Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Порохня Андрей Алексеевич
Должность: И.о. директора института перспективной инженерии
Дата подписания: 19.06.2026 14:51:47
Уникальный программный ключ:
990bea3aeecc48c23bd8699e48288f8d1960c600

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
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 производственной практике
эксплуатационной практике

Направление подготовки	09.04.04 Программная инженерия
Направленность (профиль)	Управление знаниями
Год начала обучения	2026
Форма обучения	очная
Реализуется в семестре	4

Введение

Назначение: ФОС предназначен для объективной оценки уровня сформированности компетенций.

ФОС является приложением к программе практики.

Разработчик Анна Юрьевна Орлова, к.э.н., доцент, доцент департамента цифровых, робототехнических систем и электроники института перспективной инженерии.

Проведена экспертиза ФОС.

Члены экспертной группы:

Председатель Новикова Е.Н., заведующий 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 показателей и критериев оценивания на различных этапах их формирования, описание шкал оценивания

Уровни сформированности компетенции (ий), индикатора (ов)	Дескрипторы			
	Минимальный уровень не достигнут (Неудовлетворительно) 2 балла	Минимальный уровень (удовлетворительно) 3 балла	Средний уровень (хорошо) 4 балла	Высокий уровень (отлично) 5 баллов
ПК-2				
ИД-1 $_{ПК-2}$. Описывает облачные ресурсы, сетевые компоненты и вычислительные мощности на декларативных языках конфигураций (HCL, YAML, JSON) с использованием инструментов класса IaC (Terraform, CloudFormation, Pulumi).	С трудом описывает облачные ресурсы, сетевые компоненты и вычислительные мощности на декларативных языках конфигураций с использованием инструментов класса IaC	Слабо описывает облачные ресурсы, сетевые компоненты и вычислительные мощности на декларативных языках конфигураций с использованием инструментов класса IaC	В основном описывает облачные ресурсы, сетевые компоненты и вычислительные мощности на декларативных языках конфигураций с использованием инструментов класса IaC	описывает облачные ресурсы, сетевые компоненты и вычислительные мощности на декларативных языках конфигураций с использованием инструментов класса IaC
ИД-2 $_{ПК-2}$. Разрабатывает идемпотентные сценарии управления конфигурациями (Ansible, Puppet, Chef) для настройки операционных систем, прикладного ПО и бортового программного обеспечения автономных устройств.	С трудом разрабатывает идемпотентные сценарии управления конфигурациями для настройки операционных систем	Слабо разрабатывает идемпотентные сценарии управления конфигурациями для настройки операционных систем	В основном разрабатывает идемпотентные сценарии управления конфигурациями для настройки операционных систем	разрабатывает идемпотентные сценарии управления конфигурациями для настройки операционных систем
ИД-3 $_{ПК-2}$. Применяет системы контроля версий (Git) для хранения, версионирования и совместной работы над кодом инфраструктуры и конфигураций.	С трудом применяет системы контроля версий для хранения, версионирования и совместной работы над кодом инфраструктуры и конфигураций	Слабо применяет системы контроля версий для хранения, версионирования и совместной работы над кодом инфраструктуры и конфигураций	В основном применяет системы контроля версий для хранения, версионирования и совместной работы над кодом инфраструктуры и конфигураций	применяет системы контроля версий для хранения, версионирования и совместной работы над кодом инфраструктуры и конфигураций
ИД-4 $_{ПК-2}$. Интегрирует процессы	С трудом интегрирует процессы развертывания	Слабо интегрирует процессы	В основном интегрирует процессы	интегрирует процессы развертывания инфраструктуры и

развертывания инфраструктуры и применения конфигураций в пайплайны непрерывной интеграции и доставки (CI/CD).	инфраструктуры и применения конфигураций в пайплайны непрерывной интеграции и доставки	развертывания инфраструктуры и применения конфигураций в пайплайны непрерывной интеграции и доставки	развертывания инфраструктуры и применения конфигураций в пайплайны непрерывной интеграции и доставки	применения конфигураций в пайплайны непрерывной интеграции и доставки
ИД-5 $_{ПК-2}$. Обеспечивает безопасность управления инфраструктурой через код: внедряет сканирование кода на уязвимости, управляет секретами и реализует принцип минимальных привилегий.	С трудом обеспечивает безопасность управления инфраструктурой через код: внедряет сканирование кода на уязвимости, управляет секретами и реализует принцип минимальных привилегий	Слабо обеспечивает безопасность управления инфраструктурой через код: внедряет сканирование кода на уязвимости, управляет секретами и реализует принцип минимальных привилегий	В основном обеспечивает безопасность управления инфраструктурой через код: внедряет сканирование кода на уязвимости, управляет секретами и реализует принцип минимальных привилегий	обеспечивает безопасность управления инфраструктурой через код: внедряет сканирование кода на уязвимости, управляет секретами и реализует принцип минимальных привилегий
ИД-6 $_{ПК-2}$. Реализует GitOps-подход для автоматической синхронизации состояния инфраструктуры с конфигурацией в Git-репозитории и устранения дрейфа конфигураций.	С трудом реализует GitOps-подход для автоматической синхронизации состояния инфраструктуры с конфигурацией в Git-репозитории и устранения дрейфа конфигураций	Слабо реализует GitOps-подход для автоматической синхронизации состояния инфраструктуры с конфигурацией в Git-репозитории и устранения дрейфа конфигураций	В основном реализует GitOps-подход для автоматической синхронизации состояния инфраструктуры с конфигурацией в Git-репозитории и устранения дрейфа конфигураций	реализует GitOps-подход для автоматической синхронизации состояния инфраструктуры с конфигурацией в Git-репозитории и устранения дрейфа конфигураций
ПК-4				
ИД-1 $_{ПК-4}$. Определяет ключевые показатели качества обслуживания (SLI) для различных компонентов сервисов и автономных систем (доступность, задержка, точность, целостность данных).	С трудом определяет ключевые показатели качества обслуживания (SLI) для различных компонентов сервисов и автономных систем	Слабо определяет ключевые показатели качества обслуживания (SLI) для различных компонентов сервисов и автономных систем	В основном определяет ключевые показатели качества обслуживания (SLI) для различных компонентов сервисов и автономных систем	определяет ключевые показатели качества обслуживания (SLI) для различных компонентов сервисов и автономных систем
ИД-2 $_{ПК-4}$. Разрабатывает целевые уровни	С трудом разрабатывает целевые уровни	Слабо разрабатывает целевые уровни	В основном разрабатывает целевые	Разрабатывает целевые уровни обслуживания (SLO) на основе

обслуживания (SLO) на основе бизнес-требований и пользовательских ожиданий, согласовывая их с заинтересованными сторонами.	обслуживания (SLO) на основе бизнес-требований и пользовательских ожиданий	обслуживания (SLO) на основе бизнес-требований и пользовательских ожиданий	уровни обслуживания (SLO) на основе бизнес-требований и пользовательских ожиданий	бизнес-требований и пользовательских ожиданий
ИД-3 $_{ПК-4}$. Рассчитывает бюджет ошибок (Error Budget) как количественную меру допустимого снижения надежности и использует его для балансировки приоритетов разработки и эксплуатации.	С трудом рассчитывает бюджет ошибок (Error Budget) как количественную меру допустимого снижения надежности и использует его для балансировки приоритетов разработки и эксплуатации	Слабо рассчитывает бюджет ошибок (Error Budget) как количественную меру допустимого снижения надежности и использует его для балансировки приоритетов разработки и эксплуатации	В основном рассчитывает бюджет ошибок (Error Budget) как количественную меру допустимого снижения надежности и использует его для балансировки приоритетов разработки и эксплуатации	Рассчитывает бюджет ошибок (Error Budget) как количественную меру допустимого снижения надежности и использует его для балансировки приоритетов разработки и эксплуатации
ИД-4 $_{ПК-4}$. Настраивает системы сбора и мониторинга метрик (SLI) с построением дашбордов для визуализации текущего состояния надежности и оповещений при приближении к границам SLO.	С трудом настраивает системы сбора и мониторинга метрик (SLI) с построением дашбордов для визуализации текущего состояния надежности и оповещений при приближении к границам SLO	Слабо настраивает системы сбора и мониторинга метрик (SLI) с построением дашбордов для визуализации текущего состояния надежности и оповещений при приближении к границам SLO	В основном настраивает системы сбора и мониторинга метрик (SLI) с построением дашбордов для визуализации текущего состояния надежности и оповещений при приближении к границам SLO	Настраивает системы сбора и мониторинга метрик (SLI) с построением дашбордов для визуализации текущего состояния надежности и оповещений при приближении к границам SLO
ИД-5 $_{ПК-4}$. Анализирует динамику соблюдения SLO и потребления бюджета ошибок для выявления узких мест и прогнозирования потенциальных сбоев.	С трудом анализирует динамику соблюдения SLO и потребления бюджета ошибок для выявления узких мест и прогнозирования потенциальных сбоев	Слабо анализирует динамику соблюдения SLO и потребления бюджета ошибок для выявления узких мест и прогнозирования потенциальных сбоев	В основном анализирует динамику соблюдения SLO и потребления бюджета ошибок для выявления узких мест и прогнозирования потенциальных сбоев	Анализирует динамику соблюдения SLO и потребления бюджета ошибок для выявления узких мест и прогнозирования потенциальных сбоев
ИД-6 $_{ПК-4}$. Принимает обоснованные решения о замедлении или остановке внедрения нового функционала при исчерпании	С трудом принимает обоснованные решения о замедлении или остановке внедрения нового функционала при исчерпании бюджета ошибок для обеспечения	Слабо принимает обоснованные решения о замедлении или остановке внедрения нового функционала при исчерпании бюджета ошибок	В основном принимает обоснованные решения о замедлении или остановке внедрения нового функционала	Принимает обоснованные решения о замедлении или остановке внедрения нового функционала при исчерпании бюджета ошибок для обеспечения стабильности сервиса

бюджета ошибок для обеспечения стабильности сервиса.	стабильности сервиса	для обеспечения стабильности сервиса	при исчерпании бюджета ошибок для обеспечения стабильности сервиса	
ИД-7 $_{ПК-4}$. Интегрирует проверки влияния изменений на SLO в процессы непрерывной интеграции и доставки (CI/CD).	С трудом интегрирует проверки влияния изменений на SLO в процессы непрерывной интеграции и доставки	Слабо интегрирует проверки влияния изменений на SLO в процессы непрерывной интеграции и доставки	В основном интегрирует проверки влияния изменений на SLO в процессы непрерывной интеграции и доставки	Интегрирует проверки влияния изменений на SLO в процессы непрерывной интеграции и доставки
ПК-5				
ИД-1 $_{ПК-5}$. Определяет показатели устойчивого состояния (steady state) системы и формулирует гипотезы о ее поведении при различных типах отказов.	С трудом определяет показатели устойчивого состояния системы и формулирует гипотезы о ее поведении при различных типах отказов	Слабо определяет показатели устойчивого состояния системы и формулирует гипотезы о ее поведении при различных типах отказов	В основном определяет показатели устойчивого состояния системы и формулирует гипотезы о ее поведении при различных типах отказов откате	определяет показатели устойчивого состояния системы и формулирует гипотезы о ее поведении при различных типах отказов
ИД-2 $_{ПК-5}$. Проектирует эксперименты по внедрению сбоев (инфраструктурных, сетевых, нагрузочных) с контролируемой зоной поражения (blast radius).	С трудом проектирует эксперименты по внедрению сбоев с контролируемой зоной поражения	Слабо проектирует эксперименты по внедрению сбоев с контролируемой зоной поражения	В основном проектирует эксперименты по внедрению сбоев с контролируемой зоной поражения	проектирует эксперименты по внедрению сбоев с контролируемой зоной поражения
ИД-3 $_{ПК-5.}$ Проводит эксперименты по тестированию устойчивости с использованием инструментов chaos engineering в тестовых и production-средах.	С трудом проводит эксперименты по тестированию устойчивости с использованием инструментов chaos engineering в тестовых и production-средах	Слабо проводит эксперименты по тестированию устойчивости с использованием инструментов chaos engineering в тестовых и production-средах	В основном проводит эксперименты по тестированию устойчивости с использованием инструментов chaos engineering в тестовых и production-средах	проводит эксперименты по тестированию устойчивости с использованием инструментов chaos engineering в тестовых и production-средах
ИД-4 $_{ПК-5.}$ Анализирует результаты экспериментов, сопоставляя фактические метрики с ожидаемыми, и выявляет скрытые уязвимости и слабые места архитектуры.	С трудом анализирует результаты экспериментов, сопоставляя фактические метрики с ожидаемыми, и выявляет скрытые уязвимости и слабые места	Слабо анализирует результаты экспериментов, сопоставляя фактические метрики с ожидаемыми, и выявляет скрытые уязвимости и слабые места архитектуры	В основном анализирует результаты экспериментов, сопоставляя фактические метрики с ожидаемыми, и выявляет скрытые уязвимости и слабые места архитектуры	анализирует результаты экспериментов, сопоставляя фактические метрики с ожидаемыми, и выявляет скрытые уязвимости и слабые места архитектуры

	архитектуры			
ИД-5 $_{ПК-5.}$ Разрабатывает и внедряет корректирующие меры по итогам экспериментов (настройка таймаутов, ретраев, circuit breaker, оптимизация архитектуры).	С трудом разрабатывает и внедряет корректирующие меры по итогам экспериментов	Слабо разрабатывает и внедряет корректирующие меры по итогам экспериментов	В основном разрабатывает и внедряет корректирующие меры по итогам экспериментов	разрабатывает и внедряет корректирующие меры по итогам экспериментов
ИД-6 $_{ПК-5.}$ Проводит повторные эксперименты для подтверждения эффективности внедренных изменений и повышения отказоустойчивости.	С трудом проводит повторные эксперименты для подтверждения эффективности внедренных изменений и повышения отказоустойчивости	Слабо проводит повторные эксперименты для подтверждения эффективности внедренных изменений и повышения отказоустойчивости	В основном проводит повторные эксперименты для подтверждения эффективности внедренных изменений и повышения отказоустойчивости	проводит повторные эксперименты для подтверждения эффективности внедренных изменений и повышения отказоустойчивости
ИД-7 $_{ПК-5.}$ Интегрирует практики регулярного тестирования устойчивости (chaos engineering) в процессы разработки и эксплуатации, включая автоматизацию базовых экспериментов в CI/CD.	С трудом интегрирует практики регулярного тестирования устойчивости в процессы разработки и эксплуатации, включая автоматизацию базовых экспериментов в CI/CD	Слабо интегрирует практики регулярного тестирования устойчивости в процессы разработки и эксплуатации, включая автоматизацию базовых экспериментов в CI/CD	В основном интегрирует практики регулярного тестирования устойчивости в процессы разработки и эксплуатации, включая автоматизацию базовых экспериментов в CI/CD	интегрирует практики регулярного тестирования устойчивости в процессы разработки и эксплуатации, включая автоматизацию базовых экспериментов в CI/CD
ПК-6				
ИД-1 $_{ПК-6.}$ Анализирует хронику событий и технические метрики инцидента для восстановления полной и достоверной картины происшествия.	С трудом анализирует хронику событий и технические метрики инцидента для восстановления полной и достоверной картины происшествия	Слабо анализирует хронику событий и технические метрики инцидента для восстановления полной и достоверной картины происшествия	В основном анализирует хронику событий и технические метрики инцидента для восстановления полной и достоверной картины происшествия	Анализирует хронику событий и технические метрики инцидента для восстановления полной и достоверной картины происшествия
ИД-2 $_{ПК-6.}$ Применяет методы системного анализа (5 почему, диаграмма Исикавы, анализ графов) для выявления коренных (системных) причин сбоев.	С трудом применяет методы системного анализа (5 почему, диаграмма Исикавы, анализ графов) для выявления коренных	Слабо применяет методы системного анализа (5 почему, диаграмма Исикавы, анализ графов) для выявления коренных (системных) причин сбоев	В основном применяет методы системного анализа (5 почему, диаграмма Исикавы, анализ графов) для выявления коренных (системных) причин сбоев	Применяет методы системного анализа (5 почему, диаграмма Исикавы, анализ графов) для выявления коренных (системных) причин сбоев

	(системных) причин сбоев			
ИД-3 $_{ПК-6.}$ Разрабатывает комплекс превентивных мер и технических решений, направленных на устранение выявленных коренных причин и предотвращение повторения инцидентов.	С трудом разрабатывает комплекс превентивных мер и технических решений, направленных на устранение выявленных коренных причин и предотвращение повторения инцидентов	Слабо разрабатывает комплекс превентивных мер и технических решений, направленных на устранение выявленных коренных причин и предотвращение повторения инцидентов	В основном разрабатывает комплекс превентивных мер и технических решений, направленных на устранение выявленных коренных причин и предотвращение повторения инцидентов	Разрабатывает комплекс превентивных мер и технических решений, направленных на устранение выявленных коренных причин и предотвращение повторения инцидентов
ИД-4 $_{ПК-6.}$ Автоматизирует процессы обнаружения и реагирования на инциденты с использованием скриптов и средств оркестрации для сокращения ручного труда и времени простоя.	С трудом автоматизирует процессы обнаружения и реагирования на инциденты с использованием скриптов и средств оркестрации для сокращения ручного труда и времени простоя	Слабо автоматизирует процессы обнаружения и реагирования на инциденты с использованием скриптов и средств оркестрации для сокращения ручного труда и времени простоя	В основном автоматизирует процессы обнаружения и реагирования на инциденты с использованием скриптов и средств оркестрации для сокращения ручного труда и времени простоя	Автоматизирует процессы обнаружения и реагирования на инциденты с использованием скриптов и средств оркестрации для сокращения ручного труда и времени простоя
ИД-5 $_{ПК-6.}$ Документирует результаты анализа инцидентов в формате blameless postmortem и формирует базу знаний для повышения устойчивости операций.	С трудом документирует результаты анализа инцидентов в формате blameless postmortem и формирует базу знаний для повышения устойчивости операций	Слабо документирует результаты анализа инцидентов в формате blameless postmortem и формирует базу знаний для повышения устойчивости операций	В основном документирует результаты анализа инцидентов в формате blameless postmortem и формирует базу знаний для повышения устойчивости операций	Документирует результаты анализа инцидентов в формате blameless postmortem и формирует базу знаний для повышения устойчивости операций
ПК-8				
ИД-1 $_{ПК-8.}$ Внедряет инструменты статического анализа безопасности (SAST) в процесс разработки для автоматического выявления уязвимостей в исходном коде на этапе pull request'ов.	С трудом внедряет инструменты статического анализа безопасности (SAST) в процесс разработки для автоматического выявления уязвимостей в исходном коде на этапе pull request'ов	Слабо внедряет инструменты статического анализа безопасности (SAST) в процесс разработки для автоматического выявления уязвимостей в исходном коде на этапе pull request'ов	В основном внедряет инструменты статического анализа безопасности (SAST) в процесс разработки для автоматического выявления уязвимостей в исходном коде на этапе pull request'ов	Внедряет инструменты статического анализа безопасности (SAST) в процесс разработки для автоматического выявления уязвимостей в исходном коде на этапе pull request'ов
ИД-2 $_{ПК-8.}$ Организует автоматическое сканирование	С трудом организует автоматическое	Слабо организует автоматическое сканирование	В основном организует автоматическое	Организует автоматическое сканирование

зависимостей и сторонних библиотек (SCA) для контроля уязвимостей в цепочке поставки ПО.	сканирование зависимостей и сторонних библиотек (SCA) для контроля уязвимостей в цепочке поставки ПО	зависимостей и сторонних библиотек (SCA) для контроля уязвимостей в цепочке поставки ПО	сканирование зависимостей и сторонних библиотек (SCA) для контроля уязвимостей в цепочке поставки ПО	зависимостей и сторонних библиотек (SCA) для контроля уязвимостей в цепочке поставки ПО в
ИД-3 $_{\text{ПК-8.}}$ Обеспечивает безопасность контейнерной среды: сканирует образы на уязвимости, использует минимальные базовые образы и внедряет механизмы подписывания образов.	С трудом обеспечивает безопасность контейнерной среды: сканирует образы на уязвимости, использует минимальные базовые образы и внедряет механизмы подписывания образов	Слабо обеспечивает безопасность контейнерной среды: сканирует образы на уязвимости, использует минимальные базовые образы и внедряет механизмы подписывания образов	В основном обеспечивает безопасность контейнерной среды: сканирует образы на уязвимости, использует минимальные базовые образы и внедряет механизмы подписывания образов	Обеспечивает безопасность контейнерной среды: сканирует образы на уязвимости, использует минимальные базовые образы и внедряет механизмы подписывания образов
ИД-4 $_{\text{ПК-8.}}$ Проводит динамический анализ безопасности (DAST) запущенных приложений в тестовых и стейджинговых средах для выявления уязвимостей в рантайме.	С трудом проводит динамический анализ безопасности (DAST) запущенных приложений в тестовых и стейджинговых средах для выявления уязвимостей в рантайме	Слабо проводит динамический анализ безопасности (DAST) запущенных приложений в тестовых и стейджинговых средах для выявления уязвимостей в рантайме	В основном проводит динамический анализ безопасности (DAST) запущенных приложений в тестовых и стейджинговых средах для выявления уязвимостей в рантайме	Проводит динамический анализ безопасности (DAST) запущенных приложений в тестовых и стейджинговых средах для выявления уязвимостей в рантайме
ИД-5 $_{\text{ПК-8.}}$ Сканирует конфигурации инфраструктуры как кода (IaC) на соответствие политикам безопасности и лучшим практикам до их применения.	С трудом сканирует конфигурации инфраструктуры как кода (IaC) на соответствие политикам безопасности и лучшим практикам до их применения.	Слабо сканирует конфигурации инфраструктуры как кода (IaC) на соответствие политикам безопасности и лучшим практикам до их применения.	В основном сканирует конфигурации инфраструктуры как кода (IaC) на соответствие политикам безопасности и лучшим практикам до их применения.	Сканирует конфигурации инфраструктуры как кода (IaC) на соответствие политикам безопасности и лучшим практикам до их применения.
ИД-6 $_{\text{ПК-8.}}$ Интегрирует инструменты безопасности в CI/CD-пайплайны, настраивая автоматические шлюзы качества (quality gates), блокирующие поставку при обнаружении критических уязвимостей.	С трудом интегрирует инструменты безопасности в CI/CD-пайплайны, настраивая автоматические шлюзы качества (quality gates), блокирующие поставку при обнаружении критических уязвимостей	Слабо интегрирует инструменты безопасности в CI/CD-пайплайны, настраивая автоматические шлюзы качества (quality gates), блокирующие поставку при обнаружении критических уязвимостей	В основном интегрирует инструменты безопасности в CI/CD-пайплайны, настраивая автоматические шлюзы качества (quality gates), блокирующие поставку при обнаружении критических уязвимостей	Интегрирует инструменты безопасности в CI/CD-пайплайны, настраивая автоматические шлюзы качества (quality gates), блокирующие поставку при обнаружении критических уязвимостей

ИД-7 $_{ПК-8.}$ Мониторит состояние безопасности контейнеров и инфраструктуры в рантайме, выявляя инциденты и аномальное поведение.	С трудом мониторит состояние безопасности контейнеров и инфраструктуры в рантайме, выявляя инциденты и аномальное поведение.	Слабо мониторит состояние безопасности контейнеров и инфраструктуры в рантайме, выявляя инциденты и аномальное поведение.	В основном мониторит состояние безопасности контейнеров и инфраструктуры в рантайме, выявляя инциденты и аномальное поведение.	Мониторит состояние безопасности контейнеров и инфраструктуры в рантайме, выявляя инциденты и аномальное поведение.
ИД-8 $_{ПК-8}$. Обеспечивает непрерывное соответствие (continuous compliance) систем требованиям стандартов безопасности и регуляторным нормам через автоматизированный сбор доказательств и генерацию отчетов.	С трудом обеспечивает непрерывное соответствие (continuous compliance) систем требованиям стандартов безопасности и регуляторным нормам через автоматизированный сбор доказательств и генерацию отчетов	Слабо обеспечивает непрерывное соответствие (continuous compliance) систем требованиям стандартов безопасности и регуляторным нормам через автоматизированный сбор доказательств и генерацию отчетов	В основном обеспечивает непрерывное соответствие (continuous compliance) систем требованиям стандартов безопасности и регуляторным нормам через автоматизированный сбор доказательств и генерацию отчетов	Обеспечивает непрерывное соответствие (continuous compliance) систем требованиям стандартов безопасности и регуляторным нормам через автоматизированный сбор доказательств и генерацию отчетов
ПК-11				
ИД-1 $_{ПК-11}$. Моделирует сценарии атак на автономные системы, включая подмену сигнала (спуфинг), глушение (джамминг), перехват управления и внедрение вредоносного кода.	С трудом Моделирует сценарии атак на автономные системы, включая подмену сигнала (спуфинг), глушение (джамминг), перехват управления и внедрение вредоносного кода	Слабо Моделирует сценарии атак на автономные системы, включая подмену сигнала (спуфинг), глушение (джамминг), перехват управления и внедрение вредоносного кода	В основном Моделирует сценарии атак на автономные системы, включая подмену сигнала (спуфинг), глушение (джамминг), перехват управления и внедрение вредоносного кода	Моделирует сценарии атак на автономные системы, включая подмену сигнала (спуфинг), глушение (джамминг), перехват управления и внедрение вредоносного кода
ИД-2 $_{ПК-11}$. Внедряет механизмы шифрования и аутентификации в радиоканалах и сетях передачи данных для обеспечения конфиденциальности и подлинности команд управления.	С трудом Внедряет механизмы шифрования и аутентификации в радиоканалах и сетях передачи данных для обеспечения конфиденциальности и подлинности команд управления	Слабо Внедряет механизмы шифрования и аутентификации в радиоканалах и сетях передачи данных для обеспечения конфиденциальности и подлинности команд управления	В основном Внедряет механизмы шифрования и аутентификации в радиоканалах и сетях передачи данных для обеспечения конфиденциальности и подлинности команд управления	Внедряет механизмы шифрования и аутентификации в радиоканалах и сетях передачи данных для обеспечения конфиденциальности и подлинности команд управления
ИД-3 $_{ПК-11}$. Применяет методы безопасной разработки (SAST,	С трудом Применяет методы безопасной	Слабо Применяет методы безопасной разработки (SAST,	В основном Применяет методы безопасной	Применяет методы безопасной разработки (SAST, DAST) и контроля

DAST) и контроля версий для защиты бортового ПО от модификации и эксплуатации уязвимостей.	разработки (SAST, DAST) и контроля версий для защиты бортового ПО от модификации и эксплуатации уязвимостей	DAST) и контроля версий для защиты бортового ПО от модификации и эксплуатации уязвимостей	разработки (SAST, DAST) и контроля версий для защиты бортового ПО от модификации и эксплуатации уязвимостей я	версий для защиты бортового ПО от модификации и эксплуатации уязвимостей
ИД-4 $_{ПК-11}$. Реализует меры сетевой защиты (сегментация, межсетевое экранирование, системы обнаружения вторжений) для наземных центров управления полетами/движением.	С трудом Реализует меры сетевой защиты (сегментация, межсетевое экранирование, системы обнаружения вторжений) для наземных центров управления полетами/движением	Слабо Реализует меры сетевой защиты (сегментация, межсетевое экранирование, системы обнаружения вторжений) для наземных центров управления полетами/движением	В основном Реализует меры сетевой защиты (сегментация, межсетевое экранирование, системы обнаружения вторжений) для наземных центров управления полетами/движением	Реализует меры сетевой защиты (сегментация, межсетевое экранирование, системы обнаружения вторжений) для наземных центров управления полетами/движением
ИД-5 $_{ПК-11}$. Разрабатывает процедуры проверки происхождения и целостности комплектующих на этапе производства и поставки для исключения компрометации на аппаратном уровне.	С трудом Разрабатывает процедуры проверки происхождения и целостности комплектующих на этапе производства и поставки для исключения компрометации на аппаратном уровне	Слабо Разрабатывает процедуры проверки происхождения и целостности комплектующих на этапе производства и поставки для исключения компрометации на аппаратном уровне	В основном Разрабатывает процедуры проверки происхождения и целостности комплектующих на этапе производства и поставки для исключения компрометации на аппаратном уровне	Разрабатывает процедуры проверки происхождения и целостности комплектующих на этапе производства и поставки для исключения компрометации на аппаратном уровне ием
ИД-6 $_{ПК-11}$. Оценивает эффективность реализованных мер защиты через проведение пентестов и анализа защищенности автономной системы как единого комплекса.	С трудом Оценивает эффективность реализованных мер защиты через проведение пентестов и анализа защищенности автономной системы как единого комплекса	Слабо Оценивает эффективность реализованных мер защиты через проведение пентестов и анализа защищенности автономной системы как единого комплекса	В основном Оценивает эффективность реализованных мер защиты через проведение пентестов и анализа защищенности автономной системы как единого комплекса	Оценивает эффективность реализованных мер защиты через проведение пентестов и анализа защищенности автономной системы как единого комплекса

Критерии оценивания компетенций[*]

Оценка «отлично» выставляется студенту, если он описывает облачные ресурсы, сетевые компоненты и вычислительные мощности на декларативных языках конфигураций с использованием инструментов класса IaC; разрабатывает идемпотентные сценарии управления конфигурациями для настройки операционных систем; применяет системы контроля версий для хранения, версионирования и совместной работы над кодом инфраструктуры и конфигураций; интегрирует процессы развертывания инфраструктуры и применения конфигураций в пайплайны непрерывной интеграции и доставки; обеспечивает безопасность управления инфраструктурой через код: внедряет

сканирование кода на уязвимости, управляет секретами и реализует принцип минимальных привилегий; реализует GitOps-подход для автоматической синхронизации состояния инфраструктуры с конфигурацией в Git-репозитории и устранения дрейфа конфигураций; определяет ключевые показатели качества обслуживания (SLI) для различных компонентов сервисов и автономных систем; Разрабатывает целевые уровни обслуживания (SLO) на основе бизнес-требований и пользовательских ожиданий; Рассчитывает бюджет ошибок (Error Budget) как количественную меру допустимого снижения надежности и использует его для балансировки приоритетов разработки и эксплуатации; Настраивает системы сбора и мониторинга метрик (SLI) с построением дашбордов для визуализации текущего состояния надежности и оповещений при приближении к границам SLO; Анализирует динамику соблюдения SLO и потребления бюджета ошибок для выявления узких мест и прогнозирования потенциальных сбоев; Принимает обоснованные решения о замедлении или остановке внедрения нового функционала при исчерпании бюджета ошибок для обеспечения стабильности сервиса; Интегрирует проверки влияния изменений на SLO в процессы непрерывной интеграции и доставки; определяет показатели устойчивого состояния системы и формулирует гипотезы о ее поведении при различных типах отказов; проектирует эксперименты по внедрению сбоев с контролируемой зоной поражения; проводит эксперименты по тестированию устойчивости с использованием инструментов chaos engineering в тестовых и production-средах; анализирует результаты экспериментов, сопоставляя фактические метрики с ожидаемыми, и выявляет скрытые уязвимости и слабые места архитектуры; разрабатывает и внедряет корректирующие меры по итогам экспериментов; проводит повторные эксперименты для подтверждения эффективности внедренных изменений и повышения отказоустойчивости; интегрирует практики регулярного тестирования устойчивости в процессы разработки и эксплуатации, включая автоматизацию базовых экспериментов в CI/CD; Анализирует хронику событий и технические метрики инцидента для восстановления полной и достоверной картины происшествия; Применяет методы системного анализа (5 почему, диаграмма Исикавы, анализ графов) для выявления коренных (системных) причин сбоев; Разрабатывает комплекс превентивных мер и технических решений, направленных на устранение выявленных коренных причин и предотвращение повторения инцидентов; Автоматизирует процессы обнаружения и реагирования на инциденты с использованием скриптов и средств оркестрации для сокращения ручного труда и времени простоя; Документирует результаты анализа инцидентов в формате blameless postmortem и формирует базу знаний для повышения устойчивости операций; Внедряет инструменты статического анализа безопасности (SAST) в процесс разработки для автоматического выявления уязвимостей в исходном коде на этапе pull request'ов; Организует автоматическое сканирование зависимостей и сторонних библиотек (SCA) для контроля уязвимостей в цепочке поставки ПО; Обеспечивает безопасность контейнерной среды: сканирует образы на уязвимости, использует минимальные базовые образы и внедряет механизмы подписывания образов; Проводит динамический анализ безопасности (DAST) запущенных приложений в тестовых и стейджинговых средах для выявления уязвимостей в рантайме; Сканирует конфигурации инфраструктуры как кода (IaC) на соответствие политикам безопасности и лучшим практикам до их применения; Интегрирует инструменты безопасности в CI/CD-пайплайны, настраивая автоматические шлюзы качества (quality gates), блокирующие поставку при обнаружении критических уязвимостей; Мониторит состояние безопасности контейнеров и инфраструктуры в рантайме, выявляя инциденты и аномальное поведение; Обеспечивает непрерывное соответствие (continuous compliance) систем требованиям стандартов безопасности и регуляторным нормам через автоматизированный сбор доказательств и генерацию отчетов; Моделирует сценарии атак на автономные системы, включая подмену сигнала (спуфинг), глушение (джамминг), перехват управления и внедрение вредоносного кода; Внедряет механизмы шифрования и аутентификации в

радиоканалах и сетях передачи данных для обеспечения конфиденциальности и подлинности команд управления; Применяет методы безопасной разработки (SAST, DAST) и контроля версий для защиты бортового ПО от модификации и эксплуатации уязвимостей; Реализует меры сетевой защиты (сегментация, межсетевое экранирование, системы обнаружения вторжений) для наземных центров управления полетами/движением; Разрабатывает процедуры проверки происхождения и целостности комплектующих на этапе производства и поставки для исключения компрометации на аппаратном уровне; Оценивает эффективность реализованных мер защиты через проведение пентестов и анализа защищенности автономной системы как единого комплекса.

Оценка «хорошо» выставляется студенту, если он в В основном описывает облачные ресурсы, сетевые компоненты и вычислительные мощности на декларативных языках конфигураций с использованием инструментов класса IaC; В основном разрабатывает идемпотентные сценарии управления конфигурациями для настройки операционных систем; В основном применяет системы контроля версий для хранения, версионирования и совместной работы над кодом инфраструктуры и конфигураций; В основном интегрирует процессы развертывания инфраструктуры и применения конфигураций в пайплайны непрерывной интеграции и доставки; В основном обеспечивает безопасность управления инфраструктурой через код: внедряет сканирование кода на уязвимости, управляет секретами и реализует принцип минимальных привилегий; В основном реализует GitOps-подход для автоматической синхронизации состояния инфраструктуры с конфигурацией в Git-репозитории и устранения дрейфа конфигураций; В основном определяет ключевые показатели качества обслуживания (SLI) для различных компонентов сервисов и автономных систем; В основном разрабатывает целевые уровни обслуживания (SLO) на основе бизнес-требований и пользовательских ожиданий; В основном рассчитывает бюджет ошибок (Error Budget) как количественную меру допустимого снижения надежности и использует его для балансировки приоритетов разработки и эксплуатации; В основном настраивает системы сбора и мониторинга метрик (SLI) с построением дашбордов для визуализации текущего состояния надежности и оповещений при приближении к границам SLO; В основном анализирует динамику соблюдения SLO и потребления бюджета ошибок для выявления узких мест и прогнозирования потенциальных сбоев; В основном принимает обоснованные решения о замедлении или остановке внедрения нового функционала при исчерпании бюджета ошибок для обеспечения стабильности сервиса; В основном интегрирует проверки влияния изменений на SLO в процессы непрерывной интеграции и доставки; В основном определяет показатели устойчивого состояния системы и формулирует гипотезы о ее поведении при различных типах отказов откате; В основном проектирует эксперименты по внедрению сбоев с контролируемой зоной поражения; В основном проводит эксперименты по тестированию устойчивости с использованием инструментов chaos engineering в тестовых и production-средах; В основном анализирует результаты экспериментов, сопоставляя фактические метрики с ожидаемыми, и выявляет скрытые уязвимости и слабые места архитектуры; В основном разрабатывает и внедряет корректирующие меры по итогам экспериментов; В основном проводит повторные эксперименты для подтверждения эффективности внедренных изменений и повышения отказоустойчивости; В основном интегрирует практики регулярного тестирования устойчивости в процессы разработки и эксплуатации, включая автоматизацию базовых экспериментов в CI/CD; В основном анализирует хронику событий и технические метрики инцидента для восстановления полной и достоверной картины происшествия; В основном применяет методы системного анализа (5 почему, диаграмма Исикавы, анализ графов) для выявления коренных (системных) причин сбоев; В основном разрабатывает комплекс превентивных мер и технических решений, направленных на устранение выявленных коренных причин и предотвращение повторения инцидентов; В основном автоматизирует

процессы обнаружения и реагирования на инциденты с использованием скриптов и средств оркестрации для сокращения ручного труда и времени простоя; В основном документирует результаты анализа инцидентов в формате blameless postmortem и формирует базу знаний для повышения устойчивости операций; В основном внедряет инструменты статического анализа безопасности (SAST) в процесс разработки для автоматического выявления уязвимостей в исходном коде на этапе pull request'ов; В основном организует автоматическое сканирование зависимостей и сторонних библиотек (SCA) для контроля уязвимостей в цепочке поставки ПО; В основном обеспечивает безопасность контейнерной среды: сканирует образы на уязвимости, использует минимальные базовые образы и внедряет механизмы подписывания образов; В основном проводит динамический анализ безопасности (DAST) запущенных приложений в тестовых и стейджинговых средах для выявления уязвимостей в рантайме; В основном сканирует конфигурации инфраструктуры как кода (IaC) на соответствие политикам безопасности и лучшим практикам до их применения; В основном интегрирует инструменты безопасности в CI/CD-пайплайны, настраивая автоматические шлюзы качества (quality gates), блокирующие поставку при обнаружении критических уязвимостей; В основном мониторит состояние безопасности контейнеров и инфраструктуры в рантайме, выявляя инциденты и аномальное поведение; В основном обеспечивает непрерывное соответствие (continuous compliance) систем требованиям стандартов безопасности и регуляторным нормам через автоматизированный сбор доказательств и генерацию отчетов; В основном Моделирует сценарии атак на автономные системы, включая подмену сигнала (спуфинг), глушение (джамминг), перехват управления и внедрение вредоносного кода; В основном Внедряет механизмы шифрования и аутентификации в радиоканалах и сетях передачи данных для обеспечения конфиденциальности и подлинности команд управления; В основном Применяет методы безопасной разработки (SAST, DAST) и контроля версий для защиты бортового ПО от модификации и эксплуатации уязвимостей; В основном Реализует меры сетевой защиты (сегментация, межсетевое экранирование, системы обнаружения вторжений) для наземных центров управления полетами/движением; В основном Разрабатывает процедуры проверки происхождения и целостности комплектующих на этапе производства и поставки для исключения компрометации на аппаратном уровне; В основном Оценивает эффективность реализованных мер защиты через проведение пентестов и анализа защищенности автономной системы как единого комплекса.

Оценка «удовлетворительно» выставляется студенту, если он Слабо описывает облачные ресурсы, сетевые компоненты и вычислительные мощности на декларативных языках конфигураций с использованием инструментов класса IaC; Слабо разрабатывает идемпотентные сценарии управления конфигурациями для настройки операционных систем; Слабо применяет системы контроля версий для хранения, версионирования и совместной работы над кодом инфраструктуры и конфигураций; Слабо интегрирует процессы развертывания инфраструктуры и применения конфигураций в пайплайны непрерывной интеграции и доставки; Слабо обеспечивает безопасность управления инфраструктурой через код: внедряет сканирование кода на уязвимости, управляет секретами и реализует принцип минимальных привилегий; Слабо реализует GitOps-подход для автоматической синхронизации состояния инфраструктуры с конфигурацией в Git-репозитории и устранения дрейфа конфигураций; Слабо определяет ключевые показатели качества обслуживания (SLI) для различных компонентов сервисов и автономных систем; Слабо разрабатывает целевые уровни обслуживания (SLO) на основе бизнес-требований и пользовательских ожиданий; Слабо рассчитывает бюджет ошибок (Error Budget) как количественную меру допустимого снижения надежности и использует его для балансировки приоритетов разработки и эксплуатации; Слабо настраивает системы сбора и мониторинга метрик (SLI) с построением дашбордов для визуализации текущего состояния надежности и оповещений при приближении к границам SLO; Слабо

анализирует динамику соблюдения SLO и потребления бюджета ошибок для выявления узких мест и прогнозирования потенциальных сбоев; Слабо принимает обоснованные решения о замедлении или остановке внедрения нового функционала при исчерпании бюджета ошибок для обеспечения стабильности сервиса; Слабо интегрирует проверки влияния изменений на SLO в процессы непрерывной интеграции и доставки; Слабо определяет показатели устойчивого состояния системы и формулирует гипотезы о ее поведении при различных типах отказов; Слабо проектирует эксперименты по внедрению сбоев с контролируемой зоной поражения; Слабо проводит эксперименты по тестированию устойчивости с использованием инструментов chaos engineering в тестовых и production-средах; Слабо анализирует результаты экспериментов, сопоставляя фактические метрики с ожидаемыми, и выявляет скрытые уязвимости и слабые места архитектуры; Слабо разрабатывает и внедряет корректирующие меры по итогам экспериментов; Слабо проводит повторные эксперименты для подтверждения эффективности внедренных изменений и повышения отказоустойчивости; Слабо интегрирует практики регулярного тестирования устойчивости в процессы разработки и эксплуатации, включая автоматизацию базовых экспериментов в CI/CD; Слабо анализирует хронику событий и технические метрики инцидента для восстановления полной и достоверной картины происшествия; Слабо применяет методы системного анализа (5 почему, диаграмма Исикавы, анализ графов) для выявления коренных (системных) причин сбоев; Слабо разрабатывает комплекс превентивных мер и технических решений, направленных на устранение выявленных коренных причин и предотвращение повторения инцидентов; Слабо автоматизирует процессы обнаружения и реагирования на инциденты с использованием скриптов и средств оркестрации для сокращения ручного труда и времени простоя; Слабо документирует результаты анализа инцидентов в формате blameless postmortem и формирует базу знаний для повышения устойчивости операций; Слабо внедряет инструменты статического анализа безопасности (SAST) в процесс разработки для автоматического выявления уязвимостей в исходном коде на этапе pull request'ов; Слабо организует автоматическое сканирование зависимостей и сторонних библиотек (SCA) для контроля уязвимостей в цепочке поставки ПО; Слабо обеспечивает безопасность контейнерной среды: сканирует образы на уязвимости, использует минимальные базовые образы и внедряет механизмы подписывания образов; Слабо проводит динамический анализ безопасности (DAST) запущенных приложений в тестовых и стейджинговых средах для выявления уязвимостей в рантайме; Слабо сканирует конфигурации инфраструктуры как кода (IaC) на соответствие политикам безопасности и лучшим практикам до их применения; Слабо интегрирует инструменты безопасности в CI/CD-пайплайны, настраивая автоматические шлюзы качества (quality gates), блокирующие поставку при обнаружении критических уязвимостей; Слабо мониторит состояние безопасности контейнеров и инфраструктуры в рантайме, выявляя инциденты и аномальное поведение; Слабо обеспечивает непрерывное соответствие (continuous compliance) систем требованиям стандартов безопасности и регуляторным нормам через автоматизированный сбор доказательств и генерацию отчетов; Слабо Моделирует сценарии атак на автономные системы, включая подмену сигнала (спуфинг), глушение (джамминг), перехват управления и внедрение вредоносного кода; Слабо Внедряет механизмы шифрования и аутентификации в радиоканалах и сетях передачи данных для обеспечения конфиденциальности и подлинности команд управления; Слабо Применяет методы безопасной разработки (SAST, DAST) и контроля версий для защиты бортового ПО от модификации и эксплуатации уязвимостей; Слабо Реализует меры сетевой защиты (сегментация, межсетевое экранирование, системы обнаружения вторжений) для наземных центров управления полетами/движением; Слабо Разрабатывает процедуры проверки происхождения и целостности комплектующих на этапе производства и поставки для исключения компрометации на аппаратном уровне;

Слабо Оценивает эффективность реализованных мер защиты через проведение пентестов и анализа защищенности автономной системы как единого комплекса.

Оценка «неудовлетворительно» выставляется студенту, если он С трудом описывает облачные ресурсы, сетевые компоненты и вычислительные мощности на декларативных языках конфигураций с использованием инструментов класса IaC; С трудом разрабатывает идемпотентные сценарии управления конфигурациями для настройки операционных систем; С трудом применяет системы контроля версий для хранения, версионирования и совместной работы над кодом инфраструктуры и конфигураций; С трудом интегрирует процессы развертывания инфраструктуры и применения конфигураций в пайплайны непрерывной интеграции и доставки; С трудом обеспечивает безопасность управления инфраструктурой через код: внедряет сканирование кода на уязвимости, управляет секретами и реализует принцип минимальных привилегий; С трудом реализует GitOps-подход для автоматической синхронизации состояния инфраструктуры с конфигурацией в Git-репозитории и устранения дрейфа конфигураций; С трудом определяет ключевые показатели качества обслуживания (SLI) для различных компонентов сервисов и автономных систем; С трудом разрабатывает целевые уровни обслуживания (SLO) на основе бизнес-требований и пользовательских ожиданий; С трудом рассчитывает бюджет ошибок (Error Budget) как количественную меру допустимого снижения надежности и использует его для балансировки приоритетов разработки и эксплуатации; С трудом настраивает системы сбора и мониторинга метрик (SLI) с построением дашбордов для визуализации текущего состояния надежности и оповещений при приближении к границам SLO; С трудом анализирует динамику соблюдения SLO и потребления бюджета ошибок для выявления узких мест и прогнозирования потенциальных сбоев; С трудом принимает обоснованные решения о замедлении или остановке внедрения нового функционала при исчерпании бюджета ошибок для обеспечения стабильности сервиса; С трудом интегрирует проверки влияния изменений на SLO в процессы непрерывной интеграции и доставки; С трудом определяет показатели устойчивого состояния системы и формулирует гипотезы о ее поведении при различных типах отказов; С трудом проектирует эксперименты по внедрению сбоев с контролируемой зоной поражения; С трудом проводит эксперименты по тестированию устойчивости с использованием инструментов chaos engineering в тестовых и production-средах; С трудом анализирует результаты экспериментов, сопоставляя фактические метрики с ожидаемыми, и выявляет скрытые уязвимости и слабые места архитектуры; С трудом разрабатывает и внедряет корректирующие меры по итогам экспериментов; С трудом проводит повторные эксперименты для подтверждения эффективности внедренных изменений и повышения отказоустойчивости; С трудом интегрирует практики регулярного тестирования устойчивости в процессы разработки и эксплуатации, включая автоматизацию базовых экспериментов в CI/CD; С трудом анализирует хронику событий и технические метрики инцидента для восстановления полной и достоверной картины происшествия; С трудом применяет методы системного анализа (5 почему, диаграмма Исикавы, анализ графов) для выявления коренных (системных) причин сбоев; С трудом разрабатывает комплекс превентивных мер и технических решений, направленных на устранение выявленных коренных причин и предотвращение повторения инцидентов; С трудом автоматизирует процессы обнаружения и реагирования на инциденты с использованием скриптов и средств оркестрации для сокращения ручного труда и времени простоя; С трудом документирует результаты анализа инцидентов в формате blameless postmortem и формирует базу знаний для повышения устойчивости операций; С трудом внедряет инструменты статического анализа безопасности (SAST) в процесс разработки для автоматического выявления уязвимостей в исходном коде на этапе pull request'ов; С трудом организует автоматическое сканирование зависимостей и сторонних библиотек (SCA) для контроля уязвимостей в цепочке поставки ПО; С трудом обеспечивает безопасность контейнерной среды: сканирует образы на уязвимости,

использует минимальные базовые образы и внедряет механизмы подписывания образов; С трудом проводит динамический анализ безопасности (DAST) запущенных приложений в тестовых и стейджинговых средах для выявления уязвимостей в рантайме; С трудом сканирует конфигурации инфраструктуры как кода (IaC) на соответствие политикам безопасности и лучшим практикам до их применения; С трудом интегрирует инструменты безопасности в CI/CD-пайплайны, настраивая автоматические шлюзы качества (quality gates), блокирующие поставку при обнаружении критических уязвимостей; С трудом мониторит состояние безопасности контейнеров и инфраструктуры в рантайме, выявляя инциденты и аномальное поведение; С трудом обеспечивает непрерывное соответствие (continuous compliance) систем требованиям стандартов безопасности и регуляторным нормам через автоматизированный сбор доказательств и генерацию отчетов; С трудом Моделирует сценарии атак на автономные системы, включая подмену сигнала (спуфинг), глушение (джамминг), перехват управления и внедрение вредоносного кода; С трудом Внедряет механизмы шифрования и аутентификации в радиоканалах и сетях передачи данных для обеспечения конфиденциальности и подлинности команд управления; С трудом Применяет методы безопасной разработки (SAST, DAST) и контроля версий для защиты бортового ПО от модификации и эксплуатации уязвимостей; С трудом Реализует меры сетевой защиты (сегментация, межсетевое экранирование, системы обнаружения вторжений) для наземных центров управления полетами/движением; С трудом Разрабатывает процедуры проверки происхождения и целостности комплектующих на этапе производства и поставки для исключения компрометации на аппаратном уровне; С трудом Оценивает эффективность реализованных мер защиты через проведение пентестов и анализа защищенности автономной системы как единого комплекса.

2. Оценочные средства по эксплуатационной практике

2.1. Задания, позволяющие оценить знания, полученные на практике

<table>
<tr><td colspan="2">Формируемые компетенции, индикаторы</td><td colspan="2" rowspan="2">Формулировка задания</td></tr>
<tr><td>Код компетенции</td><td>Формулировки</td></tr>
<tr><td>ИД-1 ПК-2.</td><td>Описывает облачные ресурсы, сетевые компоненты и вычислительные мощности на декларативных языках конфигураций (HCL, YAML, JSON) с использованием инструментов класса IaC (Terraform, CloudFormation, Pulumi).</td><td>Задание 1</td><td rowspan="3">применяет принципы "Инфраструктура как код" и "Конфигурация как код" для автоматизированного, идемпотентного, безопасного и версионируемого управления облачными, гибридными средами и конфигурациями парков автономных устройств с использованием современных инструментов</td></tr>
<tr><td>ИД-2 ПК-2.</td><td>Разрабатывает идемпотентные сценарии управления конфигурациями (Ansible, Puppet, Chef) для настройки операционных систем, прикладного ПО и бортового программного обеспечения автономных устройств</td><td>Задание 2</td></tr>
<tr><td>ИД-3 ПК-2.</td><td>Применяет системы контроля версий (Git) для хранения, версионирования и совместной работы над кодом инфраструктуры и</td><td>Задание 3</td></tr>
</table>

	конфигураций		
ИД-4 $_{ПК-2.}$	Интегрирует процессы развертывания инфраструктуры и применения конфигураций в пайплайны непрерывной интеграции и доставки (CI/CD)	Задание 4	
ИД-5 $_{ПК-2.}$	Обеспечивает безопасность управления инфраструктурой через код: внедряет сканирование кода на уязвимости, управляет секретами и реализует принцип минимальных привилегий	Задание 5	
ИД-6 $_{ПК-2.}$	Реализует GitOps-подход для автоматической синхронизации состояния инфраструктуры с конфигурацией в Git-репозитории и устранения дрейфа конфигураций	Задание 6	
ИД-1 $_{ПК-4.}$	Определяет ключевые показатели качества обслуживания (SLI) для различных компонентов сервисов и автономных систем (доступность, задержка, точность, целостность данных).	Задание 7	количественно определяет, измеряет, анализирует и управляет надежностью сервисов и автономных систем через разработку, мониторинг и соблюдение SLO, SLI, расчет и использование Error Budget
ИД-2 $_{ПК-4.}$	Разрабатывает целевые уровни обслуживания (SLO) на основе бизнес-требований и пользовательских ожиданий, согласовывая их с заинтересованными сторонами.	Задание 8	
ИД-3 $_{ПК-4.}$	Рассчитывает бюджет ошибок (Error Budget) как количественную меру допустимого снижения надежности и использует его для балансировки приоритетов разработки и эксплуатации.	Задание 9	
ИД-4 $_{ПК-4.}$	Настраивает системы сбора и мониторинга метрик (SLI) с построением дашбордов для визуализации текущего состояния надежности и оповещений при приближении к границам SLO.	Задание 10	

ИД-5 $_{ПК-4.}$	Анализирует динамику соблюдения SLO и потребления бюджета ошибок для выявления узких мест и прогнозирования потенциальных сбоев.	Задание 11	
ИД-6 $_{ПК-4.}$	Принимает обоснованные решения о замедлении или остановке внедрения нового функционала при исчерпании бюджета ошибок для обеспечения стабильности сервиса.	Задание 12	
ИД-7 $_{ПК-4.}$	Интегрирует проверки влияния изменений на SLO в процессы непрерывной интеграции и доставки (CI/CD).	Задание 13	
ИД-1 $_{ПК-5.}$	Определяет показатели устойчивого состояния (steady state) системы и формулирует гипотезы о ее поведении при различных типах отказов	Задание 14	проектирует, планирует и проводит эксперименты по обеспечению и проверке устойчивости систем
ИД-2 $_{ПК-5.}$	Проектирует эксперименты по внедрению сбоев (инфраструктурных, сетевых, нагрузочных) с контролируемой зоной поражения (blast radius).	Задание 15	
ИД-3 $_{ПК-5.}$	Проводит эксперименты по тестированию устойчивости с использованием инструментов chaos engineering в тестовых и production-средах.	Задание 16	
ИД-4 $_{ПК-5.}$	Анализирует результаты экспериментов, сопоставляя фактические метрики с ожидаемыми, и выявляет скрытые уязвимости и слабые места архитектуры	Задание 17	
ИД-5 $_{ПК-5.}$	Разрабатывает и внедряет корректирующие меры по итогам экспериментов (настройка таймаутов, ретраев, circuit breaker, оптимизация архитектуры).	Задание 18	анализирует их результаты, формулирует и внедряет корректирующие меры для повышения отказоустойчивости и устойчивости к сбоям
ИД-6 $_{ПК-5.}$	Проводит повторные эксперименты для подтверждения эффективности внедренных изменений и повышения отказоустойчивости	Задание 19	

ИД-7 $_{ПК-5.}$	Интегрирует практики регулярного тестирования устойчивости (chaos engineering) в процессы разработки и эксплуатации, включая автоматизацию базовых экспериментов в CI/CD.	Задание 20	
ИД-1 $_{ПК-6.}$	Анализирует хронику событий и технические метрики инцидента для восстановления полной и достоверной картины происшествия.	Задание 21	проводит глубокий анализ инцидентов по методологии blameless postmortem, выявляет системные коренные причины, разрабатывать и внедрять превентивные меры
ИД-2 $_{ПК-6.}$	Применяет методы системного анализа (5 почему, диаграмма Исикавы, анализ графов) для выявления коренных (системных) причин сбоев.	Задание 22	
ИД-3 $_{ПК-6.}$	Разрабатывает комплекс превентивных мер и технических решений, направленных на устранение выявленных коренных причин и предотвращение повторения инцидентов.	Задание 23	
ИД-4 $_{ПК-6.}$	Автоматизирует процессы обнаружения и реагирования на инциденты с использованием скриптов и средств оркестрации для сокращения ручного труда и времени простоя.	Задание 24	Автоматизирует процессы реагирования для устранения рутины и повышения устойчивости операций
ИД-5 $_{ПК-6.}$	Документирует результаты анализа инцидентов в формате blameless postmortem и формирует базу знаний для повышения устойчивости операций.	Задание 25	
ИД-1 $_{ПК-8.}$	Внедряет инструменты статического анализа безопасности (SAST) в процесс разработки для автоматического выявления уязвимостей в исходном коде на этапе pull request'ов.	Задание 26	Интегрирует практики безопасности на всех этапах жизненного цикла, включая статический и динамический анализ кода, анализ уязвимостей зависимостей, безопасность контейнеров и облачной инфраструктуры, обеспечивая непрерывное соответствие требованиям безопасности
ИД-2 $_{ПК-8.}$	Организует автоматическое сканирование зависимостей и сторонних библиотек (SCA) для контроля уязвимостей в цепочке поставки ПО.	Задание 27	

ИД-3 $_{ПК-8.}$	Обеспечивает безопасность контейнерной среды: сканирует образы на уязвимости, использует минимальные базовые образы и внедряет механизмы подписывания образов.	Задание 28	
ИД-4 $_{ПК-8.}$	Проводит динамический анализ безопасности (DAST) запущенных приложений в тестовых и стейджинговых средах для выявления уязвимостей в рантайме.	Задание 29	
ИД-5 $_{ПК-8.}$	Сканирует конфигурации инфраструктуры как кода (IaC) на соответствие политикам безопасности и лучшим практикам до их применения.	Задание 30	
ИД-6 $_{ПК-8.}$	Интегрирует инструменты безопасности в CI/CD-пайплайны, настраивая автоматические шлюзы качества (quality gates), блокирующие поставку при обнаружении критических уязвимостей.	Задание 31	
ИД-7 $_{ПК-8.}$	Мониторит состояние безопасности контейнеров и инфраструктуры в рантайме, выявляя инциденты и аномальное поведение.	Задание 32	
ИД-8 $_{ПК-8.}$	Обеспечивает непрерывное соответствие (continuous compliance) систем требованиям стандартов безопасности и регуляторным нормам через автоматизированный сбор доказательств и генерацию отчетов.	Задание 33	
ИД-1 $_{ПК-11.}$	Моделирует сценарии атак на автономные системы, включая подмену сигнала (спуфинг), глушение (джамминг), перехват управления и внедрение вредоносного кода.	Задание 34	проводит анализ угроз кибербезопасности для автономных систем, проектирует и внедряет меры защиты на уровнях канала связи, бортового ПО, наземной инфраструктуры управления и цепочек поставок, обеспечивая целостность,
ИД-2 $_{ПК-11.}$	Внедряет механизмы шифрования и аутентификации в	Задание 35	

	радиоканалах и сетях передачи данных для обеспечения конфиденциальности и подлинности команд управления.		конфиденциальность и доступность системы
ИД-3 $_{ПК-11.}$	Применяет методы безопасной разработки (SAST, DAST) и контроля версий для защиты бортового ПО от модификации и эксплуатации уязвимостей.	Задание 36	
ИД-4 $_{ПК-11}$	. Реализует меры сетевой защиты (сегментация, межсетевое экранирование, системы обнаружения вторжений) для наземных центров управления полетами/движением.	Задание 37	
ИД-5 $_{ПК-11.}$	Разрабатывает процедуры проверки происхождения и целостности комплектующих на этапе производства и поставки для исключения компрометации на аппаратном уровне.	Задание 38	
ИД-6 $_{ПК-11.}$	Оценивает эффективность реализованных мер защиты через проведение пентестов и анализа защищенности автономной системы как единого комплекса.	Задание 39	

2.2. Задания, позволяющие оценить умения и навыки, полученные на практике

Формируемые компетенции, индикаторы		Формулировка задания	
Код компетенции	Формулировки		
ИД-1 $_{ПК-2.}$	Описывает облачные ресурсы, сетевые компоненты и вычислительные мощности на декларативных языках конфигураций (HCL, YAML, JSON) с использованием инструментов класса IaC (Terraform, CloudFormation, Pulumi).	Задание 1	применяет принципы "Инфраструктура как код" и "Конфигурация как код" для автоматизированного, идемпотентного, безопасного и версионируемого управления облачными, гибридными средами и конфигурациями парков автономных устройств с использованием современных инструментов
ИД-2 $_{ПК-2.}$	Разрабатывает идемпотентные сценарии управления конфигурациями (Ansible, Puppet, Chef) для настройки операционных систем,	Задание 2	

	прикладного ПО и бортового программного обеспечения автономных устройств		
ИД-3 $_{ПК-2.}$	Применяет системы контроля версий (Git) для хранения, версионирования и совместной работы над кодом инфраструктуры и конфигураций	Задание 3	
ИД-4 $_{ПК-2.}$	Интегрирует процессы развертывания инфраструктуры и применения конфигураций в пайплайны непрерывной интеграции и доставки (CI/CD)	Задание 4	
ИД-5 $_{ПК-2.}$	Обеспечивает безопасность управления инфраструктурой через код: внедряет сканирование кода на уязвимости, управляет секретами и реализует принцип минимальных привилегий	Задание 5	
ИД-6 $_{ПК-2.}$	Реализует GitOps-подход для автоматической синхронизации состояния инфраструктуры с конфигурацией в Git-репозитории и устранения дрейфа конфигураций	Задание 6	
ИД-1 $_{ПК-4.}$	Определяет ключевые показатели качества обслуживания (SLI) для различных компонентов сервисов и автономных систем (доступность, задержка, точность, целостность данных).	Задание 7	количественно определяет, измеряет, анализирует и управляет надежностью сервисов и автономных систем через разработку, мониторинг и соблюдение SLO, SLI, расчет и использование Error Budget
ИД-2 $_{ПК-4.}$	Разрабатывает целевые уровни обслуживания (SLO) на основе бизнес-требований и пользовательских ожиданий, согласовывая их с заинтересованными сторонами.	Задание 8	
ИД-3 $_{ПК-4.}$	Рассчитывает бюджет ошибок (Error Budget) как количественную меру допустимого снижения надежности и использует его для балансировки приоритетов разработки и эксплуатации.	Задание 9	

ИД-4 $_{ПК-4.}$	Настраивает системы сбора и мониторинга метрик (SLI) с построением дашбордов для визуализации текущего состояния надежности и оповещений при приближении к границам SLO.	Задание 10	
ИД-5 $_{ПК-4.}$	Анализирует динамику соблюдения SLO и потребления бюджета ошибок для выявления узких мест и прогнозирования потенциальных сбоев.	Задание 11	
ИД-6 $_{ПК-4.}$	Принимает обоснованные решения о замедлении или остановке внедрения нового функционала при исчерпании бюджета ошибок для обеспечения стабильности сервиса.	Задание 12	
ИД-7 $_{ПК-4.}$	Интегрирует проверки влияния изменений на SLO в процессы непрерывной интеграции и доставки (CI/CD).	Задание 13	
ИД-1 $_{ПК-5.}$	Определяет показатели устойчивого состояния (steady state) системы и формулирует гипотезы о ее поведении при различных типах отказов	Задание 14	проектирует, планирует и проводит эксперименты по обеспечению и проверке устойчивости систем
ИД-2 $_{ПК-5.}$	Проектирует эксперименты по внедрению сбоев (инфраструктурных, сетевых, нагрузочных) с контролируемой зоной поражения (blast radius).	Задание 15	
ИД-3 $_{ПК-5.}$	Проводит эксперименты по тестированию устойчивости с использованием инструментов chaos engineering в тестовых и production-средах.	Задание 16	
ИД-4 $_{ПК-5.}$	Анализирует результаты экспериментов, сопоставляя фактические метрики с ожидаемыми, и выявляет скрытые уязвимости и слабые места архитектуры	Задание 17	
ИД-5 $_{ПК-5.}$	Разрабатывает и внедряет корректирующие меры по итогам экспериментов (настройка таймаутов,	Задание 18	анализирует их результаты, формулирует и внедряет корректирующие меры для повышения

	ретраев, circuit breaker, оптимизация архитектуры).		отказоустойчивости и устойчивости к сбоям
ИД-6 $_{ПК-5.}$	Проводит повторные эксперименты для подтверждения эффективности внедренных изменений и повышения отказоустойчивости	Задание 19	
ИД-7 $_{ПК-5.}$	Интегрирует практики регулярного тестирования устойчивости (chaos engineering) в процессы разработки и эксплуатации, включая автоматизацию базовых экспериментов в CI/CD.	Задание 20	
ИД-1 $_{ПК-6.}$	Анализирует хронику событий и технические метрики инцидента для восстановления полной и достоверной картины происшествия.	Задание 21	проводит глубокий анализ инцидентов по методологии blameless postmortem, выявляет системные коренные причины, разрабатывать и внедрять превентивные меры
ИД-2 $_{ПК-6.}$	Применяет методы системного анализа (5 почему, диаграмма Исикавы, анализ графов) для выявления коренных (системных) причин сбоев.	Задание 22	
ИД-3 $_{ПК-6.}$	Разрабатывает комплекс превентивных мер и технических решений, направленных на устранение выявленных коренных причин и предотвращение повторения инцидентов.	Задание 23	
ИД-4 $_{ПК-6.}$	Автоматизирует процессы обнаружения и реагирования на инциденты с использованием скриптов и средств оркестрации для сокращения ручного труда и времени простоя.	Задание 24	Автоматизирует процессы реагирования для устранения рутины и повышения устойчивости операций
ИД-5 $_{ПК-6.}$	Документирует результаты анализа инцидентов в формате blameless postmortem и формирует базу знаний для повышения устойчивости операций.	Задание 25	
ИД-1 $_{ПК-8.}$	Внедряет инструменты статического анализа безопасности (SAST) в процесс разработки для автоматического выявления уязвимостей в	Задание 26	Интегрирует практики безопасности на всех этапах жизненного цикла, включая статический и динамический анализ

	исходном коде на этапе pull request'ов.		кода, анализ уязвимостей зависимостей, безопасность контейнеров и облачной инфраструктуры, обеспечивая непрерывное соответствие требованиям безопасности
ИД-2 $_{ПК-8.}$	Организует автоматическое сканирование зависимостей и сторонних библиотек (SCA) для контроля уязвимостей в цепочке поставки ПО.	Задание 27	
ИД-3 $_{ПК-8.}$	Обеспечивает безопасность контейнерной среды: сканирует образы на уязвимости, использует минимальные базовые образы и внедряет механизмы подписывания образов.	Задание 28	
ИД-4 $_{ПК-8.}$	Проводит динамический анализ безопасности (DAST) запущенных приложений в тестовых и стейджинговых средах для выявления уязвимостей в рантайме.	Задание 29	
ИД-5 $_{ПК-8.}$	Сканирует конфигурации инфраструктуры как кода (IaC) на соответствие политикам безопасности и лучшим практикам до их применения.	Задание 30	
ИД-6 $_{ПК-8.}$	Интегрирует инструменты безопасности в CI/CD-пайплайны, настраивая автоматические шлюзы качества (quality gates), блокирующие поставку при обнаружении критических уязвимостей.	Задание 31	
ИД-7 $_{ПК-8.}$	Мониторит состояние безопасности контейнеров и инфраструктуры в рантайме, выявляя инциденты и аномальное поведение.	Задание 32	
ИД-8 $_{ПК-8.}$	Обеспечивает непрерывное соответствие (continuous compliance) систем требованиям стандартов безопасности и регуляторным нормам через автоматизированный сбор доказательств и генерацию отчетов.	Задание 33	
ИД-1 $_{ПК-11.}$	Моделирует сценарии	Задание 34	проводит анализ угроз

	атак на автономные системы, включая подмену сигнала (спуфинг), глушение (джамминг), перехват управления и внедрение вредоносного кода.		кибербезопасности для автономных систем, проектирует и внедряет меры защиты на уровнях канала связи, бортового ПО, наземной инфраструктуры управления и цепочек поставок, обеспечивая целостность, конфиденциальность и доступность системы
ИД-2 $_{ПК-11}$.	Внедряет механизмы шифрования и аутентификации в радиоканалах и сетях передачи данных для обеспечения конфиденциальности и подлинности команд управления.	Задание 35	
ИД-3 $_{ПК-11}$.	Применяет методы безопасной разработки (SAST, DAST) и контроля версий для защиты бортового ПО от модификации и эксплуатации уязвимостей.	Задание 36	
ИД-4 $_{ПК-11}$	. Реализует меры сетевой защиты (сегментация, межсетевое экранирование, системы обнаружения вторжений) для наземных центров управления полетами/движением.	Задание 37	
ИД-5 $_{ПК-11}$.	Разрабатывает процедуры проверки происхождения и целостности комплектующих на этапе производства и поставки для исключения компрометации на аппаратном уровне.	Задание 38	
ИД-6 $_{ПК-11}$.	Оценивает эффективность реализованных мер защиты через проведение пентестов и анализа защищенности автономной системы как единого комплекса.	Задание 39	

3. Методические материалы, определяющие процедуры оценивания и характеризующих этапы формирования компетенций

Процедура прохождения производственной эксплуатационной практики включает в себя следующие этапы: организация практики, подготовительный этап, исследовательский этап, этап обработки и анализа информации, подготовка отчета. На каждом этапе практики осуществляется текущий контроль за процессом формирования компетенций.

На каждом этапе практики осуществляется текущий контроль за процессом формирования компетенций.

Предлагаемые студенту задания позволяют проверить компетенции ПК-2, ПК-4, ПК-5, ПК-6, ПК-8, ПК-11.

При прохождении практики необходимо:

При организации практики:

– присутствовать на всех организационных собраниях и консультациях по практике;

– познакомиться с программой прохождения практики;

– получить документацию по практике (программу практики и дневник практики с направлением на практику) в сроки, определенные программой;

– получить индивидуальное задание у научного руководителя по практике и согласовать с ним календарный план работы на период практики.

В период подготовительного этапа практики:

– активно овладевать практическими навыками работы в сфере профессиональной деятельности;

– подготовка и проведение семинарских и лекционных занятий.

– разработка оценочных средств для реализации форм контроля в учебном процессе;

– проведение мероприятия в рамках воспитательной и научно-исследовательской работы студентов по соответствующим кафедрам.

На этапе подготовки отчета:

– оформить дневник по установленной форме и сдать на кафедру в срок не позднее 5 дней после окончания практики;

– подготовить отчет по практике в соответствии с требованиями программы практики и своевременно сдать руководителю практикой или на кафедру прикладной информатики;

– защитить в установленные сроки отчёт по практике.

При проверке задания, оцениваются документы обучающегося:

Оцениваются следующие компоненты дневника:

– разбивка по дням,

– понятность, полнота, подробность описаний,

– наглядность иллюстративных материалов,

– соблюдение правил оформления текста,

– обоснованность и целесообразность выполнения исследований,

– наличие необходимых сведений для установления контакта с представителями профильной организации.

При проверке отчетов оцениваются:

– календарно-тематический план прохождения практики;

– развернутые планы-конспекты практических (семинарских) или лекционных занятий;

– самоанализ одного из проведенных занятий.

По результатам прохождения практики, руководитель после проведенной заключительной конференции, представляет на кафедру прикладной информатики:

– проверенные и подписанные дневники обучающихся о прохождении научно-исследовательской практики;

– отчет о проведении практики (выполнение программы практики, сроки проведения, количество, места прохождения);

– характеристики обучающихся, подписанные руководителем;

– итоговые ведомости.

При защите отчета оцениваются:

– полнота, материал, полученный в процессе прохождения практики и выводы отчета;

– корректность и правильность заполнения;

– использование новых информационных и интеллектуальных технологий.

– наличие конкретных достижений обучающегося.