МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

МИНИСТЕРСТВО НАУКИ, ВЫСШЕГО ОБРАЗОВАНИЯ И ИННОВАЦИЙ КЫРГЫЗСКОЙ РЕСПУБЛИКИ

МОО ВОКыргызско-Российский Славянский университет имени первого Президента Российской Федерации Б. Н. Ельцина

Высшая школа цифровых технологий

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине

«Управление конфигурацией программного обеспечения (DevOps)»

Уровень высшего образования

МАГИСТРАТУРА

Направление подготовки

09.04.04 - РФ, 710400 - КР Программная инженерия

(код и наименование направления подготовки)

"Основы анализа и разработки приложений с большими объемами распределенных данных"

(профиля образовательной программы)

Квалификация

магистр

Фонд оценочных средств предназначен для текущего, рубежного и промежуточного контроля результатов обучения магистрантов по дисциплине «Управление конфигурацией программного обеспечения (DevOps)» направления 09.04.04 «Программная инженерия».

Фонд оценочных средств рассмотрен и утвержден
на заседании Высшей школы Цифровых технологий протокол №1 от 02.09.26

Директор Высшей школы Цифровых технологий:

к.т.н., доцент Хмелева И.В.

Руководитель образовательной программы:

д.т.н., профессор Лыченко Н.М.

Исполнители (разработчики):

ст. преподаватель Беляев А.А.

СОГЛАСОВАНО:
Зам. директора по учебной работе
Политехнического института КРСУ Комарцов Н.М.

Раздел 1. Перечень компетенций с указанием этапов их формирования

Шифр	Планируемые результаты обучения	Виды оценочных средств
ОПК-8 Способен осуществлять эффективное управление разработкой программных средств и проектов	Знать: модели жизненного цикла программного обеспечения; основные источники рисков в программном проекте.	Блок А -вопросы и контрольные работы № 1–3; блок D - теоретические вопросы зачёта.
	Уметь: применять методы и модели жизненного цикла ПО на практике; применять методы оценки рисков.	Блок В -практические задания № 1–8; блок D -практические кейсы.
	Владеть: инструментарием планирования деятельности в программном проекте; методическим инструментарием оценки рисков.	Блок С -комплексное задание и курсовая работа; блок D - практические задания на планирование и анализ рисков.

Раздел 2. Технологическая карта дисциплины

Название модуля	Контроль	Форма контроля	Мин., балл	Макс., балл	Неделя
Модуль 1. Основы DevOps, управления конфигурацией и контроля версий	Текущий	Сдача и защита практических работ № 1–3; устный опрос по материалам раздела	7	12	4
	Рубежный	Контрольная работа № 1: DevOps, Git, процессы контроля версий, планирование и риски	6	10	4
Модуль 2. Инфраструктура как код и автоматизация управления конфигурациями	Текущий	Сдача и защита практических работ № 4–5 (Ansible, инфраструктура как код)	7	12	10
	Рубежный	Контрольная работа № 2: конфигурационные единицы, Ansible, роли, идемпотентность, риски	6	10	10
Модуль 3. Непрерывная поставка ПО,	Текущий	Сдача и защита практических работ № 6–8 (CI/CD, мониторинг, Kubernetes)	8	14	16

мониторинг и Kubernetes	Рубежный	Контрольная работа № 3: CI/CD, метрики и Kubernetes, анализ рисков поставки	6	12	16
ИТОГО за семестр		Текущий и рубежный контроль	40	70	16
Промежуточная аттестация	Зачёт с оценкой	Теоретические вопросы (до 10 баллов) и практическое задание (до 20 баллов)	20	30	По расписанию
Семестровый рейтинг			60	100	

Технологическая карта курсовой работы

Курсовая работа является самостоятельной формой промежуточной аттестации, предусмотренной учебным планом. Темы и оценочные требования приведены в блоке С.

Модуль	Контроль	Форма контроля	Мин.	Макс.	Неделя
Курсовая работа	Текущий	Утверждение темы, плана, архитектурной схемы, состава конфигураций и реестра рисков	20	35	4–10
	Рубежный	Демонстрация функционирующего прототипа/проектной инфраструктуры; отчёт и воспроизводимость сборки	20	35	16
ИТОГО за семестр			40	70	16
Промежуточная аттестация	Защита КР	Доклад, демонстрация результата, ответы на вопросы	20	30	По расписанию
Семестровый рейтинг КР			60	100	

Раздел 3. Типовые контрольные задания и иные материалы

Блок А. Задания репродуктивного уровня

Назначение: проверка знаний терминологии, принципов и процессов управления конфигурацией программного обеспечения; понимания жизненного цикла ПО, планирования работ и источников рисков.

Модуль 1. Контрольная работа № 1 «Основы DevOps и системы контроля версий»

Типовой вариант рубежного контроля: 5 теоретических вопросов и одно ситуационное задание из соответствующего модуля. Состав билета определяется преподавателем.

1. Дайте определение DevOps. Какие проблемы разработки и эксплуатации помогает решать этот подход?
2. Назовите виды жизненного цикла ПО. Как выбор модели влияет на планирование поставки?
3. Что такое конфигурационная единица? Чем она отличается от версии исходного файла?
4. Какие виды деятельности включены в управление конфигурацией ПО?
5. Сравните централизованные и распределённые системы контроля версий (SVN и Git).
6. Опишите назначение ветки, фиксации (commit), слияния (merge) и тега в Git.
7. Для чего используются GitLab, Bitbucket и системы отслеживания задач?
8. Приведите пример конфликта слияния и опишите порядок его разрешения.
9. Назовите типичные риски совместной работы распределённой команды и способы их снижения.
10. Какие результаты должны содержаться в плане управления конфигурацией проекта?

Модуль 2. Контрольная работа № 2 «Инфраструктура как код и Ansible»

Типовой вариант рубежного контроля: 5 теоретических вопросов и одно ситуационное задание из соответствующего модуля. Состав билета определяется преподавателем.

1. Раскройте смысл подхода Infrastructure as Code (IaC).
2. Как управление инфраструктурой как кодом связано с принципами DevOps?
3. Назовите назначение инвентаря (inventory), playbook и модуля Ansible.
4. Что означает идемпотентность при применении конфигурации?
5. Опишите назначение ролей Ansible и пример их переиспользования.
6. Как организовать параметры для тестового и производственного окружений?
7. Какие данные недопустимо хранить открыто в репозитории инфраструктурного кода?
8. Как контролировать и документировать изменения конфигураций?
9. Какие риски возникают при выполнении конфигурационных изменений на нескольких узлах?
10. Какие проверки позволяют подтвердить корректность применения конфигурации?

Модуль 3. Контрольная работа № 3 «CI/CD, мониторинг и Kubernetes»

Типовой вариант рубежного контроля: 5 теоретических вопросов и одно ситуационное задание из соответствующего модуля. Состав билета определяется преподавателем.

1. Дайте определения непрерывной интеграции, непрерывной поставки и развёртывания.
2. Назовите типичные стадии конвейера CI/CD и условия его остановки.
3. Как связаны контроль версий, автоматизированное тестирование и управление релизами?
4. Какие виды мониторинга применяются к инфраструктуре, приложению и бизнес-логике?
5. Чем отличаются метрики, журналы событий и сигналы оповещения?
6. Что такое Kubernetes и какие задачи решает его оркестрация?
7. Назовите основные объекты Kubernetes: Pod, Deployment и Service.
8. В чём назначение readiness- и liveness-проверок?
9. Как действовать при неуспешном развёртывании новой версии приложения?
10. Предложите меры снижения риска недоступности сервиса после релиза.

Блок В. Задания реконструктивного уровня

Практические задания соответствуют занятиям в РПД (всего 26 часов). В каждом задании магистрант должен представить результаты в виде отчёта, ссылок на репозиторий/артефакты и краткой защиты.

Практическая работа № 1. Обзор инфраструктуры и проектный план (2 ч.)

1. Выберите пример веб-приложения с БД, опишите компоненты, роли участников и используемые окружения.
2. Составьте план внедрения управления конфигурацией: перечень конфигурационных единиц, этапы, ответственные и сроки.
3. Определите минимум пять рисков, оцените вероятность/влияние и запишите меры реагирования.

Представляемые материалы: Диаграмма инфраструктуры, план работ, реестр рисков.

Практическая работа № 2. Совместная работа над распределённым проектом (2 ч.)

1. Создайте общий репозиторий и согласуйте правила ветвления, сообщений фиксаций и проверки запросов на слияние.
2. Распределите задачи между участниками, выполните изменения в разных ветках и объедините результаты.
3. Зафиксируйте конфликт или риск параллельных изменений и опишите решение.

Представляемые материалы: История Git, запросы на слияние, таблица распределения задач и выводы.

Практическая работа № 3. Система контроля версий Git (4 ч.)

1. Создайте репозиторий, не менее трёх осмысленных фиксаций, отдельную ветку функции и тег релиза.
2. Продемонстрируйте слияние изменений и процедуру отката ошибочной фиксации на тестовом примере.
3. Подготовьте инструкцию восстановления выбранной версии и краткий анализ рисков ошибок слияния.

Представляемые материалы: Репозиторий Git, набор команд, журнал версий, отчёт.

Практическая работа № 4. Автоматизация конфигураций с Ansible (4 ч.)

1. Подготовьте inventory двух логических или виртуальных узлов и playbook установки и настройки выбранного сервиса.
2. Выполните playbook повторно и проверьте его идемпотентность.
3. Оформите тест выполнения, условия отката и план действий при сбое.

Представляемые материалы: Файлы inventory и playbook, вывод выполнения, проверки конфигурации.

Практическая работа № 5. Организация инфраструктурного кода (4 ч.)

1. Разделите исходный playbook на минимум две роли Ansible, определите переменные окружений.
2. Выделите конфигурационные параметры, которые нужно хранить отдельно от исходного кода.

3. Проанализируйте поддержку версий, повторное использование и риски открытого хранения секретов.

Представляемые материалы: Структура ролей, пример конфигурационных файлов, README и отчёт.

Практическая работа № 6. Конвейер CI/CD (4 ч.)

1. Создайте конфигурацию конвейера с этапами проверки/сборки/тестирования для учебного проекта.
2. Добавьте правило, не допускающее поставку версии при провале тестов.
3. Представьте план выпуска и отката, сводку рисков и лог успешного/неуспешного выполнения.

Представляемые материалы: Файл конвейера, результаты запусков, описание релизной процедуры.

Практическая работа № 7. Мониторинг приложения (4 ч.)

1. Определите метрики доступности и качества работы учебного сервиса, укажите их источники.
2. Подготовьте пример сбора и визуализации метрик с порогами предупреждения.
3. Смоделируйте отклонение показателей и оформите порядок реагирования команды.

Представляемые материалы: План мониторинга, снимки панели/таблиц, сценарий инцидента.

Практическая работа № 8. Введение в Kubernetes (2 ч.)

1. Подготовьте манифесты Deployment и Service для контейнеризированного учебного приложения.
2. Покажите способ масштабирования и проверки состояния экземпляров приложения (в тестовой среде или на моделируемом примере).
3. Назовите риски недоступности при обновлении и меры их снижения.

Представляемые материалы: Манифесты YAML, команды проверки, краткий отчёт.

Блок С. Задания практико-ориентированного и исследовательского уровня

Комплексное ситуационное задание

В команде сопровождается программный сервис, размещённый на нескольких узлах. Перед релизом обнаружено расхождение конфигураций тестовой и рабочей сред, а в конвейере CI/CD периодически падают тесты.

1. Составить план анализа ситуации, перечень конфигурационных единиц и участников работ.
2. Выбрать порядок устранения дрейфа конфигураций и предложить автоматизацию средствами IaC.
3. Построить схему поставки с контрольными точками качества и возможностью отката.
4. Сформировать реестр рисков по вероятности, влиянию, владельцу и ответным действиям.
5. Назвать критерии успешности изменений и способы подтверждения стабильности после развёртывания.

Курсовая работа

Курсовая работа выполняется индивидуально. Конкретную тему и особенности среды утверждает преподаватель. Ниже предложены типовые темы, разработанные в соответствии с содержанием РПД, поскольку источник не содержит утверждённого перечня тем курсовых работ.

1. Проектирование Git-процесса сопровождения распределённого программного проекта.

2. Разработка воспроизводимой конфигурации тестового окружения средствами Ansible.

3. Проектирование процесса управления изменениями конфигураций программной системы.

4. Разработка прототипа CI/CD-конвейера сборки, тестирования и выпуска приложения.

5. Автоматизация развёртывания типового веб-сервиса с разделением окружений.

6. Разработка регламента мониторинга программного сервиса и реагирования на инциденты.

7. Подготовка Kubernetes-манифестов для масштабируемого учебного приложения.

8. Проектирование единого процесса управления версиями, конфигурациями и релизами.

Обязательные результаты: пояснительная записка, схема архитектуры и конфигурационных единиц; план работ и реестр рисков; функционирующий прототип или воспроизводимый учебный сценарий; исходные файлы в репозитории; инструкция по запуску, результаты испытаний и выводы.

Пример задания: «Автоматизация настройки тестового окружения веб-приложения средствами Ansible». Подготовить inventory и роли, добавить проверку доступности, воспроизвести развёртывание два раза, проверить идемпотентность, представить инструкцию запуска и реестр рисков.

Блок D. Промежуточный контроль -зачёт с оценкой

В рамках зачёта проверяются теоретические знания, умения применить инструменты DevOps и навыки планирования программного проекта и анализа рисков. Ниже приведён перечень вопросов и заданий для подготовки.

Вопросы для проверки уровня «ЗНАТЬ»

1. Понятие и цели управления конфигурацией ПО.

2. Принципы DevOps и связь с жизненным циклом ПО.

3. Состав конфигурационных единиц и документирование изменений.

4. Модели жизненного цикла ПО и их влияние на управление версиями.

5. Роль Git в управлении исходным кодом; ветвление и слияние.

6. Системы управления разработкой GitLab, Bitbucket, Redmine.

7. Планирование работ, оценка трудозатрат и источники рисков программного проекта.

8. Понятие инфраструктуры как кода, её преимущества и ограничения.

9. Архитектура и основные сущности Ansible.

10. Организация ролей и окружений Ansible.

11. Идемпотентность и контроль конфигурационного дрейфа.

12. Принципы непрерывной интеграции и непрерывной поставки.

13. Этапы CI/CD-конвейера и автоматизированные проверки качества.

14. Управление версиями, релизами и откатами.

15. Метрики, журналы и оповещения в системах мониторинга.

16. Мониторинг инфраструктуры, приложения и бизнес-логики.

17. Базовая архитектура Kubernetes.

18. Назначение Pod, Deployment, Service.

19. Методы установки и проверки состояния кластера Kubernetes.

20. Управление рисками разработки и эксплуатации ПО.

Задания для проверки уровней «УМЕТЬ» и «ВЛАДЕТЬ»

1. Разработать схему ветвления Git для проекта из трёх участников с описанием процедуры выпуска релиза.

2. Составить перечень конфигурационных единиц и план управления изменениями учебной системы.

3. Подготовить пример структуры playbook и inventory Ansible для двух окружений.

4. Спроектировать CI/CD-конвейер с обязательным тестированием и правилами отката.

5. Определить набор метрик учебного сервиса и предложить пороги уведомлений.

6. Составить Deployment и Service для типового приложения Kubernetes на уровне основных полей манифеста.

7. Составить реестр рисков миграции приложения на новую конфигурацию, определить меры реагирования.

Пример билета зачёта с оценкой № 1

1. Объясните назначение конфигурационных единиц и процессов управления изменениями в DevOps.

2. Сравните непрерывную интеграцию и непрерывную поставку ПО.

3. Перечислите ключевые объекты Kubernetes и поясните назначение Deployment.

4. Практическое задание. Для учебного сервиса с двумя окружениями предложите схему Git-ветвления, этапы CI/CD-конвейера и реестр не менее трёх рисков релиза с мерами снижения.

Раздел 4. Методические материалы и процедуры оценивания

4.1. Процедуры контроля

- Текущий контроль: проверка и защита отчётов по практическим работам, устный опрос. Преподаватель оценивает качество решений, объяснение действий, воспроизводимость результатов и выводы.
- Рубежный контроль: одна контрольная работа в конце каждого из трёх модулей на 4-й, 10-й и 16-й учебных неделях.
- Промежуточная аттестация: зачёт с оценкой по билету, включающему теоретические вопросы и практическое задание.
- Курсовая работа: отдельные этапы проверки и защита результатов в соответствии с технологической картой курсовой работы.

4.2. Шкала оценивания практических работ (текущий контроль)

Доля от максимума	Критерий
85–100 %	Полное и самостоятельное выполнение задания; результат воспроизводим; обоснованные выводы и корректные ответы.
70–84 %	Основные требования выполнены; имеются небольшие недочёты оформления или объяснения.
60–69 %	Достигнут частичный результат; часть требований реализована с замечаниями.
31–59 %	Выполнены отдельные элементы, существенные ошибки и пробелы в результате.
0–30 %	Задание не выполнено либо отсутствуют подтверждения результата.

4.3. Шкала оценивания контрольных работ (рубежный контроль)

Доля от максимума	Критерий
85–100 %	Все базовые положения изложены верно; практическая ситуация разобрана, решения обоснованы.
70–84 %	Основные вопросы раскрыты; есть некритичные ошибки в деталях.
60–69 %	Раскрыты основные определения, но анализ и обоснование недостаточно полны.
31–59 %	Большинство вопросов раскрыты частично; существенные ошибки.
0–30 %	Ответ отсутствует или обнаружено непонимание фундаментальных положений.

4.4. Зачёт с оценкой

Теоретическая часть оценивается от 0 до 10 баллов: 9–10 -полные аргументированные ответы; 7–8 -верные ответы с небольшими неточностями; 5–6 -частично раскрытые вопросы; 1–4 -существенные пробелы; 0 -нет ответа.

Практическое задание оценивается от 0 до 20 баллов: 18–20 -корректное и обоснованное решение с выявленными рисками; 14–17 -работоспособное решение с небольшими недочётами; 10–13 -частичный результат; 1–9 -фрагментарное или неверное решение; 0 -нет решения.

4.5. Шкала перевода семестрового рейтинга в оценку

Итоговые баллы	Оценка
0–60	Неудовлетворительно / не зачтено

61–75	Удовлетворительно / зачтено
76–89	Хорошо / зачтено
90–100	Отлично / зачтено

Шкала соответствует 100-балльной системе оценивания. Итоговая оценка по дисциплине определяется суммой баллов за три модуля (до 70) и зачёт с оценкой (до 30), всего не более 100.

4.6. Оценивание курсовой работы

Показатель	Максимум, баллы	Порядок проверки
План проекта, состав конфигураций, архитектурная схема	20	Текущий контроль
Анализ трудозатрат и рисков, обоснование решений	15	Текущий контроль
Качество реализации и воспроизводимость прототипа	20	Рубежный контроль
Документация, испытания и выводы	15	Рубежный контроль
Защита: доклад, демонстрация и ответы на вопросы	30	Промежуточная аттестация
ИТОГО	100	

Минимальные пороги по этапам приведены в технологической карте курсовой работы. Работа оценивается на основании её содержания, самостоятельности выполнения и аргументированности ответов. Шкала перевода баллов в оценку -та же, что в пункте 4.5.

4.7. Особые случаи и объективность оценивания

- Отсутствие файла отчёта или репозитория не компенсируется устным пересказом; по решению преподавателя студент предоставляет эквивалентное воспроизводимое подтверждение выполнения.
- При командном выполнении заданий каждый обучающийся демонстрирует собственный вклад и знания по выполненной работе.
- Ошибки эксплуатации учебного стенда должны отделяться от ошибок решения: если среда недоступна, допускается проверка конфигурационных файлов, сценария выполнения и тестовых логов.
- Использование сторонних фрагментов программ и конфигураций сопровождается ссылкой на источник; самостоятельность проектных решений проверяется на защите.

Раздел 5. Методические указания обучающемуся

5.1. Общие рекомендации

Для успешного освоения дисциплины необходимо посещать лекционные и практические занятия, вести краткий конспект, систематически выполнять задания и изучать рекомендованную литературу. При подготовке к контрольным работам повторите определения

и основные процессы управления конфигурацией, жизненный цикл ПО, Git, Ansible, CI/CD, мониторинг и Kubernetes.

5.2. Выполнение и оформление практических работ

1. Перед началом практической работы изучите постановку задачи, требования и ожидаемые результаты.
2. Используйте учебный репозиторий с понятной структурой каталогов; не размещайте пароли, токены и закрытые ключи в открытом виде.
3. В отчёте укажите цель, исходные данные, последовательность действий, команды или файлы конфигурации, результаты проверок, выявленные риски и выводы.
4. Приложите ссылку на репозиторий, фрагменты логов и снимки результатов, необходимые для воспроизведения проверки.
5. На защите объясните выбор инструментов и покажите ключевые результаты; будьте готовы обосновать план изменений и меры управления рисками.

5.3. Подготовка к рубежному контролю

В каждом модуле изучите перечень вопросов блока A и повторите команды и концепции из практических заданий блока B. На контрольной работе используйте точные определения, схему решения и аргументы; при анализе ситуаций указывайте предпосылки, ожидаемые последствия и риски.

5.4. Порядок выполнения курсовой работы

1. Согласовать тему, цель, исходные ограничения и критерии успеха с преподавателем.
2. Подготовить календарный план и схему конфигурационных единиц; составить реестр рисков.
3. Разработать прототип либо воспроизводимый проект автоматизации и провести проверки.
4. Оформить пояснительную записку: постановка задачи, анализ вариантов, схема реализации, результаты тестирования, оценка рисков, выводы и библиография.
5. Представить исходные материалы и провести демонстрацию; подготовить доклад для защиты и ответы на вопросы.

5.5. Подготовка к зачёту с оценкой

Изучите вопросы блока D и выполните тренировочные практические задания: составление схем Git-процесса, конфигурационных сценариев, CI/CD-конвейеров, перечня метрик и реестра рисков. Во время зачёта необходимо показать не только знание терминов, но и умение применять методы управления разработкой ПО к конкретным случаям..