

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Порохня Андрей Алексеевич
Должность: И.о. директора института перспективной инженерии
Дата подписания: 19.06.2026 16:02:21
Уникальный программный ключ:
990bea3aeec48c23bd8699e48288f8d1960c600

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

УТВЕРЖДАЮ

канд. физ.-мат. наук, доцент,
Новикова Е.Н..

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

**для проведения текущего контроля успеваемости и промежуточной аттестации по производственной
преддипломной практике**

Командный проект по программной инженерии

Направление подготовки	09.03.04 «Программная инженерия»
Направленность (профиль)	Разработка и сопровождение программного обеспечения
Форма обучения	Очная
Год начала подготовки	2026
Реализуется в 8 семестре	

Ставрополь, 2026

Предисловие

1. Назначение:

Фонд оценочных средств по учебной практике предназначен для объективной оценки уровня сформированности компетенций.

2. ФОС является приложением к программе «Производственная преддипломная практика (Командный проект по программной инженерии)» в соответствии с образовательной программой по направлению подготовки 09.03.04 «Программная инженерия».

3. Разработчик: заведующий межинститутской базовой кафедрой Новикова Е.Н.

4. Проведена экспертиза ФОС. Члены экспертной группы:

Председатель: И. В. Азаров, и.о. директора департамента цифровых, робототехнических систем и электроники

Члены комиссии:

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

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

Представитель организации работодателя:

Свистунов И.В., Руководитель рабочей группы ООО «Стилсофт».

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

«_____» _____

5. Срок действия ФОС: на срок реализации образовательной программы.

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

Уровни сформированности и 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
Компетенция: ПК-1 – Способен осуществлять управление программными проектами: планировать, контролировать сроки и ресурсы, применять системы контроля версий и инструменты управления задачами				
ИД-1пк-1 Применяет методы декомпозиции работ (WBS), оценки трудоёмкости (PERT, Planning Poker) и календарного планирования (диаграмма Ганта, критический путь) для формирования плана проекта	Не способен составить WBS и диаграмму Ганта. Не владеет методами оценки трудоёмкости. План проекта отсутствует или не соответствует задаче.	Составляет простейшую WBS (2–3 уровня) и линейный график работ. Оценивает трудоёмкость интуитивно, без использования методов PERT или Planning Poker. План проекта имеет существенные пробелы.	Составляет детальную WBS (3–4 уровня) и диаграмму Ганта с идентификацией критического пути. Применяет PERT или Planning Poker для оценки трудоёмкости. План проекта в целом корректен.	Составляет многоуровневую WBS (4+ уровней), детализированную диаграмму Ганта с ресурсным планированием. Комбинирует методы оценки трудоёмкости (PERT + Planning Poker). План проекта оптимизирован по срокам и ресурсам.
ИД-2пк-1 Использует системы контроля версий (Git, Git Flow) и платформы совместной разработки (GitLab, GitHub, GitFlic) для организации	Не использует Git. Код не версионизируется. Не умеет создавать ветки и выполнять слияние. Командная работа не организована.	Использует Git на базовом уровне: создаёт репозиторий, выполняет commit, push, pull. Ветвление и слияние применяет эпизодически.	Уверенно использует Git: создаёт и сливает ветки, разрешает конфликты, применяет Git Flow. Работает в команде через pull requests.	В совершенстве владеет Git: использует продвинутые стратегии ветвления (Git Flow, GitHub Flow), организует code review, настраивает защиту веток. Использует отечественную

командной работы				платформу GitFlic.
ИД-3пк-1 Применяет инструменты управления задачами для отслеживания прогресса (burndown chart), ведения бэклога и контроля исполнения	Не использует инструменты управления задачами. Бэклог отсутствует. Прогресс не отслеживается.	Использует простейший инструмент (Trello или доска задач). Ведёт бэклог без приоритезации. Burndown chart не строит или строит формально.	Уверенно работает в Jira, YouTrack или Trello: ведёт приоритезированный бэклог, планирует спринты, отслеживает прогресс по burndown chart.	В совершенстве владеет инструментами: настраивает дашборды, автоматические отчёты, использует расширенную аналитику (velocity, cycle time). Интегрирует задачу с Git и CI/CD.
ИД-4пк-1 Анализирует план-фактные отклонения по срокам, ресурсам и бюджету, управляет изменениями и рисками, разрабатывает корректирующие действия	Не анализирует отклонения. Изменения и риски не управляются. Корректирующие действия не разрабатывает.	Проводит поверхностный анализ план-фактных отклонений (только по срокам). Управление изменениями и рисками – эпизодическое.	Систематически анализирует отклонения по срокам, ресурсам и бюджету. Документирует изменения и риски. Разрабатывает адекватные корректирующие действия.	Проводит глубокий анализ отклонений с выявлением причин. Использует формальные методы управления рисками. Разрабатывает и реализует эффективные корректирующие действия.
Компетенция: ПК-3 – Владеет навыками разработки программного обеспечения: способен реализовывать алгоритмы, писать код на современных языках программирования, создавать веб- и desktop-приложения, работать с базами данных				
ИД-1пк-3 Разрабатывает алгоритмы и реализует их на современных языках программирования (C++, C#, Python, Java, JavaScript/TypeSc	Не способен разработать алгоритм решения задачи. Код отсутствует или содержит грубые ошибки. Структуры данных выбраны неверно.	Разрабатывает простейшие линейные алгоритмы. Реализует код на одном языке программирования (базовый уровень). Использует	Разрабатывает разветвлённые и циклические алгоритмы средней сложности. Уверенно пишет код на 1–2 языках. Выбирает оптимальные	Разрабатывает сложные алгоритмы (рекурсия, динамическое программирование, графы). Свободно пишет код на 2+ языках. Применяет продвинутые структуры данных

ript, Go) с использованием эффективных структур данных и стандартных библиотек		стандартные структуры данных (массивы, списки).	структуры данных (словари, множества, деревья).	(хеш-таблицы, кучи, В-деревья).
ИД-2пк-3 Создаёт клиент-серверные, веб- (frontend + backend) и desktop-приложения с графическим интерфейсом, применяя современные фреймворки (React, Spring, Django, WPF) и протоколы (REST, gRPC, WebSocket)	Не способен создать приложение с графическим интерфейсом. Клиент-серверное взаимодействие не реализовано.	Создаёт простейшее desktop-приложение с графическим интерфейсом. Клиент-серверное взаимодействие на уровне HTTP-запросов (без соблюдения REST).	Создаёт веб-приложение с разделением на frontend и backend (один фреймворк). Реализует REST API. Использует асинхронные запросы.	Создаёт полноценное клиент-серверное приложение с современным frontend (React) и backend (Spring/Django). Использует REST, gRPC или WebSocket.
ИД-3пк-3 Проектирует, мигрирует и оптимизирует запросы к реляционным и нереляционным БД с использованием SQL и ORM (Hibernate, Entity Framework, Django ORM)	Не умеет проектировать БД. SQL-запросы содержат грубые ошибки. ORM не использует.	Проектирует простейшую реляционную БД (2–3 таблицы). Пишет базовые SQL-запросы. Начинает использовать ORM (базовые CRUD-операции).	Проектирует нормализованную БД (3НФ) с индексами. Пишет сложные SQL-запросы (JOIN, GROUP BY, подзапросы). Уверенно использует ORM для миграций и запросов.	Проектирует высокопроизводительную БД с учётом нагрузки (денормализация, партиционирование). Оптимизирует запросы (EXPLAIN). Использует реляционные и нереляционные БД.
ИД-4пк-3 Пишет модульный,	Код неструктурированный,	Код частично структурирован (есть	Код модульный, соблюдает стандарты	Код идеально структурирован, полностью

документированный и тестируемый код, соблюдая стандарты оформления и принципы SOLID, использует системы сборки (Make, Gradle, npm)	недокументированный. Принципы SOLID не соблюдаются. Системы сборки не используются.	функции/методы). Документация отсутствует или минимальна. Начинает применять некоторые принципы SOLID.	оформления. Применяет принципы SOLID на практике. Пишет простые unit-тесты. Использует системы сборки.	документирован. Строго соблюдает SOLID и стандарты кодирования. Пишет покрытые тестами модули. Настраивает CI для автоматической сборки и тестирования.
Компетенция: ПК-10 Готов соблюдать правовые нормы, стандарты и требования информационной безопасности, участвовать в разработке нормативной и технической документации, учитывать законодательство в области ИТ				
ИД-1пк-10 Применяет правовые нормы (лицензирование ПО, авторское право, ФЗ-152, ФЗ-149) и стандарты ИБ (ГОСТ Р ИСО/МЭК 27001, ГОСТ Р 56545-2015) при	Не знаком с правовыми нормами в ИТ. Не учитывает требования лицензирования и авторского права. ФЗ-152 и ФЗ-149 игнорирует.	Знает основные правовые нормы, но применяет их эпизодически. Указывает лицензию на ПО формально. ФЗ-152 учитывает на базовом уровне.	Систематически применяет правовые нормы при разработке и эксплуатации ПО. Корректно выбирает лицензии. Соблюдает требования ФЗ-152 и ФЗ-149.	В совершенстве владеет правовыми нормами. Участвует в выборе лицензионной политики организации. Обеспечивает полное соответствие ПО требованиям законодательства и стандартам ИБ.

разработке и эксплуатации ПО				
ИД-2пк-10 Реализует меры защиты информации: аутентификацию (JWT, OAuth2), авторизацию (RBAC), шифрование, защиту от типовых уязвимостей OWASP Top 10 (SQL-инъекции, XSS, CSRF, IDOR) с учётом требований регуляторов (ФСТЭК)	Не реализует меры защиты информации. Аутентификация и авторизация отсутствуют. Код уязвим к атакам OWASP Top 10.	Реализует базовую аутентификацию (логин/пароль) без хеширования. Авторизация на уровне одной роли. Частично защищается от SQL-инъекций.	Реализует JWT-аутентификацию, ролевую модель (RBAC). Использует хеширование паролей (bcrypt). Защищается от основных уязвимостей OWASP.	Реализует OAuth2, JWT с refresh-токенами, многофакторную аутентификацию. Применяет шифрование. Использует SAST/DAST. Полностью защищает приложение от OWASP Top 10.
ИД-3пк-10 Разрабатывает нормативную и техническую документацию по информационной безопасности: политики парольной защиты, модели угроз, инструкции пользователя, регламенты реагирования на инциденты	Не разрабатывает документацию по ИБ. Политики и инструкции отсутствуют.	Разрабатывает простейшие документы: базовую инструкцию пользователя по безопасности. Политика паролей – минимальная.	Разрабатывает комплект документов: политику парольной защиты, модель угроз (формально), инструкцию пользователя по ИБ.	Разрабатывает полный комплект документации: детальную модель угроз (с активами, угрозами, оценкой рисков), политики безопасности, инструкции, регламент реагирования на инциденты. Учитывает требования ФСТЭК.

2. Оценочные средства по преддипломной практике

2.1. Задания, позволяющие оценить знания, полученные на практике(базовый уровень)

Контролируемые компетенции или их части (код компетенции)	Формулировка задания
ПК-1	Задание 1
	1.1. Изучить методы декомпозиции работ (WBS) и календарного планирования (диаграмма Ганта, критический путь).
	1.2. Изучить методы оценки трудоёмкости программных проектов (PERT, Planning Poker, COCOMO II).
	1.3. Ознакомиться с системами контроля версий (Git, Git Flow) и платформами совместной разработки (GitLab, GitHub, GitFlic).
	1.4. Изучить инструменты управления задачами (Jira, Trello, YouTrack, GitLab Issues) и методы Agile (Scrum, Kanban).
	1.5. Ознакомиться с методами анализа план-фактных отклонений и управления рисками в IT-проектах.
ПК-1 ПК-3	Задание 2
	1.6. Разработать техническое задание (ТЗ) на разработку программного обеспечения в соответствии с ГОСТ.
	1.7. Составить план и этапы разработки с использованием WBS и диаграммы Ганта.
	1.8. Изучить современные технологии разработки программного обеспечения (структурное, объектно-ориентированное, компонентное).

Контролируемые компетенции или их части (код компетенции)	Формулировка задания
	1.9. Ознакомиться со стандартами оформления кода и принципами SOLID.
ПК-3	Задание 3
	1.10. Изучить современные языки программирования (Python, Java, C++, Go, JavaScript/TypeScript) и их стандартные библиотеки.
	1.11. Изучить способы создания программных интерфейсов (REST API, gRPC, WebSocket).
	1.12. Изучить методы проектирования и оптимизации запросов к реляционным и нереляционным базам данных (SQL, ORM).
	1.13. Изучить системы сборки и автоматизации (Make, Gradle, npm, Maven).
ПК-10	Задание 4
	1.14. Изучить правовые нормы в сфере ИТ (лицензирование ПО, авторское право, ФЗ-152 «О персональных данных», ФЗ-149 «Об информации»).
	1.15. Изучить стандарты информационной безопасности (ГОСТ Р ИСО/МЭК 27001, ГОСТ Р 56545-2015).
	1.16. Изучить методы защиты информации: аутентификацию (JWT, OAuth2), авторизацию (RBAC), шифрование.
	1.17. Изучить типовые уязвимости веб-приложений (OWASP Top 10) и методы защиты от них (SQL-инъекции, XSS, CSRF, IDOR).
	1.18. Изучить требования регуляторов (ФСТЭК) к разработке и эксплуатации программного

Контролируемые компетенции или их части (код компетенции)	Формулировка задания
	обеспечения.
ПК-10	Задание 5
	1.19. Изучить структуру и порядок разработки нормативной и технической документации по информационной безопасности (политики парольной защиты, модели угроз, инструкции пользователя, регламенты реагирования на инциденты).

3.2 Задания, позволяющие оценить знания, полученные на практике(повышенный уровень)

Контролируемые компетенции или их части (код компетенции)	Формулировка задания
ПК-1	Задание 1
	2.1. Проанализировать и сравнить классические (Waterfall, V-model) и гибкие (Scrum, Kanban, XP) методологии управления проектами, обосновать выбор методологии для конкретного проекта.
	2.2. Разработать детальный план управления проектом с использованием WBS (до 4-го уровня), диаграммы Ганта и метода критического пути (CPM), оценить трудоёмкость методами PERT и Planning Poker.
	2.3. Настроить стратегию ветвления Git Flow для команды из 3–5 человек, организовать процесс code review и CI/CD (GitLab CI / GitHub Actions).
	2.4. Разработать систему метрик для контроля исполнения проекта (velocity, cycle time, burndown chart, cumulative flow), настроить дашборды в Jira/YouTrack.

Контролируемые компетенции или их части (код компетенции)	Формулировка задания
	2.5. Провести анализ план-фактных отклонений по срокам и бюджету, разработать корректирующие действия и план управления рисками (реестр рисков, матрица вероятности и воздействия).
ПК-3	Задание 2
	2.6. Разработать архитектуру программной системы средней сложности (5–10 микросервисов/модулей) с использованием нотаций C4 (Context, Container, Component) и UML (диаграммы классов, последовательности, состояний).
	2.7. Реализовать прототип программного продукта с использованием не менее двух технологий (например, Python/Java backend + React frontend), обеспечив асинхронное взаимодействие через REST/gRPC и WebSocket.
	2.8. Спроектировать высокопроизводительную базу данных (реляционную + нереляционную) с учётом требований к нагрузке (индексы, партиционирование, шардирование), оптимизировать сложные SQL-запросы с использованием EXPLAIN.
	2.9. Разработать модульный, документированный и тестируемый код с соблюдением принципов SOLID, настроить статический анализ (SonarQube) и автоматическое тестирование (unit + интеграционные тесты) в CI/CD.
	2.10. Настроить контейнеризацию приложения с помощью Docker (multi-stage builds) и оркестрацию через Docker Compose для локальной разработки.
ПК-10	Задание 3
	2.11. Разработать политику лицензирования для

Контролируемые компетенции или их части (код компетенции)	Формулировка задания
	создаваемого программного продукта (выбор между MIT, GPL, Apache, проприетарной лицензией) с обоснованием выбора.
	2.12. Разработать модель угроз информационной безопасности для проектируемой системы (идентификация активов, угроз, уязвимостей, оценка рисков) в соответствии с методикой ФСТЭК.
	2.13. Реализовать комплексную систему аутентификации и авторизации (JWT с refresh-токенами, OAuth2, RBAC), настроить защиту от типовых уязвимостей OWASP Top 10 (SQL-инъекции, XSS, CSRF, IDOR, небезопасная десериализация).
	2.14. Разработать комплект нормативной и технической документации по информационной безопасности: политику парольной защиты, инструкцию пользователя, регламент реагирования на инциденты.
	2.15. Провести анализ соответствия разрабатываемого ПО требованиям Ф3-152 «О персональных данных» и Ф3-149 «Об информации», подготовить рекомендации по доработке.
ПК-1, ПК-3, ПК-10 (комплексное)	Задание 4 (интеграционное)
	2.16. Разработать полный пакет проектной документации для командного проекта: Техническое задание (ТЗ), План управления проектом, Архитектурную документацию (C4/UML/ADR), Модель угроз ИБ, Политики безопасности, API-спецификацию (OpenAPI), Руководство пользователя и Итоговый отчёт.
	2.17. Провести публичную защиту результатов командного проекта с презентацией (10–15 слайдов) и демонстрацией работающего программного продукта, CI/CD пайплайна и системы мониторинга

Контролируемые компетенции или их части (код компетенции)	Формулировка задания
	(Prometheus/Grafana).
	2.18. Выполнить ретроспективный анализ проекта: выявить успешные практики и ошибки, предложить улучшения для будущих проектов, оценить эффективность командной работы и использованных методологий.

3. Критерии оценивания компетенций

Оценка «отлично» (5 баллов) – высокий уровень

Оценка «отлично» выставляется обучающемуся, если студент:

Знает:

- методы декомпозиции работ (WBS) и календарного планирования (диаграмма Ганта, критический путь);
- методы оценки трудоёмкости программных проектов (PERT, Planning Poker, COCOMO II);
- системы контроля версий (Git, Git Flow) и платформы совместной разработки (GitLab, GitHub, GitFlic);
- инструменты управления задачами (Jira, Trello, YouTrack, GitLab Issues) и методы Agile (Scrum, Kanban);
- современные языки программирования (Python, Java, C++, Go, JavaScript/TypeScript) и их стандартные библиотеки;
- способы создания программных интерфейсов (REST API, gRPC, WebSocket);
- методы проектирования и оптимизации запросов к реляционным и нереляционным базам данных (SQL, ORM);
- правовые нормы в сфере ИТ (лицензирование ПО, авторское право, ФЗ-152, ФЗ-149);
- стандарты информационной безопасности (ГОСТ Р ИСО/МЭК 27001, ГОСТ Р 56545-2015);
- типовые уязвимости веб-приложений (OWASP Top 10) и методы защиты от них.

Умеет:

- составлять WBS, диаграмму Ганта и определять критический путь проекта;
- применять методы PERT и Planning Poker для оценки трудоёмкости;
- организовывать командную работу с использованием Git Flow и платформ совместной разработки;
- вести бэклог, планировать спринты и отслеживать прогресс по burndown chart;
- разрабатывать архитектуру программной системы с использованием нотаций UML и C4;
- создавать клиент-серверные и веб-приложения с применением современных фреймворков (React, Spring, Django, WPF);
- проектировать нормализованные базы данных и оптимизировать сложные SQL-запросы;
- реализовывать меры защиты информации (JWT, OAuth2, RBAC, шифрование);
- проводить анализ план-фактных отклонений и управлять рисками проекта.

Критическое мышление:

- практического применения моделей и методов менеджмента в управлении программными проектами;
- использования систем контроля версий для организации командной разработки;
- применения современных технологий разработки ПО (структурное, объектно-ориентированное, компонентное);
- написания модульного, документированного и тестируемого кода с соблюдением принципов SOLID;
- оценки временной и ёмкостной сложности алгоритмов и программного обеспечения;
- создания современных программных интерфейсов (REST API, gRPC, WebSocket);
- разработки нормативной и технической документации по информационной безопасности;
- защиты программного обеспечения от типовых уязвимостей OWASP Top 10.

Оценка «хорошо» (4 балла) – средний уровень

Оценка «хорошо» выставляется обучающемуся, если студент:

Знает:

- только основные методы декомпозиции работ и календарного планирования;
- только основные методы оценки трудоёмкости программных проектов;
- только основные возможности систем контроля версий (Git) и платформ совместной разработки;
- только базовые инструменты управления задачами (Trello или Jira);
- современные языки программирования на базовом уровне (1–2 языка);
- только основные способы создания программных интерфейсов (REST API);
- только базовые принципы проектирования баз данных (SQL, нормализация до 2НФ);
- только основные правовые нормы в сфере ИТ (лицензирование, ФЗ-152);
- только основные типовые уязвимости OWASP Top 10.

Умеет:

- составлять простейшую WBS (2–3 уровня) и линейный график работ;
- применять интуитивную оценку трудоёмкости (без формальных методов PERT/Planning Poker);
- использовать Git на базовом уровне (commit, push, pull, branch, merge);
- вести бэклог в простейшем инструменте без приоритезации;

- разрабатывать простейшую архитектуру (без формальных нотаций UML/C4);
- создавать простые desktop-приложения с графическим интерфейсом;
- проектировать простейшую реляционную БД (2–3 таблицы) и писать базовые SQL-запросы;
- реализовывать базовую аутентификацию (логин/пароль) без хеширования;
- проводить поверхностный анализ план-фактных отклонений (только по срокам).
- *Допускает незначительные ошибки при выполнении вышеуказанных действий.*
- *Владеет навыками (с незначительными ошибками):*
- практического применения базовых моделей менеджмента в управлении ПО;
- использования Git для организации командной разработки;
- применения основных технологий разработки ПО (структурное, объектно-ориентированное);
- написания структурированного кода с частичным соблюдением принципов SOLID;
- оценки временной сложности простейших алгоритмов;
- создания базовых программных интерфейсов (REST API);
- разработки минимальной документации по информационной безопасности (инструкция пользователя).

Оценка «удовлетворительно» (3 балла) – минимальный уровень

Оценка «удовлетворительно» выставляется обучающемуся, если студент:

Знает (имеет общее представление):

- о методах декомпозиции работ и календарного планирования;
- о методах оценки трудоёмкости программных проектов;
- о системах контроля версий и платформах совместной разработки;
- об инструментах управления задачами;
- о современных языках программирования (без привязки к конкретным);
- о способах создания программных интерфейсов;
- о правовых нормах в сфере ИТ (общие понятия);
- об информационной безопасности (общие понятия).

Умеет (использует основные умения):

- использовать основные модели менеджмента в управлении проектом;
- организовать базовые работы по управлению проектом ИС;
- использовать основные технологии разработки ПО;
- вычислять временную сложность простейших алгоритмов (визуально);
- создавать простейшие программные интерфейсы (без соблюдения стандартов).

Не полностью освоил учебные умения, испытывает затруднения при их применении.

Владеет навыками (испытывает затруднения):

- практического применения моделей и методов менеджмента в управлении ПО;
- проведения переговоров и осуществления контроля версий;
- применения современных технологий разработки ПО;
- оценки временной и ёмкостной сложности ПО;
- создания современных программных интерфейсов.

Оценка «неудовлетворительно» (2 балла) – минимальный уровень не достигнут

Оценка «неудовлетворительно» выставляется обучающемуся, который:

Знает (низкий уровень знаний):

- концептуальных моделей менеджмента;
- методов контроля проекта;
- современных технологий разработки ПО (структурное, объектно-ориентированное);
- методов оценки временной и ёмкостной сложности программного обеспечения;
- способов создания программных интерфейсов;
- правовых норм и стандартов информационной безопасности.

Умеет (допускает грубые ошибки):

- использовать основные модели менеджмента в управлении проектом;
- организовать работы по управлению проектом ИС;
- использовать современные технологии разработки ПО;
- вычислять временную и ёмкостную сложность ПО;
- создавать интуитивно понятные программные интерфейсы;
- реализовывать меры защиты информации.

- Владеет навыками (допускает грубые ошибки):
- практического применения моделей и методов менеджмента в управлении ПО;
- проведения переговоров и осуществления контроля версий;
- применения современных технологий разработки ПО;
- оценки временной и ёмкостной сложности ПО;
- создания современных программных интерфейсов;
- разработки документации по информационной безопасности.

4. Описание шкалы оценивания

Максимальная сумма баллов по практике устанавливается в 100 баллов и переводится в оценку по 5-балльной системе в соответствии со шкалой:

Шкала соответствия рейтингового балла 5-балльной системе

Рейтинговый балл	Оценка по 5-балльной системе
88 – 100	Отлично
72 – 87	Хорошо
53 – 71	Удовлетворительно
< 53	Неудовлетворительно

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

На каждом этапе практики осуществляется текущий контроль за процессом формирования компетенций. Предлагаемые обучающемуся задания позволяют проверить следующие компетенции: ПК-1, ПК-3, ПК-10.

При проверке заданий, оцениваются:

- Последовательность и рациональность выполнения;
- логичность изложения;
- полнота описания.

При проверке отчетов оцениваются

- самостоятельность выполнения;
- качество оформления представления результатов работы;

□ уровень защиты и ответов на вопросы. При

защите отчета оцениваются:

- самостоятельность выполнения;
- качество оформления представления результатов работы;
- уровень защиты и ответов на вопросы.

По результатам выступления и подготовленной презентации студент обосновывает результаты выполненной работы в рамках учебной практики и подтверждает перечнем представленных документов:

- индивидуальное задание,
- дневник,
- отчет.