения для различных сред (dev/staging/prod)	Не понимает разницы между конфигурациями и секретами, хранит секреты в коде.	Использует переменные CI/CD для хранения секретов, но конфигурации разных сред не разделены (только одно окружение).	Разделяет конфигурации для 2-3 сред через переменные или файлы, секреты хранит безопасно (в CI/CD variables).	Полностью внедрён подход Config as Code, настроены конфигурации для всех сред, интегрирован HashiCorp Vault для динамических секретов, маскирование секретов в логах обеспечено.
ИД-5 ПК-1 Оптимизирует производительность пайплайна за счёт параллелизации этапов, кэширования зависимостей и использования асинхронных/многопоточных подходов	Не применяет никаких методов оптимизации, пайплайн выполняется последовательно и медленно.	Использует один метод оптимизации (например, кэширование зависимостей), но параллелизация и асинхронность отсутствуют	Применяет 2 из 3 методов оптимизации (например, параллелизация тестов и кэширование), но есть неоптимальные настройки.	Применяет все методы оптимизации: параллельные джобы, кэширование, асинхронные/многопоточные скрипты, обосновывает выбор и демонстрирует ускорение пайплайна.
ИД-6 ПК-1. Реализует стратегии развертывания (Blue-Green, Canary, Rolling Update) и автоматический откат при сбоях на основе health-проверок	Не знает стратегий развертывания, деплой выполняется вручную или с остановкой сервиса.	Реализует одну простейшую стратегию (Rolling Update) без автоматического отката.	Реализует 2 стратегии (например, Rolling Update и Blue-Green), но откат выполняется вручную или health-проверки настроены не полностью	Реализует 2+ стратегии (Blue-Green и Canary), настроены health-проверки и автоматический откат при регистрации 5xx ошибок или превышении задержки, стратегии переключаются через параметры пайплайна.
ИД-7 ПК-1. Внедряет GitOps подход с использованием ArgoCD для синхронизации состояния кластера с репозиторием манифестов	Не знает GitOps, ArgoCD не установлен, синхронизация отсутствует	ArgoCD установлен, репозиторий манифестов подключён, но синхронизация выполняется вручную	ArgoCD настроен с автоматической синхронизацией (auto-sync), но обновление тега образа в манифестах выполняется вручную	Полностью внедрён GitOps: ArgoCD с auto-sync, CI/CD автоматически обновляет тег образа в репозитории манифестов, синхронизация происходит без задержек, self-healing включён.
ИД-8 ПК-1. Настраивает наблюдаемость пайплайна (метрики, логи,	Не настраивает мониторинг, проверки SLI/SLO отсутствуют.	Prometheus установлен и собирает метрики, но дашборда и алертов нет,	Настроены Prometheus и Grafana, дашборд создан, но проверка SLI/SLO	Полностью настроена наблюдаемость: метрики экспортируются,

алерты) и проверку SLI/SLO после деплоя		SLI/SLO не проверяются.	в пайплайне не автоматизирована.	Prometheus собирает, Grafana визуализирует, в CD-пайплайн добавлен этап проверки SLI с автоматическим откатом при нарушении SLO, настроены алерты.
---	--	-------------------------	----------------------------------	--

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕРКИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Номер задания	Правильный ответ	Содержание вопроса	Компетенция
1	В	Что означает аббревиатура CI/CD в современной разработке ПО? A. Computer Integration / Computer Design B. Continuous Integration / Continuous Delivery C. Code Implementation / Code Deployment D. Continuous Improvement / Continuous Development	ПК-1
2	А	Какой из перечисленных инструментов является российский CI/CD платформой? A. SourceCraft B. Jenkins C. CircleCI D. Travis CI	ПК-1
3	С	Что такое Quality Gate в SonarQube? A. Шлюз безопасности для доступа к репозиторию B. Механизм шифрования артефактов C. Набор пороговых метрик, при нарушении которых сборка блокируется D. Инструмент для деплоя в Kubernetes	ПК-1
4	True	Верно ли, что покрытие кода 100% гарантирует отсутствие ошибок в программе? (True/False)	ПК-1
5	В	Какой инструмент используется для статического анализа безопасности (SAST) в Python?	ПК-1

Номер задания	Правильный ответ	Содержание вопроса	Компетенция
		A. pytest B. Bandit C. flake8 D. black	
6	D	Какой принцип SOLID требует инверсии зависимостей (зависимость от абстракций, а не от конкретных реализаций)? A. SRP B. OCP C. LSP D. DIP	ПК-1
7	A, C	Какие паттерны проектирования GoF наиболее полезны для построения CI/CD пайплайнов? (выберите все верные) A. Strategy B. Composite C. Observer D. Flyweight	ПК-1
8	Hexagonal (Ports & Adapters)	Как называется архитектурный подход, в котором доменный слой изолирован от внешних систем через порты и адаптеры?	ПК-1
9	True	Верно ли, что чистые функции в функциональном программировании не имеют побочных эффектов и детерминированы? (True/False)	ПК-1

Номер задания	Правильный ответ	Содержание вопроса	Компетенция
10	В	Что такое asyncio в Python? А. Многопоточная библиотека для CPU-bound задач В. Библиотека для асинхронного программирования на основе корутин С. Инструмент для синхронизации потоков Д. Фреймворк для тестирования	ПК-1
11	Асинхронность (asynchronous)	Как называется парадигма программирования, позволяющая выполнять несколько операций ввода-вывода без блокировки основного потока выполнения?	ПК-1
12	С	Что такое SemVer (семантическое версионирование)? А. Схема нумерации коммитов в Git В. Алгоритм шифрования артефактов С. Правило формирования версий MAJOR.MINOR.PATCH с семантическим значением Д. Протокол обмена данными между микросервисами	ПК-1
13	В	Какой инструмент используется для управления секретами (паролями, токенами, ключами) в CI/CD? А. SonarQube В. HashiCorp Vault С. Nexus Д. Grafana	ПК-1
14	Configuration as	Как называется практика хранения конфигураций приложения и инфраструктуры	ПК-1

Номер задания	Правильный ответ	Содержание вопроса	Компетенция
	Code	в системе контроля версий?	
15	A	Что такое multi-stage build в Docker? A. Техника сборки образа в несколько этапов для минимизации размера B. Запуск нескольких контейнеров одновременно C. Сборка образов для разных архитектур процессоров D. Кэширование слоёв Docker	ПК-1
16	B	Какой ресурс является «единым источником истины» в GitOps подходе? A. Kubernetes кластер B. Git репозиторий с манифестами C. CI/CD пайплайн D. Container registry	ПК-1
17	Argo CD	Какой инструмент используется для синхронизации состояния Kubernetes кластера с Git-репозиторием манифестов?	ПК-1
18	A	Какая стратегия развертывания предполагает наличие двух идентичных сред (blue и green) с переключением трафика между ними? A. Blue-Green B. Canary C. Rolling Update D. A/B testing	ПК-1

Номер задания	Правильный ответ	Содержание вопроса	Компетенция
19	Canary	Какая стратегия развертывания предполагает постепенное наращивание процента трафика на новую версию?	ПК-1
20	B	Что такое SLI (Service Level Indicator)? A. Целевой уровень надёжности сервиса B. Количественно измеряемый показатель качества сервиса C. Документ по анализу инцидента D. Бюджет ошибок сервиса	ПК-1
21	C	Что такое Error Budget? A. Бюджет на закупку серверов B. Количество ошибок в коде на 1000 строк C. Допустимое количество сбоев в рамках SLO D. Время восстановления после сбоя	ПК-1
22	True	Верно ли, что ReadinessProbe в Kubernetes проверяет, готов ли контейнер принимать трафик, а LivenessProbe определяет, нужно ли перезапустить контейнер? (True/False)	ПК-1
23	C	Какой из перечисленных инструментов относится к системам сбора метрик? A. Grafana B. SonarQube C. Prometheus D. Nexus	ПК-1

Номер задания	Правильный ответ	Содержание вопроса	Компетенция
24	Grafana	Какой инструмент используется для визуализации метрик и построения дашбордов в связке с Prometheus?	ПК-1
25	D	Что такое Observability в контексте программных систем? A. Только сбор метрик B. Только сбор логов C. Только сбор трейсов D. Совокупность метрик, логов и трейсов для понимания состояния системы	ПК-1
26	B	Какой инструмент используется для нагрузочного тестирования? A. pytest B. k6 C. bandit D. flake8	ПК-1
27	D	Что такое технический долг (technical debt)? A. Задолженность перед заказчиком B. Стоимость лицензий на ПО C. Время на развёртывание приложения D. Цена ускоренной разработки, проявляющаяся в сложности поддержки кода	ПК-1
28	A, B	Какие из перечисленных практик относятся к DevSecOps? (выберите все верные) A. SAST (статический анализ безопасности) B. Сканирование уязвимостей зависимостей	ПК-1

Номер задания	Правильный ответ	Содержание вопроса	Компетенция
		С. Unit-тестирование D. Нагрузочное тестирование	
29	False	Верно ли, что SAST анализ требует запуска приложения для обнаружения уязвимостей? (True/False)	ПК-1
30	В	Что такое DAST? А. Статический анализ кода В. Динамический анализ безопасности запущенного приложения С. Анализ зависимостей D. Анализ покрытия кода	ПК-1
31	False	Верно ли, что в Python можно использовать истинную многопоточность для ускорения CPU-bound задач благодаря отсутствию GIL? (True/False)	ПК-1
32	GIL (Global Interpreter Lock)	Как называется механизм в CPython, ограничивающий выполнение только одного потока Python-кода одновременно?	ПК-1
33	В	Что такое Bounded Context в DDD? А. Ограниченная память сервера В. Явно заданная граница модели предметной области С. Ограничение по времени выполнения тестов D. Бюджет ошибок	ПК-1

Номер задания	Правильный ответ	Содержание вопроса	Компетенция
34	A	Какой паттерн архитектуры предполагает разделение операций на команды (изменение состояния) и запросы (чтение)? A. CQRS B. Event Sourcing C. Saga D. Circuit Breaker	ПК-1
35	C	Какой архитектурный паттерн предотвращает каскадные отказы, отключая проблемные компоненты? A. Retry B. Timeout C. Circuit Breaker D. Bulkhead	ПК-1
36	D	Что такое SBOM (Software Bill of Materials)? A. Бюджет на программное обеспечение B. Смета на разработку C. План тестирования D. Спецификация всех компонентов и зависимостей ПО	ПК-1
37	D	Какой инструмент используется для управления артефактами (бинарными репозиториями)? A. Prometheus B. Grafana	ПК-1

Номер задания	Правильный ответ	Содержание вопроса	Компетенция
		C. Argo CD D. Nexus	
38	True	Верно ли, что Helm является пакетным менеджером для Kubernetes? (True/False)	ПК-1
39	B	Что такое Git hook? A. Интеграция с GitHub B. Сценарий, автоматически запускаемый Git при определённых событиях C. Ветка в репозитории D. Инструмент для слияния кода	ПК-1
40	A, D	Какие инструменты можно использовать для статического анализа кода Python? (выберите все верные) A. flake8 B. pytest C. k6 D. mypy	ПК-1
41	D	Что такое artifact в контексте CI/CD? A. Ошибка в коде B. Ветка в репозитории C. Результат выполнения тестов D. Результат сборки (библиотека, Docker-образ, установочный файл)	ПК-1

Номер задания	Правильный ответ	Содержание вопроса	Компетенция
42	В	Какая команда Git используется для создания тега версии? A. git commit B. git tag C. git branch D. git merge	ПК-1
43	False	Верно ли, что при использовании GitOps изменения в Kubernetes можно вносить напрямую командой kubectl apply, и ArgoCD автоматически обновит репозиторий? (True/False)	ПК-1
44	В	Что такое Self-Healing в контексте Kubernetes? A. Автоматическое обновление приложения B. Автоматическое восстановление отказавших подов C. Автоматическое масштабирование D. Автоматическое резервное копирование	ПК-1
45	А, В	Какие метрики относятся к DORA (DevOps Research and Assessment)? (выберите все верные) A. Deployment Frequency B. Lead Time for Changes C. Code Coverage D. Cyclomatic Complexity	ПК-1
46	А	Что такое pre-commit hook?	ПК-1

Номер задания	Правильный ответ	Содержание вопроса	Компетенция
		А. Проверка кода перед коммитом В. Проверка кода после коммита С. Проверка перед push D. Проверка после pull	
47	testcontainers	Как называется библиотека для запуска реальных зависимостей (БД, очереди) в контейнерах для интеграционного тестирования?	ПК-1
48	В	Какой принцип SOLID утверждает, что объекты в программе могут быть заменены их наследниками без изменения корректности программы? А. SRP В. LSP С. ISP D. DIP	ПК-1
49	D	Что такое cyclomatic complexity? А. Количество строк кода В. Количество классов в проекте С. Количество комментариев D. Мера сложности кода, основанная на количестве независимых путей выполнения	ПК-1
50	С	Какой инструмент используется для создания временных Kubernetes кластеров для тестирования?	ПК-1

Номер задания	Правильный ответ	Содержание вопроса	Компетенция
		A. Docker Compose B. Argo CD C. Minikube D. Helm	

Описание шкалы оценивания

В рамках рейтинговой системы успеваемость студентов по каждой дисциплине оценивается в ходе текущего контроля и промежуточной аттестации. Рейтинговая система оценки знаний студентов основана на использовании совокупности контрольных мероприятий по проверке пройденного материала, оптимально расположенных на всем временном интервале изучения дисциплины. Принципы рейтинговой системы оценки знаний студентов основываются на требованиях, описанных в Положении об организации образовательного процесса на основе рейтинговой системы оценки знаний студентов в ФГАОУ ВО «СКФУ».

Критерии оценивания компетенций

Оценка «**отлично**» выставляется студенту, если он:

1. Демонстрирует полное и глубокое понимание теоретических основ CI/CD, включая архитектуру пайплайнов, принципы непрерывной интеграции, поставки и развертывания, методы обеспечения качества кода, безопасности и наблюдаемости.
2. Самостоятельно проектирует архитектуру CI/CD пайплайна для различных типов приложений (монолитные, микросервисные, встраиваемые системы), обосновывая выбор этапов, инструментов и последовательности выполнения.
3. Реализует полностью рабочий пайплайн непрерывной интеграции, включающий все обязательные этапы: линтинг, unit-тестирование, интеграционное тестирование, статический анализ безопасности (SAST), измерение покрытия кода с пороговым значением не менее 80%.
4. Настраивает Quality Gate в SonarQube с корректными пороговыми значениями, демонстрирует автоматическую блокировку сборки при нарушении требований к качеству кода.
5. Внедряет управление артефактами: автоматическую сборку, семантическое версионирование (SemVer) и публикацию артефактов (Python wheel и Docker-образ) в бинарный репозиторий.
6. Реализует безопасное управление конфигурациями и секретами, используя подход Configuration as Code, переменные CI/CD и HashiCorp Vault с динамической генерацией секретов.
7. Применяет методы оптимизации пайплайна: параллелизацию этапов (matrix, parallel), кэширование зависимостей и асинхронные/многопоточные подходы для ускорения выполнения.
8. Реализует стратегии развертывания (Blue-Green и Canary) в Kubernetes с автоматическим откатом на основе health-проверок (readinessProbe, livenessProbe).
9. Внедряет GitOps подход с использованием ArgoCD, настраивает автоматическую синхронизацию кластера с репозиторием манифестов и self-healing.
10. Настраивает наблюдаемость пайплайна и приложения: развёртывает Prometheus и Grafana, создаёт дашборды, определяет SLI/SLO и добавляет в CD-пайплайн этап проверки SLI с автоматическим откатом при нарушении SLO.
11. Выполняет сквозной проект в полном объёме (не менее 14 этапов из 16), демонстрирует работоспособность пайплайна «от коммита до деплоя» на реальном или учебном приложении.
12. Оформляет отчёт по всем требованиям: структурирован, содержит все необходимые скриншоты, листинги кода и конфигураций, анализ результатов и выводы.
13. Уверенно отвечает на дополнительные вопросы преподавателя, демонстрирует понимание принципов работы используемых инструментов и способность адаптировать их для решения нестандартных задач.

Оценка «**хорошо**» выставляется студенту, если он:

1. Демонстрирует хорошее понимание теоретических основ CI/CD, но допускает отдельные неточности (например, в определении терминов или последовательности этапов).
2. Проектирует архитектуру CI/CD пайплайна с учётом требований, но допускает незначительные ошибки в выборе инструментов или последовательности этапов.
3. Реализует пайплайн непрерывной интеграции, включающий не менее 10–12 этапов из 16, с небольшими замечаниями по конфигурации (не оптимальные пороги, отсутствие блокировки при падении отдельных тестов).
4. Настраивает Quality Gate в SonarQube, но пороговые значения установлены не оптимально или блокировка сборки работает не для всех нарушений.
5. Внедряет управление артефактами, но версионирование не полностью автоматизировано или публикация в репозиторий требует ручных действий.
6. Реализует управление конфигурациями и секретами, но использует только переменные CI/CD без интеграции с Vault, или конфигурации разделены не для всех сред.
7. Применяет методы оптимизации пайплайна, но использует 1-2 метода из трёх (например, кэширование без параллелизации).
8. Реализует одну стратегию развертывания (Rolling Update или Blue-Green) без автоматического отката или с частично настроенными health-проверками.
9. Внедряет GitOps подход, но синхронизация выполняется вручную или auto-sync настроен с задержками.
10. Настраивает наблюдаемость, но дашборд содержит менее 3 панелей, проверка SLI/SLO не автоматизирована.
11. Выполняет сквозной проект в объёме 11–13 этапов, пайплайн работает, но есть незначительные ошибки или недочёты.
12. Оформляет отчёт с небольшими замечаниями (неполные скриншоты, отсутствие отдельных разделов).
13. Отвечает на большинство вопросов преподавателя, но на сложные вопросы может отвечать с наводящими подсказками.

Оценка «**удовлетворительно**» выставляется студенту, если он:

1. Демонстрирует базовое понимание основных понятий CI/CD, но допускает существенные ошибки в определении терминов и принципов.
2. Проектирует пайплайн только для одного типа приложений, допускает ошибки в последовательности этапов (например, запуск деплоя до тестов).
3. Реализует пайплайн непрерывной интеграции, включающий 8–10 этапов из 16, но пайплайн работает с ошибками, требует доработки или помощи преподавателя.
4. Пытается настроить Quality Gate, но пороговые значения не соответствуют требованиям, блокировка сборки не работает.
5. Частично внедряет управление артефактами (например, только сборка без публикации или публикация без версионирования).
6. Использует конфигурации, но секреты могут быть раскрыты в коде или логах, интеграция с Vault отсутствует.
7. Не применяет методы оптимизации пайплайна или применяет их с грубыми ошибками.
8. Реализует только базовое развертывание (например, `docker-compose up`) без стратегий и автоматического отката.
9. Знаком с GitOps на уровне названия, но ArgoCD не настроен или не работает.
10. Настраивает только сбор метрик, но дашборд отсутствует, проверки SLI/SLO нет.
11. Выполняет сквозной проект в объёме 8–10 этапов, пайплайн работает с ошибками, требует исправлений.
12. Оформляет отчёт минимально (только листинги, отсутствуют скриншоты и

анализ результатов).

13. Отвечает только на простые вопросы, на дополнительные отвечает неправильно или не отвечает совсем.

Оценка «неудовлетворительно» выставляется студенту, если он:

1. Не понимает основных понятий CI/CD, не может объяснить разницу между CI, CD и CD (deployment).
2. Не может спроектировать архитектуру пайплайна или предлагает заведомо нерабочую схему.
3. Пайплайн непрерывной интеграции отсутствует или неработоспособен, этапы не выполняются или выполняются с критическими ошибками.
4. Не настраивает или не может настроить Quality Gate, статический анализ отсутствует.
5. Не внедряет управление артефактами (артефакты не собираются или не сохраняются).
6. Не использует управление конфигурациями, секреты хранит в коде в открытом виде.
7. Не применяет методы оптимизации, пайплайн выполняется последовательно и медленно.
8. Не реализует развертывание или развертывание выполняется вручную вне пайплайна.
9. Не знаком с GitOps, ArgoCD не установлен и не настроен.
10. Не настраивает наблюдаемость, мониторинг отсутствует.
11. Сквозной проект не выполнен или выполнен менее чем на 8 этапов, решение неработоспособно.
12. Отчёт отсутствует или оформлен формально, не соответствует требованиям.
13. Не может ответить ни на один вопрос преподавателя или демонстрирует полное непонимание предмета.