

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Порохня Андрей Александрович

Должность: И.о. директора института перспективной инженерии

Дата подписания: 19.06.2026 16:00:03

Уникальный программный ключ:

990bea3aeec548c23bd8699e48288f8d1960c600

**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
А.А. Порохня

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

Проектная деятельность

Направление подготовки	09.03.04 Программная инженерия
Направленность (профиль)	Разработка и сопровождение программного обеспечения
Год начала обучения	2026
Форма обучения	очная
Реализуется семестре	1, 2

Введение

1. Назначение: Оценочные материалы (оценочные средства) предназначены для проведения текущего контроля успеваемости и промежуточной аттестации обучающихся, и представляют собой совокупность контрольно-измерительных материалов и методов их использования для измерения уровня достижения обучающимися установленных результатов обучения.

2. ФОС является приложением к программе дисциплины «Проектная деятельность»

3. Разработчик Потапов И.Р., ассистент межинститутской базовой кафедры департамента цифровых, робототехнических систем и электроники

4. Проведена экспертиза ФОС.

Члены экспертной группы:

Председатель: Новикова Е. Н., заведующий межинститутской базовой кафедрой

(ФИО, должность)

Члены комиссии:

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

(ФИО, должность)

Винокурский Д. Л., доцент департамента цифровых, робототехнических систем и электроники

(ФИО, должность)

Представитель организации-работодателя: А. В. Новиков, начальник отдела АСУТП ООО «ЕНДС-Ставрополь»

(ФИО, должность)

Экспертное заключение: Оценочные материалы (оценочные средства) по дисциплине «Проектная деятельность» составлены в соответствии с требованиями самостоятельно устанавливаемого Федерального государственного образовательного стандарта высшего образования по направлению 09.03.04 «Программная инженерия» и позволяют оценить уровень сформированности компетенции УК-2, УК-3, ПК-7.

5. Срок действия ФОС определяется сроком реализации образовательной программы.

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

Уровни сформированности компетенци(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
Компетенция: УК-2. Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений				
Результаты обучения по дисциплине (модулю): <i>Индикатор:</i> УК-2ид-1 Формулирует цель проекта, определяет совокупность взаимосвязанных задач, обеспечивающих ее достижение и определяет ожидаемые результаты решения задач	Не умеет формулировать цель проекта, не выделяет задачи; ТЗ отсутствует или не связано с предметной областью	Формулирует общую цель (например, «разработать веб-приложение») и 2–3 размытые задачи, ТЗ содержит только перечень экранов без требований	Формулирует измеримую цель, декомпозирует на 5–7 задач, в ТЗ выделяет функциональные (что делает система) и 1–2 нефункциональных требования (производительность, безопасность)	Полное ТЗ: цель с критериями успеха, декомпозиция на задачи с привязкой к модулям, функциональные требования в формате «пользователь может...», нефункциональные (до 5 штук), приоритеты требований (MoSCoW)
Результаты обучения по дисциплине (модулю): <i>Индикатор:</i> УК-2ид-2 Разрабатывает план действий для решения задач проекта, выбирая оптимальный способ их решения, исходя из действующих правовых норм и имеющихся ресурсов и ограничений	Не может обосновать выбор инструментов; использует пиратское ПО без анализа	Перечисляет выбранные инструменты (Git, Trello, VS Code) без сравнения, но ссылается на бесплатность для образования	Сравнивает 2–3 альтернативы (Trello vs Yougile, Mermaid vs Draw.io), обосновывает выбор с учётом размера команды и сроков, проверяет лицензии (MIT, GPL, проприетарная студенческая)	Проводит мини-исследование: таблица сравнения инструментов по критериям (цена, функционал, порог входа), выбирает оптимальные, подкрепляя ссылками на лицензии, учитывает правовые нормы (ФЗ-149)

Результаты обучения по дисциплине (модулю): <i>Индикатор:</i> УК-2ид-3 Обеспечивает выполнение проекта в соответствии с установленными целями, сроками и затратами, исходя из действующих правовых норм, имеющихся ресурсов и ограничений, в том числе с использованием цифровых инструментов	Не использует инструменты управления проектом; не может оценить сроки	Создаёт простую Kanban-доску (To Do / In Progress / Done) в Trello/Yougile, ведёт историю коммитов в Git, но не оценивает реализуемость	Ведёт доску с приоритетами, дедлайнами, назначает ответственных; использует Wiki для документации; оценивает реализуемость по времени (диаграмма Ганта) и ресурсам (кто что делает)	Полностью управляет проектом: доска с бэклогом, спринтами, бёрндаун-чартом; Git с pull request-ами и код-ревью; Wiki структурирована; регулярно пересматривает реализуемость при изменении ограничений (риск-менеджмент)
Компетенция: УК-3. Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде				
Результаты обучения по дисциплине (модулю): <i>Индикатор:</i> УК-3ид-1 Участвует в межличностном и групповом взаимодействии, используя инклюзивный подход, эффективную коммуникацию, методы командообразования и командного взаимодействия	Роли не распределены, работа хаотична; инструментами синхронизации не пользуются	Роли назначены формально (каждый делает всё), но Git используется только для сдачи финального кода, синхронизация через чат	Чёткое распределение ролей с описанием зон ответственности, регулярные синхронизации через Git (merge, pull), общая документация в Markdown, проводятся короткие стендапы	Роли распределены с учётом навыков (кто-то силён в аналитике, кто-то в коде), используется Git Flow (ветки, ревью), Markdown + Mermaid для совместной документации, команда ретроспективно корректирует роли
Результаты обучения по дисциплине (модулю): <i>Индикатор</i> УК-3ид-2 Обеспечивает работу команды для получения оптимальных результатов совместной	Не участвует в обсуждениях или игнорирует мнение других	Присутствует на обсуждениях, но не аргументирует свои предложения, соглашается с большинством без анализа	Предлагает свои варианты архитектуры, использует UML-диаграммы для иллюстрации, аргументирует (производительность, сложность), принимает чужое	Весь процесс командного принятия решений: сравнительный анализ 2–3 вариантов архитектуры с взвешиванием плюсов и минусов, фиксация ADR

работы, с учетом индивидуальных возможностей её членов			мнение при весомых аргументах	(Architectural Decision Record), учёт всех мнений, достижение консенсуса
Компетенция: ПК-7. Способен применять современные информационные технологии, инструменты и платформы (включая отечественные), использовать DevOps-практики, облачные сервисы и средства автоматизации.				
Результаты обучения по дисциплине (модулю): <i>Индикатор:</i> ПК-7ид-1 Использует современные IDE, линтеры, формatters и инструменты статического анализа (SonarQube) для обеспечения качества кода	Не использует Markdown и Mermaid; документация в виде текстового файла без разметки	Пишет простой README.md (заголовки, списки), одну диаграмму последовательности в Mermaid с ошибками	Оформляет полную документацию на Markdown (структура, ссылки, таблицы), использует Mermaid для диаграммы Use Case и диаграммы классов (3–5 сущностей)	Профессиональная документация: несколько связанных md-файлов с оглавлением, UML-диаграммы в Mermaid (прецеденты, классы, последовательность, состояние), документация интегрирована в репозиторий, версионизируется
Результаты обучения по дисциплине (модулю): <i>Индикатор</i> ПК-7ид-2 Настраивает мониторинг, логирование и алертинг (Prometheus, Grafana, ELK/Loki, Sentry) для отслеживания состояния ПО	Не использует Git для документации; CI/CD не настроен	Коммитит документацию в тот же репозиторий, что и код, но без отдельной истории; CI/CD не использует	Ведёт документацию в отдельной ветке или папке, настраивает автоматическую сборку Markdown в HTML/PDF через GitHub Actions (простой workflow)	Полный пайплайн: при коммите в main документация собирается и публикуется на GitHub Pages (или аналоги), проверяется валидность ссылок, генерируется PDF-версия, настраивается линтинг Markdown
Результаты обучения по дисциплине (модулю): <i>Индикатор:</i> ПК-7ид-3 Применяет GitOps-подходы (ArgoCD), Helm и операторы Kubernetes для автоматизации	Не знает отечественных платформ	Может назвать GitFlic или аналоги (например, GosUslugi), но не работал с ними	Создаёт репозиторий на GitFlic, приглашает членов команды, ведёт там документацию и код (простое использование)	Использует GitFlic наравне с GitHub: настраивает CI/CD через встроенные средства (если есть), управляет доступом, знает различия в лицензировании и

развертывания в кластере				правовых аспектах по ФЗ-188
Результаты обучения по дисциплине (модулю): <i>Индикатор:</i> ПК-7 _{ид-4} Демонстрирует знание отечественных платформ и инструментов	Не пользуется облачными сервисами	Публикует готовую Wiki в репозитории (GitHub Wiki), не использует внешний хостинг	Публикует сгенерированную документацию (HTML) на бесплатном хостинге (GitHub Pages, Netlify, Cloudflare Pages)	Настраивает автоматический деплой документации на облачный хостинг из CI/CD, использует облачные редакторы (например, GitPod) для совместной правки, обеспечивает доступ для заказчика

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕРКИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Номер задания	Правильный ответ	Содержание вопроса	Компетенция
1.	1,2,3	Какие из перечисленных признаков характеризуют ПРОЕКТ (в отличие от процесса)? (1) уникальность результата (2) ограниченность во времени (3) наличие цели (4) повторяемость операций	УК-2
2.	2	Какой тип лицензии ПО требует, чтобы все производные работы распространялись под той же лицензией (обладает «вирусным» эффектом)? (1) MIT (2) GPL (3) Apache 2.0 (4) BSD	УК-2
3.	3	Что обозначает категория «Must have» в методе приоритизации MoSCoW? (1) функцию, которую хорошо бы иметь, но не критично (2) функцию, которую сознательно откладывают на следующий релиз (3) функцию, без которой продукт теряет смысл (4) функцию, которая важна, но может быть реализована позже	УК-2
4.	2	Какой элемент UML-диаграммы вариантов использования обозначается человечком? (1) прецедент (2) актёр (3) границы системы (4) отношение include	ПК-7
5.	1,3	Какие из перечисленных документов относятся к нефункциональным требованиям в ТЗ? (1) требования к надёжности (2) функция «вход в систему» (3) требования к эргономике (4) сценарий «зарегистрироваться»	УК-2

6.	2	Какой метод сбора требований позволяет получить наиболее глубокую информацию, но требует больших временных затрат? (1) анкетирование (2) интервью (3) анализ документов (4) наблюдение	УК-2
7.	3	Какой символ в Markdown используется для создания заголовка первого уровня? (1) * (2) - (3) # (4) >	ПК-7
8.	1,2,4	Какие инструменты относятся к системам контроля версий? (1) Git (2) GitHub (3) Yougile (4) GitLab	УК-3
9.	4	Какая команда Git используется для сохранения изменений в локальном репозитории? (1) git push (2) git add (3) git status (4) git commit	УК-3
10.	3	Что обозначает отношение <<extend>> на диаграмме use case? (1) один прецедент всегда вызывает другой (2) актёр наследует свойства другого актёра (3) один прецедент расширяет другой при определённом условии (4) прецедент находится внутри границ системы	ПК-7
11.	2,4	Какие из перечисленных требований являются функциональными? (1) время отклика системы не более 2 секунд (2) пользователь должен иметь возможность сбросить пароль по email (3) система должна работать на ОС Windows 10 (4) система должна отправлять уведомление после регистрации	УК-2

12.	1	Какой раздел технического задания (по ГОСТ 34.602-89) содержит формулировку целей и критериев эффективности системы? (1) Назначение разработки (2) Основание для разработки (3) Требования к надёжности (4) Состав и содержание работ	
13.	3	Какая лицензия является наиболее «мягкой» (пермиссивной) и разрешает использование кода в закрытых коммерческих продуктах с минимальными ограничениями? (1) GPL v3 (2) Apache 2.0 (3) MIT (4) LGPL	УК-2
14.	2	Что такое WBS (Work Breakdown Structure)? (1) инструмент для визуализации задач в виде доски (2) иерархическая декомпозиция работ проекта на управляемые компоненты (3) метод оценки трудоёмкости через анонимное голосование (4) документ, фиксирующий роли в команде	УК-2
15.	1,3	Какие элементы обязательно должны быть на диаграмме классов UML? (1) имя класса (2) методы с реализацией (3) атрибуты (4) значения атрибутов по умолчанию	ПК-7
16.	4	Что обозначает множественность 1..* на конце ассоциации между классами? (1) ровно один экземпляр (2) ноль или один (3) ноль или более (4) один или более	ПК-7
17.	2	Какой инструмент предназначен для визуализации задач и ограничения количества незавершённой работы (WIP)? (1) WBS (2) Kanban-доска (3) Use Case Diagram	УК-3

		(4) IDEF0-диаграмма	
18.	3	Какой код состояния HTTP означает «Успешное создание ресурса» (например, после POST-запроса)? (1) 200 (2) 400 (3) 201 (4) 404	ПК-7
19.	1,2	Какие из перечисленных элементов относятся к контекстной диаграмме IDEF0? (1) входы (2) управления (3) хранилища данных (4) потоки управления	УК-2
20.	4	Что такое MVP (Minimum Viable Product)? (1) документ с требованиями к продукту (2) финальная версия продукта (3) продукт со всеми запланированными функциями (4) минимальный набор функций, позволяющий получить обратную связь от пользователей	УК-2
21.	2	Какая роль в IT-команде отвечает за сбор и формализацию требований? (1) разработчик (2) аналитик (3) тестировщик (4) менеджер проекта	УК-3
22.	3	Что означает термин «Lead Time» в Kanban? (1) время выполнения одной задачи (2) количество задач в колонке «В работе» (3) время от появления задачи в бэклоге до её завершения (4) время от начала работы над задачей до её завершения	УК-2
23.	1,4	Какие из перечисленных платформ являются отечественными (российскими) аналогами GitHub? (1) GitFlic (2) SourceForge (3) Bitbucket	ПК-7

		(4) GitVerse	
24.	2	Что такое прототипирование (wireframes)? (1) написание рабочего кода (2) создание схематичных макетов экранов для визуализации интерфейса (3) тестирование готового продукта на пользователях (4) написание технического задания	ПК-7
25.	4	Какой размер компенсации за нарушение авторских прав на ПО предусмотрен ГК РФ (минимальный порог)? (1) 1 000 рублей (2) 5 000 рублей (3) 1 000 000 рублей (4) 10 000 рублей	УК-2
26.	2,3	Какие методы оценки трудоёмкости используются в IT-проектах? (1) метод Монте-Карло (2) экспертная оценка (3) Planning Poker (4) SWOT-анализ	УК-2
27.	3	Какой инструмент используется для создания диаграмм из текста (например, use case, классов) в Markdown-документации? (1) Draw.io (2) Balsamiq (3) Mermaid (4) Figma	ПК-7
28.	1	Что такое Issue в GitHub/GitLab? (1) единица работы для отслеживания задач, багов и фич (2) инструмент для документирования проекта (3) раздел с вопросами и ответами (4) файл с описанием лицензии	УК-3
29.	2	Какой процент отчислений с заработной платы во внебюджетные фонды условно принимается при расчёте себестоимости разработки ПО в РФ? (1) 13% (2) 30,2% (3) 50%	УК-2

		(4) 6%	
30.	1,4	Какие из перечисленных диаграмм относятся к UML? (1) диаграмма вариантов использования (2) IDEF0 (3) BPMN (4) диаграмма классов	ПК-7
31.	3	Что означает принцип «Золотое сечение» в питч-презентации? (1) каждый слайд должен содержать не более 10 строк текста (2) презентация должна длиться ровно 20 минут (3) 10% времени — проблема, 80% — решение, 10% — призыв к действию (4) количество слайдов должно быть нечётным	УК-3
32.	2	Какая команда Git используется для отправки изменений в удалённый репозиторий? (1) git pull (2) git push (3) git fetch (4) git clone	УК-3
33.	4	Что такое CI/CD в контексте DevOps? (1) система контроля версий (2) метод управления проектами (3) язык разметки документации (4) практика непрерывной интеграции и непрерывной доставки	ПК-7
34.	2,3	Какие элементы включает себестоимость разработки ПО? (1) цена продажи продукта (2) затраты на оплату труда (3) накладные расходы (4) прибыль компании	УК-2
35.	1	Какой вид ответственности наступает за незаконное использование ПО в крупном размере (свыше 100 000 рублей)? (1) уголовная (2) административная (3) гражданско-правовая (4) дисциплинарная	УК-2

36.	3	Как обозначается атрибут с видимостью private в UML? (1) + (2) # (3) - (4) ~	ПК-7
37.	1,4	Какие из перечисленных инструментов подходят для создания low-fidelity wireframes? (1) бумага и маркер (2) Adobe Photoshop (3) Visual Studio Code (4) Draw.io	ПК-7
38.	2	Что означает понятие «точка безубыточности» в экономике проекта? (1) объём продаж, при котором прибыль максимальна (2) объём продаж, при котором доходы равны расходам (3) минимальная цена продукта (4) максимальные затраты на разработку	УК-2
39.	1,3,4	Какие элементы обязательно должны быть в описании REST API метода? (1) метод HTTP (GET, POST и т.д.) (2) стоимость вызова (3) URL эндпоинта (4) формат тела запроса и ответа (JSON)	ПК-7
40.	2	Какой тип связи между классами UML обозначается закрашенным ромбом? (1) агрегация (2) композиция (3) ассоциация (4) обобщение	ПК-7

2. Описание шкалы оценивания

Оценка успеваемости студентов по дисциплине осуществляется в ходе текущего контроля и промежуточной аттестации. Для проверки уровня сформированности компетенций используется совокупность оценочных средств, позволяющих оценить знания, умения и навыки, приобретённые студентами в процессе изучения дисциплины.

Основными формами текущего контроля являются:

- выполнение и защита практических работ;
- устные опросы (собеседования) по темам занятий;
- проверка индивидуальных заданий;
- тестирование по отдельным разделам дисциплины.

Промежуточная аттестация проводится в форме зачета и включает выполнение практического задания и/или ответы на теоретические вопросы.

3. Критерии оценивания компетенций заданий открытого типа

Оценка **«отлично»** – высокий уровень сформированности компетенции **УК-2** **выставляется студенту, если:**

Цель проекта сформулирована грамотно, конкретно, измеримо и достижима в рамках учебного проекта.

Выполнена полная и логичная декомпозиция цели на совокупность функциональных и нефункциональных задач, представленная в техническом задании (ТЗ) с использованием метода MoSCoW.

Обоснован выбор оптимальных инструментов (языки программирования, платформы, библиотеки) с учётом всех имеющихся ресурсов (время, бюджет, стек технологий, уровень команды).

При планировании архитектуры и выборе сторонних компонентов учтены правовые нормы: проверены лицензии всех зависимостей, для собственного проекта выбрана корректная лицензия (MIT, GPL, Apache и др.).

Выполнена количественная оценка реализуемости задач (WBS, человеко-часы), проведён анализ рисков, предложены меры их снижения.

Представленный документ отчёта не содержит ошибок, приведены все ключевые определения и аргументированные выводы по каждому разделу.

Оценка **«хорошо»** – средний уровень сформированности компетенции **УК-2** **выставляется студенту, если:**

Цель проекта сформулирована чётко, но недостаточно измерима или имеет незначительные неточности.

Выполнена декомпозиция на функциональные и нефункциональные задачи (не менее 70% задач), ТЗ оформлено с незначительными замечаниями.

Выбор инструментов обоснован с учётом основных ограничений, приведено сравнение с 1–2 альтернативами, но аргументация неполная.

Правовые нормы учтены для всех используемых компонентов, но обоснование выбора лицензии для собственного проекта отсутствует или поверхностно.

Выполнена качественная оценка реализуемости (WBS), но отсутствует анализ рисков.

Представленный документ отчёта содержит незначительные ошибки, не представлены некоторые определения и выводы, не влияющие на общий результат.

Оценка **«удовлетворительно»** – минимальный уровень сформированности компетенции **УК-2**

выставляется студенту, если:

Цель проекта сформулирована, но недостаточно конкретна, содержит общие формулировки («сделать хороший продукт»).

Декомпозиция выполнена частично (менее 50% задач), ТЗ содержит грубые ошибки в структуре или отсутствуют важные разделы.

Выбор инструментов обоснован поверхностно (1–2 аргумента), ресурсы учтены частично, сравнение с альтернативами отсутствует.

Правовые нормы упоминаются, но применяются непоследовательно: лицензии проверены не для всех компонентов, для собственного проекта лицензия не выбрана.

Оценка реализуемости выполнена на качественном уровне («выполнимо / невыполнимо»), без количественных показателей и привязки к срокам.

Представленный документ отчёта содержит ошибки, не позволяющие оценить качество разработки, ключевые определения и выводы представлены не в полном объёме.

Оценка **«неудовлетворительно»** – компетенция **УК-2** не сформирована

выставляется студенту, если:

Цель проекта не сформулирована или сформулирована некорректно (недостижима, не измерима, не соответствует предметной области).

Декомпозиция отсутствует, либо задачи не соответствуют цели. ТЗ не соответствует требованиям или не представлено.

Выбор инструментов не обоснован или отсутствует, предложены заведомо неподходящие технологии, ресурсы не учтены.

Правовые нормы не учитываются, использование сторонних компонентов без проверки лицензий, отсутствует понимание авторских прав.

Оценка реализуемости отсутствует, задачи заведомо невыполнимы в рамках учебного проекта.

Представленный документ отчёта содержит много грубых ошибок, ключевые определения и выводы отсутствуют, структура документа нарушена.

Оценка **«отлично»** – высокий уровень сформированности компетенции **УК-3**

выставляется студенту, если:

Роли в команде распределены оптимально на каждом этапе проекта с учётом индивидуальных сильных сторон и предпочтений участников; зоны ответственности чётко определены.

Инструменты коллективной работы (Git с ветками и Pull Requests, Kanban-доска, совместное редактирование документации в Markdown) применяются на профессиональном уровне; синхронизация действий команды полная и регулярная.

Студент активно участвует в обсуждении архитектурных решений и UML-моделей, приводит развёрнутую аргументацию с опорой на стандарты и практический опыт, интегрирует лучшие идеи команды в итоговое решение.

Обязательства в рамках командного проекта выполняются в срок или досрочно, соблюдаются все общие стандарты оформления кода и документации; студент помогает другим членам команды соблюдать договорённости.

Командная работа демонстрирует высокий уровень взаимопомощи, лидерские качества и ориентацию на общий результат.

Оценка **«хорошо»** – средний уровень сформированности компетенции **УК-3**

выставляется студенту, если:

Роли распределены с учётом этапов проекта и индивидуальных предпочтений; зоны ответственности определены, возможна сменяемость ролей при необходимости.

Инструменты коллективной работы применяются корректно (Git, Kanban, Markdown), но не в полном объёме (например, Git без веток или Kanban обновляется с задержкой); синхронизация регулярная.

Студент активно участвует в обсуждениях, приводит 2–3 аргумента за свою позицию, учитывает и конструктивно обсуждает альтернативные точки зрения.

Обязательства выполняются полностью и в срок, стандарты оформления соблюдаются; студент ответственно подходит к качеству своих артефактов.

Командная работа эффективна, но не всегда инициируется самим студентом.

Оценка **«удовлетворительно»** – минимальный уровень сформированности компетенции **УК-3**

выставляется студенту, если:

Роли распределены формально, но не всегда соответствуют этапам и склонностям участников; есть дублирование или пробелы в зонах ответственности.

Инструменты коллективной работы используются, но не в полном объёме (например, Git без веток, Kanban не обновляется, синхронизация происходит с задержками).

Студент участвует в обсуждениях, но аргументация слабая (1–2 аргумента); мнение других учитывает без анализа, часто соглашается с большинством без собственной позиции.

Обязательства выполняются частично (70–80%) или с незначительными задержками; стандарты оформления соблюдаются не всегда.

Командная работа носит формальный характер, студент не проявляет инициативы.

Оценка **«неудовлетворительно»** – компетенция **УК-3** не сформирована
выставляется студенту, если:

Роли не распределены или распределены некорректно; каждый выполняет произвольные задачи, возникают конфликты из-за неопределённости зон ответственности.

Инструменты коллективной работы не используются или используются неэффективно (только пересылка файлов по почте/мессенджеру).

Студент не участвует в обсуждениях или участвует пассивно; не может аргументировать свою позицию, игнорирует мнение других.

Обязательства не выполняются в срок или выполняются некачественно; стандарты оформления игнорируются; студент мешает работе команды.

Командная работа отсутствует, студент действует изолированно или деструктивно.

Оценка **«отлично»** – высокий уровень сформированности компетенции **ПК-7**

выставляется студенту, если:

Документация проекта оформлена профессионально в Markdown с использованием таблиц, ссылок, встроенных изображений и диаграмм Mermaid; созданы шаблоны для типовых артефактов.

Применены базовые DevOps-практики: Git используется с полным циклом (Git Flow, ревью, теги); настроен полноценный CI/CD-пайплайн (сборка документации, проверка ссылок, публикация Wiki) через GitHub/GitLab Actions.

Студент демонстрирует глубокое знание отечественных платформ и инструментов (GitFlic, Yandex Cloud, VK Cloud Solutions и др.), имеет практический опыт работы с российским инструментом, может аргументированно выбрать между российским и зарубежным решением.

Используются профессиональные облачные сервисы (GitHub Pages, GitLab Pages) для развёртывания результатов; Wiki является полноценным сайтом документации с поиском, версионированием и адаптивным дизайном.

Все артефакты проекта (диаграммы, отчёты, API-документация) оформлены с соблюдением современных стандартов и доступны в едином репозитории.

Оценка **«хорошо»** – средний уровень сформированности компетенции **ПК-7**

выставляется студенту, если:

Документация оформлена в Markdown с использованием таблиц, ссылок, встроенных изображений; диаграммы созданы в Mermaid и встроены в документацию.

Git используется с ветками и пул-реквестами; настроен простой CI/CD-пайплайн (например, автоматическая сборка Markdown в PDF через GitHub Actions) – ознакомительно.

Студент знает 3+ отечественные платформы, провёл ознакомительное тестирование, может назвать основные отличия от зарубежных аналогов.

Используются облачные редакторы для совместной работы; Wiki опубликована на GitHub/GitLab, имеет боковую панель и базовую навигацию.

Артефакты проекта оформлены корректно, но могут иметь незначительные недочёты.

Оценка «удовлетворительно» – минимальный уровень сформированности компетенции **ПК-7**

выставляется студенту, если:

Документация оформлена в Markdown с базовыми элементами (заголовки, списки), но без таблиц и ссылок; диаграммы созданы вручную (скриншоты) или с помощью внешних инструментов, но не встроены в документацию.

Git используется на базовом уровне (commit, push, pull); ознакомление с CI/CD проведено теоретически (на примере, без настройки).

Студент знает 1–2 отечественные платформы (названия), но не использовал их; не может сравнить с зарубежными аналогами.

Используются базовые облачные сервисы (Google Docs); Wiki создана, но не структурирована, отсутствует навигация.

Артефакты проекта содержат ошибки, не позволяющие оценить качество документации.

Оценка «неудовлетворительно» – компетенция не сформирована **ПК-7**

выставляется студенту, если:

Документация не оформлена или оформлена в неструктурированном виде (простой текст без разметки); диаграммы отсутствуют.

Git не используется; CI/CD не применяется; понимание DevOps-практик отсутствует.

Студент не знает отечественные платформы и инструменты, не может назвать ни одной альтернативы зарубежным решениям.

Облачные сервисы не используются; Wiki проекта не опубликована или недоступна.

Артефакты проекта отсутствуют или содержат грубые ошибки, ключевые определения и выводы не представлены.

4. Критерии оценивания компетенций заданий закрытого типа

Оценка «отлично» выставляется студенту, если он правильно ответил на 88-100% от общего количества заданий. Компетенция УК-2 освоена на высоком уровне.

Оценка «хорошо» выставляется студенту, если он правильно ответил на 72-87% от общего количества заданий. Компетенция УК-2 освоена на среднем уровне.

Оценка «удовлетворительно» выставляется студенту, если он правильно ответил на 53-71% от общего количества заданий. Компетенция УК-2 освоена на минимальном уровне.

Оценка «неудовлетворительно» выставляется студенту, выставляется студенту, если он правильно ответил на менее 53% от общего количества заданий. Компетенция УК-2 не сформирована.

Оценка «отлично» выставляется студенту, если он правильно ответил на 88-100% от общего количества заданий. Компетенция УК-3 освоена на высоком уровне.

Оценка «хорошо» выставляется студенту, если он правильно ответил на 72-87% от общего количества заданий. Компетенция УК-3 освоена на среднем уровне.

Оценка «удовлетворительно» выставляется студенту, если он правильно ответил на 53-71% от общего количества заданий. Компетенция УК-3 освоена на минимальном уровне.

Оценка «неудовлетворительно» выставляется студенту, выставляется студенту, если он правильно ответил на менее 53% от общего количества заданий. Компетенция УК-3 не сформирована.

Оценка «отлично» выставляется студенту, если он правильно ответил на 88-100% от общего количества заданий. Компетенция ПК-7 освоена на высоком уровне.

Оценка «хорошо» выставляется студенту, если он правильно ответил на 72-87% от общего количества заданий. Компетенция ПК-7 освоена на среднем уровне.

Оценка «удовлетворительно» выставляется студенту, если он правильно ответил на 53-71% от общего количества заданий. Компетенция ПК-7 освоена на минимальном уровне.

Оценка «неудовлетворительно» выставляется студенту, выставляется студенту, если он правильно ответил на менее 53% от общего количества заданий. Компетенция ПК-7 не сформирована.