

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
Уникальный программный ключ:
990bea3aeec848c23bd8699e48288f8d1960c600

Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

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

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ
Проектирование и оптимизация конвейеров CI/CD

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	1

Введение

Назначение: ФОС предназначен для объективной оценки уровня сформированности компетенций.

ФОС является приложением к программе дисциплины «Проектирование и оптимизация конвейеров CI/CD».

Разработчик Е.Н. Новикова, к.ф.-м.н., доцент, заведующий межинститутский 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				
ИД-1 _{ПК-1} Проектирует архитектуру CI/CD-конвейеров, определяя этапы сборки, тестирования и развертывания для различных типов приложений и платформ.	Не может назвать этапы CI/CD-пайплайна. Не понимает различий между типами приложений	С помощью преподавателя определяет базовые этапы (build, test) для простого приложения. Знает определение CI/CD.	Самостоятельно проектирует пайплайн из 3-4 этапов для типового приложения. Обосновывает выбор этапов.	Проектирует сложные пайплайны (6+ этапов) для различных типов систем (микросервисы, встроенные). Учитывает специфику платформы
ИД-2 _{ПК-1} Реализует автоматизированные процессы непрерывной интеграции, включая компиляцию кода, запуск модульных и интеграционных тестов, статический анализ	Не может написать конфигурационный файл CI/CD. Тесты не запускаются или отсутствуют	С помощью шаблонов настраивает базовый CI-пайплайн (сборка + запуск unit-тестов).	Самостоятельно настраивает многостадийный CI-пайплайн с модульными и интеграционными тестами, генерацией отчетов	Реализует CI-пайплайн с параллелизацией, кэшированием, матричными стратегиями, интеграцией статического анализа (линтеры).
ИД-3 _{ПК-1} Интегрирует инструменты безопасности (SAST, DAST, SCA) в конвейеры CI/CD для автоматического сканирования уязвимостей на ранних этапах разработки	Не знает инструментов безопасности. Не может объяснить, зачем нужен SAST/SCA	Устанавливает и запускает один SAST-инструмент (Bandit/Semgrep) по инструкции	Интегрирует SAST и SCA-инструменты в пайплайн. Настраивает генерацию отчетов	Настраивает Quality Gate (пороги) с автоматическим срывом пайплайна при HIGH/CRITICAL уязвимостях. Интерпретирует отчеты
ИД-4 _{ПК-1} Настраивает системы управления артефактами для централизованного хранения,	Не знает, что такое реестр артефактов. Не может опубликовать артефакт	По инструкции публикует артефакт (например, wheel-пакет) в package registry	Настраивает публикацию Docker-образа и версионирование (commit hash, git tag).	Настраивает подпись артефактов (Cosign), реализует политику хранения артефактов (retention).

версионирования и подписывания бинарных файлов, контейнеров и прошивок				
ИД-5 _{ПК-1} Организует процессы непрерывной поставки и развертывания с использованием стратегий zero-downtime (blue-green, canary) для облачных сервисов	Не знает стратегий развертывания. Не может развернуть приложение в K8s	С помощью инструкции разворачивает приложение в Kubernetes (kubectl apply).	Реализует rolling update или blue-green развертывание через kubectl	Реализует canary-развертывание (10% трафика) с анализом метрик и автоматическим откатом. Использует Argo CD
ИД-6 _{ПК-1} Разрабатывает механизмы доставки обновлений (ОТА) для встроенных систем и автономных устройств с учетом ограничений каналов связи и требований к целостности	Не знает особенностей CI/CD для встроенных систем. Не может собрать прошивку	С помощью инструкции настраивает кросс-компилятор и собирает прошивку (.bin/.hex).	Настраивает пайплайн сборки прошивки с сохранением артефактов. Запускает SITL-тесты в симуляторе	Реализует полный ОТА-пайплайн: сборка → подпись (Cosign) → публикация → симуляция получения обновления устройством.
ИД-7 _{ПК-1} Оценивает эффективность работы CI/CD-конвейеров и оптимизирует их для сокращения времени цикла разработки и повышения надежности развертываний	Не может назвать метрики эффективности. Не анализирует время выполнения пайплайна	По инструкции замеряет время выполнения пайплайна. Знает метрики DORA	Анализирует узкие места пайплайна. Внедряет кэширование и параллелизацию, добиваясь ускорения >30%.	Оптимизирует пайплайн комплексно (кэширование, параллелизация, шардирование тестов, выбор раннеров). Документирует улучшения
ПК-3				
ИД-1 _{ПК-3} Анализирует архитектуру распределенных систем и определяет оптимальные стратегии развертывания (blue-green, rolling update, canary) с учетом требований к доступности и допустимых рисков	Не может объяснить различия между стратегиями развертывания. Не учитывает требования к доступности	Понимает разницу между rolling update, blue-green и canary. Может выбрать стратегию по подсказке преподавателя	Анализирует архитектуру сервиса и самостоятельно выбирает стратегию развертывания, обосновывая выбор SLA/SLO	Для сложных систем предлагает гибридные стратегии (canary + feature flags). Учитывает риски отката и миграции данных
ИД-2 _{ПК-3} Реализует механизмы zero-downtime развертывания с использованием балансировщиков нагрузки и оркестраторов	Не может настроить zero-downtime развертывание. При обновлении сервис становится недоступен	По инструкции настраивает rolling update в Kubernetes	Самостоятельно реализует blue-green развертывание с переключением Service	Реализует zero-downtime развертывание с canary-анализом (метрики ошибок) и автоматическим откатом

контейнеров.				
ИД-3 ПК-3 Внедряет канареечные (canary) релизы для поэтапного тестирования новых версий на ограниченной аудитории пользователей	Не знает концепции canary-релизов	По инструкции настраивает canary с фиксированным весом (например, 10%) через Nginx Ingress.	Настраивает canary-развертывание с поэтапным увеличением трафика	Интегрирует canary-анализ на основе метрик (error rate, latency P99) с автоматическим откатом при ухудшении метрик
ИД-4 ПК-3 Применяет системы функциональных флагов (feature flags) для контролируемого включения/выключения нового функционала без повторного развертывания	Не знает, что такое feature flags. Не может реализовать простейший флаг в коде	По инструкции реализует простой feature flag (переменная окружения).	Интегрирует систему feature flags (Unleash/LaunchDarkly) в приложение. Настраивает включение флага через консоль	Реализует сложные сценарии: A/B тестирование, процентный rollout флага, таргетинг на пользователей
ИД-5 ПК-3 Проектирует автоматизированные процедуры отката (rollback) на случай выявления проблем в процессе развертывания, минимизируя время восстановления сервиса (MTTR).	Не может выполнить откат вручную. Не понимает, как минимизировать MTTR.	По инструкции выполняет ручной откат (kubectl rollout undo).	Проектирует автоматический откат при провале smoke-тестов (в CI/CD пайплайне).	Реализует автоматический откат на основе анализа метрик (error rate, latency) без участия человека.
ИД-6 ПК-3 Интегрирует стратегии развертывания и отката в финальные стадии CI/CD-пайплайнов с автоматическим сбором метрик и алертингом.	Не интегрирует развертывание в CI/CD. Выполняет деплой вручную.	Добавляет этап развертывания в CI/CD (manual или automatic).	Настраивает сбор метрик (Prometheus) и алертинг (Alertmanager) для этапов развертывания	Реализует полноценный progressive delivery пайплайн с canary-анализом, автоматическим откатом и дашбордом в Grafana
ИД-7 ПК-3 Обеспечивает обратную совместимость изменений и безопасность миграции данных при развертывании новых версий и последующем возможном откате	Не учитывает миграции БД при развертывании. При откате теряет данные	Понимает проблему, но не может реализовать обратную совместимую миграцию	Проектирует миграции с поддержкой обратной совместимости (expand/contract pattern).	Реализует тестирование обратной совместимости в CI/CD, обеспечивает безопасный откат без потери данных

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕРКИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Номер задания	Правильный ответ	Содержание вопроса	Компетенция
Семестр 1			
1.	Continuous Integration	Что означает аббревиатура CI в контексте DevOps?	ПК-1
2.	Continuous Delivery, Continuous Deployment	Что означает аббревиатура CD в контексте DevOps? (укажите два варианта)	ПК-1
3.	.github/workflows/.yml	Какой файл является основным конфигурационным файлом пайплайна в GitHub Actions?	ПК-1
4.	.gitlab-ci.yml	Какой файл является основным конфигурационным файлом пайплайна в GitLab CI?	ПК-1
5.	kubectl rollout status	Какая команда используется для проверки статуса развертывания в Kubernetes?	ПК-3
6.	Rolling Update	Какая стратегия развертывания предполагает постепенную замену старых подов новыми?	ПК-3
7.	Blue-Green Deployment	Какая стратегия развертывания предполагает два идентичных окружения (активное и неактивное)?	ПК-3
8.	Canary Deployment	Какая стратегия развертывания направляет лишь небольшой процент трафика на новую версию?	ПК-3
9.	Argo CD	Какой инструмент используется для GitOps в Kubernetes?	ПК-3
10.	Cosign	Какой инструмент используется для подписи контейнеров и артефактов?	ПК-1
11.	kubectl apply -f	Какая команда используется для применения манифестов в Kubernetes?	ПК-3
12.	Bandit (или Semgrep)	Какой инструмент статического анализа безопасности подходит для Python?	ПК-1
13.	OWASP Dependency-Check (или Trivy)	Какой инструмент используется для анализа зависимостей (SCA)?	ПК-1
14.	.bin	Какой формат артефакта используется для прошивки микроконтроллеров в виде сырых бинарных данных?	ПК-1
15.	Lead Time for Changes	Какая метрика DORA измеряет время от коммита до деплоя в production?	ПК-1
16.	Change Failure Rate	Какая метрика DORA измеряет процент неудачных развертываний?	ПК-1
17.	kubectl rollout undo	Какая команда используется для отката (rollback) deployment в Kubernetes?	ПК-3
18.	parallel	Какой инструмент используется для параллельного выполнения заданий в GitLab CI?	ПК-1
19.	strategy: matrix	Какой инструмент используется для матричных стратегий в GitHub Actions?	ПК-1
20.	Blue-Green	Какая стратегия развертывания минимизирует MTTR за счет мгновенного переключения трафика?	ПК-3
21.	Feature flags	Какой механизм позволяет включать/выключать функционал без повторного развертывания?	ПК-3

22.	Unleash (или Flagsmith)	Какой инструмент можно использовать для feature flags в self-hosted среде?	ПК-3
23.	Software-in-the-Loop	Что означает аббревиатура SITL?	ПК-1
24.	Gazebo	Какой симулятор часто используется с ROS 2 для тестирования роботов?	ПК-1
25.	nginx.ingress.kubernetes.io/canary-weight	Какая аннотация Nginx Ingress используется для указания процента canary-трафика?	ПК-3
26.	kubectl create namespace	Какая команда используется для создания namespace в Kubernetes?	ПК-3
27.	maxSurge	Какой параметр Kubernetes Deployment отвечает за максимальное количество дополнительных подов при rolling update?	ПК-3
28.	maxUnavailable	Какой параметр Kubernetes Deployment отвечает за максимальное количество недоступных подов при rolling update?	ПК-3
29.	Prometheus	Какой инструмент позволяет собирать метрики и отслеживать состояние приложений в K8s?	ПК-3
30.	Grafana	Какой инструмент используется для визуализации метрик Prometheus?	ПК-3
31.	.hex	Какой формат артефакта используется для прошивки в текстовом представлении (Intel HEX)?	ПК-1
32.	arm-none-eabi-gcc	Какой инструмент используется для кросс-компиляции под ARM?	ПК-1
33.	kubectl rollout history	Какая команда используется для просмотра истории развертываний в Kubernetes?	ПК-3
34.	pytest --split	Какой инструмент позволяет организовать параллельный запуск тестов в pytest?	ПК-1
35.	CI_COMMIT_SHORT_SHA	Какая переменная окружения в GitLab CI содержит короткий хэш коммита?	ПК-1
36.	JUnit XML	Какой формат отчета о тестировании поддерживается большинством CI-платформ?	ПК-1
37.	pytest-cov (или coverage.py)	Какой инструмент используется для измерения покрытия кода в Python?	ПК-1
38.	Jest --coverage	Какой инструмент используется для измерения покрытия кода в JavaScript?	ПК-1
39.	git tag <tagname>	Какая команда Git используется для создания тега?	ПК-1
40.	actions/upload-artifact	Какая команда используется для загрузки артефакта в GitHub Actions?	ПК-1
41.	HTTPS / TLS	Какой протокол используется для безопасной передачи артефактов?	ПК-1

42.	Expand/Contract pattern	Какой паттерн миграции БД обеспечивает обратную совместимость при blue-green развертывании?	ПК-3
43.	BuildPulse (или аналоги)	Какой инструмент позволяет анализировать flaky-тесты в CI/CD?	ПК-1
44.	GitLab SaaS Runner	Какой тип раннера в GitLab CI позволяет запускать задания в облаке?	ПК-1
45.	GITHUB_REF_NAME	Какая переменная в GitHub Actions содержит имя текущей ветки?	ПК-1
46.	Trivy (или Clair)	Какой инструмент используется для сканирования уязвимостей Docker-образов?	ПК-1
47.	when: manual	Какой параметр в GitLab CI позволяет задать ручное подтверждение выполнения задания?	ПК-3
48.	Canary by header	Какая стратегия развертывания позволяет тестировать новую версию на внутренних пользователях по HTTP-заголовку?	ПК-3
49.	GitOps (Argo CD)	Какой инструмент позволяет управлять конфигурацией через Git?	ПК-3
50.	Deployment Frequency	Какая метрика DORA измеряет частоту развертываний в production?	ПК-1
51.	B	Что означает аббревиатура CI в контексте DevOps A) Continuous Integration B) Continuous Integration C) Core Integration D) Code Integration	ПК-1
52.	B	A) Core Delivery B) Continuous Delivery / Continuous Deployment C) Code Deployment D) Centralized Delivery	ПК-1
53.	B	Какой файл является основным конфигурационным файлом пайплайна в GitLab CI A) .github/workflows/ci.yml B) .gitlab-ci.yml C) Jenkinsfile D) pipeline.yaml	ПК-1
54.	B	Какой инструмент используется для GitOps в Kubernetes A) Jenkins B) Argo CD C) GitLab Runner D) Prometheus	ПК-3
55.	A	Какая стратегия развертывания предполагает постепенную замену старых подов новыми A) Rolling Update B) Blue-Green Deployment C) Canary Deployment D) Recreate	ПК-3

56.	B	Какая стратегия развертывания предполагает два идентичных окружения (активное и неактивное) A) Rolling Update B) Blue-Green Deployment C) Canary Deployment D) Shadow Deployment	ПК-3
57.	C	Какая стратегия развертывания направляет лишь небольшой процент трафика на новую версию A) Rolling Update B) Blue-Green Deployment C) Canary Deployment D) A/B Deployment	ПК-3
58.	B	Какая команда используется для применения манифестов в Kubernetes A) kubectl create -f B) kubectl apply -f C) kubectl run -f D) kubectl exec -f	ПК-3
59.	C	Какой инструмент используется для подписи контейнеров и артефактов A) GPG B) OpenSSL C) Cosign D) HashiCorp Vault	ПК-1
60.	A	Какой инструмент статического анализа безопасности подходит для Python A) Bandit B) ESLint C) Clang-Tidy D) SpotBugs	ПК-1
61.	B	Какой инструмент используется для анализа зависимостей (SCA) A) Semgrep B) OWASP Dependency-Check C) Flake8 D) Pytest	ПК-1
62.	A	Какой формат артефакта используется для прошивки микроконтроллеров в виде сырых бинарных данных A) .bin B) .hex C) .elf D) .exe	ПК-1

63.	A	Какая метрика DORA измеряет время от коммита до деплоя в production A) Lead Time for Changes B) Deployment Frequency C) Change Failure Rate D) Time to Restore Service	ПК-1
64.	C	Какая метрика DORA измеряет процент неудачных развертываний A) Lead Time for Changes B) Deployment Frequency C) Change Failure Rate D) Time to Restore Service	ПК-1
65.	C	Какая команда используется для отката (rollback) deployment в Kubernetes A) kubectl delete deployment B) kubectl revert deployment C) kubectl rollout undo D) kubectl rollback	ПК-3
66.	B	Какой параметр в GitLab CI позволяет задать ручное подтверждение выполнения задания A) allow_failure: true B) when: manual C) only: manual D) stage: manual	ПК-3
67.	A	Какой механизм позволяет включать/выключать функционал без повторного развертывания A) Feature flags B) Environment variables C) Config maps D) Secrets	ПК-3
68.	B	Какой инструмент можно использовать для feature flags в self-hosted среде A) LaunchDarkly B) Unleash C) Flagsmith D) Split.io	ПК-3
69.	A	Что означает аббревиатура SITL A) Software-in-the-Loop B) System-in-the-Loop C) Simulation-in-the-Loop D) Safety-in-the-Loop	ПК-1
70.	A	Какой симулятор часто используется с ROS 2 для тестирования роботов A) Gazebo B) Unity C) Unreal Engine	ПК-1

		D) Blender	
71.	B	Какая аннотация Nginx Ingress используется для указания процента canary-трафика A) nginx.ingress.kubernetes.io/canary B) nginx.ingress.kubernetes.io/canary-weight C) nginx.ingress.kubernetes.io/canary-percent D) nginx.ingress.kubernetes.io/traffic-split	ПК-3
72.	B	Какой параметр Kubernetes Deployment отвечает за максимальное количество дополнительных подов при rolling update A) maxUnavailable B) maxSurge C) minReadySeconds D) revisionHistoryLimit	ПК-3
73.	A	Какой параметр Kubernetes Deployment отвечает за максимальное количество недоступных подов при rolling update A) maxUnavailable B) maxSurge C) minReadySeconds D) revisionHistoryLimit	ПК-3
74.	B	Какой инструмент позволяет собирать метрики и отслеживать состояние приложений в K8s A) Grafana B) Prometheus C) Alertmanager D) Loki	ПК-3
75.	A	Какой инструмент используется для визуализации метрик Prometheus A) Grafana B) Kibana C) Tableau D) Power BI	ПК-3
76.	B	Какой инструмент используется для кросс-компиляции под ARM A) gcc B) arm-none-eabi-gcc C) clang D) rustc	ПК-1
77.	C	Какая команда используется для просмотра истории развертываний в Kubernetes A) kubectl get deployment B) kubectl describe deployment C) kubectl rollout history D) kubectl logs deployment	ПК-3

78.	A	Какой формат отчета о тестировании поддерживается большинством CI-платформ A) JUnit XML B) JSON C) YAML D) CSV	ПК-1
79.	B	Какой инструмент используется для измерения покрытия кода в Python A) Jest B) pytest-cov C) Mocha D) JUnit	ПК-1
80.	C	Какая команда Git используется для создания тега A) git commit -m B) git push C) git tag <tagname> D) git branch	ПК-1
81.	A	Какой паттерн миграции БД обеспечивает обратную совместимость при blue-green разворачивании A) Expand/Contract pattern B) Strangler pattern C) CQRS pattern D) Saga pattern	ПК-3
82.	A	Какой тип раннера в GitLab CI позволяет запускать задания в облаке A) GitLab SaaS Runner B) GitLab Self-hosted Runner C) GitLab Shared Runner D) GitLab Custom Runner	ПК-1
83.	A	Какая переменная в GitHub Actions содержит имя текущей ветки A) GITHUB_REF_NAME B) GITHUB_BRANCH C) GITHUB_REF_BRANCH D) GITHUB_BRANCH_NAME	ПК-1
84.	A	Какой инструмент используется для сканирования уязвимостей Docker-образов A) Trivy B) Semgrep C) Bandit D) Flake8	ПК-1
85.	B	Какая переменная окружения в GitLab CI содержит короткий хэш коммита A) CI_COMMIT_SHA B) CI_COMMIT_SHORT_SHA C) CI_COMMIT_HASH	ПК-1

		D) CI COMMIT ID	
86.	A	Что такое Quality Gate в контексте CI/CD A) Пороговое значение качества, при недостижении которого пайплайн останавливается B) Шлюз для доступа к production-среде C) Инструмент мониторинга качества кода D) Система контроля версий	ПК-1
87.	A	Что такое Pipeline as Code A) Хранение конфигурации CI/CD в репозитории вместе с кодом B) Написание пайплайна на языке программирования C) Автоматическая генерация пайплайна из кода D) Визуальное проектирование пайплайна	ПК-1
88.	B	Что такое артефакт в CI/CD A) Результат работы тестов B) Результат сборки (бинарный файл, пакет, образ) C) Конфигурационный файл D) Лог выполнения пайплайна	ПК-1
89.	A	Что такое GitOps A) Подход, при котором Git является единственным источником истины для инфраструктуры B) Система контроля версий для DevOps C) Инструмент для управления Git репозиториями D) Методология разработки через форки	ПК-3
90.	B	Что такое readinessProbe в Kubernetes A) Проверка, жив ли контейнер B) Проверка, готов ли контейнер принимать трафик C) Проверка доступности сервиса D) Проверка версии образа	ПК-3
91.	A	Что такое livenessProbe в Kubernetes A) Проверка, жив ли контейнер (требуется перезапуска) B) Проверка, готов ли контейнер принимать трафик C) Проверка доступности сервиса D) Проверка версии образа	ПК-3
92.	A	Что такое MTTR A) Mean Time To Recovery (среднее время восстановления) B) Mean Time To Response C) Maximum Time To Recovery D) Minimum Time To Response	ПК-3
93.	B	Что такое Error Budget A) Бюджет на исправление ошибок B) Допустимый процент времени недоступности сервиса	ПК-1

		C) Количество ошибок в коде D) Бюджет на тестирование	
94.	A	Что такое SLO A) Service Level Objective (целевой уровень обслуживания) B) Service Level Agreement (соглашение об уровне обслуживания) C) Service Level Indicator (индикатор уровня обслуживания) D) Service Level Objective	ПК-1
95.	C	Что такое SLI A) Service Level Objective B) Service Level Agreement C) Service Level Indicator D) Service Level Integration	ПК-1
96.	B	Какой протокол используется для безопасной передачи артефактов A) HTTP B) HTTPS / TLS C) FTP D) SSH	ПК-1
97.	A	Какой инструмент позволяет анализировать flaky-тесты в CI/CD A) BuildPulse B) SonarQube C) Grafana D) Prometheus	ПК-1
98.	B	Что такое canary by header в Nginx Ingress A) Маршрутизация по cookie B) Маршрутизация по HTTP-заголовку для определенных пользователей C) Маршрутизация по IP-адресу D) Маршрутизация по геолокации	ПК-3
99.	B	Что такое OTA (Over-the-Air) обновление A) Обновление через USB B) Беспроводное обновление прошивки C) Обновление через Ethernet D) Обновление через JTAG	ПК-1
100.	A	Какой инструмент используется для проверки покрытия кода в JavaScript A) Jest –coverage B) pytest-cov C) flake D) go test -cover	ПК-1

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» (5)

выставляется студенту, если он выполняет в полном объеме и безошибочно задания оценочных средств по дисциплине, относящиеся к данной компетенции (индикаторам достижения компетенции), при этом демонстрирует на высоком уровне способность:

- Проектировать архитектуру CI/CD-пайплайнов для различных типов приложений (микросервисные, встроенные, робототехнические);
- Реализовывать автоматизированные процессы непрерывной интеграции, поставки и развертывания с использованием современных инструментов (GitLab CI / GitHub Actions);
- Интегрировать инструменты безопасности (SAST, SCA) в конвейеры CI/CD с настройкой Quality Gate;
- Организовывать zero-downtime развертывание с использованием стратегий rolling update, blue-green, canary и feature flags;
- Разрабатывать OTA-обновления для встроенных систем и автономных устройств с обеспечением целостности и подписи;
- Оценивать эффективность CI/CD-пайплайнов и оптимизировать их по времени выполнения и стоимости.

Оценка «хорошо» (4)

выставляется студенту, если он выполняет, допуская незначительные ошибки, задания оценочных средств по дисциплине, относящиеся к данной компетенции (индикаторам достижения компетенции), при этом демонстрирует на среднем уровне способность:

- Проектировать CI/CD-пайплайны для типовых приложений;
- Реализовывать базовые процессы CI (сборка, тестирование) и CD (развертывание);
- Настраивать инструменты безопасности по инструкции;
- Реализовывать одну-две стратегии развертывания (rolling update или blue-green);
- Настраивать кросс-компиляцию и сборку прошивки;
- Анализировать время выполнения пайплайна и применять базовые методы оптимизации (кэширование).

Оценка «удовлетворительно» (3)

выставляется студенту, если он выполняет, допуская ошибки, исправление которых осуществляет с помощью наводящих вопросов преподавателя, задания оценочных средств по дисциплине, относящиеся к данной компетенции (индикаторам достижения компетенции), при этом демонстрирует на минимальном уровне способность:

- Называть этапы CI/CD-пайплайна;
- Запускать сборку и тесты по готовому шаблону;
- Выполнять базовые команды Kubernetes (kubectl apply, rollout status);
- Публиковать артефакт в реестр по инструкции;
- Собирать прошивку по готовому Makefile.

Оценка «неудовлетворительно» (2)

выставляется студенту, если он, выполняя задания оценочных средств по дисциплине,

не достигает минимального уровня сформированности компетенции (индикаторам достижения компетенции), при этом:

- Не демонстрирует способность проектировать CI/CD-пайплайны;
- Допускает грубые ошибки в понимании принципов CI/CD, которые не может исправить даже с помощью наводящих вопросов преподавателя;
- Не может выполнить базовые операции (сборка, тестирование, развертывание) даже по инструкции;
- Не ориентируется в основных инструментах и стратегиях развертывания.