

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Крюков Вадим Николаевич

Должность: Проректор по образовательной деятельности и инновационной политике

Дата подписания: 17.06.2026 16:21:18

Уникальный программный ключ:

1b0adb7fd710f6a0705d90c58682bd0c5f2f25b2

Министерство науки и высшего образования РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Заполярный государственный университет им. Н. М. Федоровского»

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине

Технологии коллективной разработки программных систем

Направление подготовки 09.04.03 «Прикладная информатика»

Профиль «Информационные системы и технологии в бизнесе»

Уровень образования: магистратура

Кафедра «Информационные системы и технологии»

Разработчик ФОС:

доцент, Е.А. Дыптан _____

Е.А. Дыптан

Оценочные материалы по дисциплине рассмотрены и одобрены на заседании кафедры, протокол от 10.04.2026г. № 5.

Заведующий кафедрой _____ к.э.н., Беляев И.С.

Фонд оценочных средств по дисциплине Технологии коллективной разработки программных систем для текущей/ промежуточной аттестации разработан в соответствии с Федеральным государственным образовательным стандартом высшего образования по специальности / направлению подготовки 09.04.03 Прикладная информатика на основе Рабочей программы дисциплины Технологии коллективной разработки программных систем, утвержденной решением ученого совета от г., Положения о формировании Фонда оценочных средств по дисциплине (ФОС), Положения о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся ЗГУ, Положения о государственной итоговой аттестации (ГИА) выпускников по образовательным программам высшего образования в ЗГУ им. Н.М. Федоровского.

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами образовательной программы

Таблица 1. Компетенции и индикаторы их достижения

Код и наименование компетенции	Индикаторы достижения
УК-3 Способен организовывать и руководить работой команды, выработывая командную стратегию для достижения поставленной цели	УК-3.2 Осуществляет организацию работы команды и выработку командной стратегии
	УК-3.3 Разрабатывает на основе анализа и оценки методов организации и управления коллективом командную стратегию
ПК-4 Способен управлять инфраструктурой коллективной среды разработки	ПК-4.1 Выбирает и применяет практики и методологии управления разработкой ПО
	ПК-4.2 Комбинирует навыки выбора средств создания и ведения репозитория, учета задач, сборки и непрерывной интеграции базы знаний

Таблица 2. Паспорт фонда оценочных средств

№п/п	Контролируемые разделы(темы) дисциплины	Кодрезультатаобучения по дисциплине/ модулю	Оценочные средства текущей		Оценочные средства промежуточной	
			Наименование	Форма	Наименование	Форма
4 семестр						

2. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующие процесс формирования компетенций в ходе освоения образовательной программы.

2.1. Задания для текущего контроля успеваемости

1. Основы коллективной разработки

1. Что такое коллективная разработка программного обеспечения? Приведите примеры.

2. Какие основные методологии коллективной разработки существуют?

(например, Agile, Scrum, Waterfall)

3. В чем заключается разница между Agile и Waterfall?
4. Что такое DevOps? Как он применяется в коллективной разработке?
5. Как организуется работа команды в процессе коллективной разработки?
6. Что такое Git? Какую роль он играет в коллективной разработке?
7. Как работает система контроля версий? Приведите пример.
8. Что такое CI/CD? Где это применяется?
9. Как организуется автоматизация тестирования в коллективной разработке?
10. Какие инструменты используются для управления задачами в команде?

2. Управление проектами

11. Что такое жизненный цикл программного обеспечения? Опишите его этапы.
12. Как проводится планирование проекта в коллективной разработке?
13. Что такое Product Backlog? Как он используется в Scrum?
14. Как организуется распределение ролей в команде разработчиков?
15. Что такое Sprint в Scrum? Как он организуется?
16. Как проводится оценка трудозатрат при разработке программных систем?
17. Что такое Burndown Chart? Для чего он используется?
18. Как организуется ретроспектива в Agile-командах?
19. Какие метрики используются для оценки эффективности команды?
20. Как управлять рисками в коллективной разработке?

3. Инструменты и технологии

21. Как работает Git? Приведите примеры команд.
22. Что такое GitHub и GitLab? В чем их различия?
23. Как организуется совместная работа над кодом с использованием Git?
24. Что такое Pull Request? Как он используется в Git?
25. Как работает система ветвления в Git? Приведите пример.
26. Какие инструменты используются для автоматизации сборки проекта?

(например, Jenkins, Travis CI)

27. Что такое Docker? Как он применяется в коллективной разработке?
28. Как организуется контейнеризация приложений в команде разработчиков?
29. Какие инструменты используются для мониторинга производительности системы?
30. Как работают системы логирования в коллективной разработке?

4. Совместная работа и коммуникация

31. Как организуется коммуникация между участниками команды?
32. Какие инструменты используются для общения в команде? (например, Slack, Microsoft Teams)
33. Как проводятся ежедневные встречи (Daily Standup) в Scrum?
34. Как организуется документация проекта в коллективной разработке?
35. Какие подходы используются для написания технической документации?
36. Как организуется обучение новых участников команды?
37. Как решаются конфликты в команде разработчиков?
38. Как организуется обратная связь в процессе разработки?
39. Какие методы мотивации команды применяются в коллективной разработке?
40. Как поддерживается культура командной работы?

5. Контроль качества

41. Как организуется тестирование в коллективной разработке?
42. Что такое юнит-тесты? Как они пишутся и используются?
43. Какие виды тестирования применяются в коллективной разработке? (например, интеграционное, нагрузочное)
44. Как работает автоматизированное тестирование? Приведите пример.
45. Что такое регрессионное тестирование? Где оно применяется?
46. Как организуется ручное тестирование в команде разработчиков?
47. Какие инструменты используются для отслеживания ошибок? (например, Jira, Trello)
48. Как организуется исправление багов в коллективной разработке?
49. Как проводится анализ качества кода?
50. Какие стандарты качества применяются в коллективной разработке?

6. Безопасность и управление данными

51. Как организуется защита данных в коллективной разработке?
52. Что такое шифрование данных? Как оно применяется в разработке?
53. Как организуется аутентификация и авторизация в программных системах?
54. Как предотвратить SQL-инъекции в коллективной разработке?
55. Что такое аудит безопасности? Зачем он нужен?
56. Как организуется мониторинг безопасности системы?
57. Какие стандарты безопасности применяются для программных систем?
58. Какие методы защиты данных от утечек существуют?
59. Как организуется управление доступом к данным?
60. Какие угрозы безопасности существуют для программных систем?

7. Масштабирование и оптимизация

61. Как организуется масштабирование программных систем?
62. Что такое микросервисная архитектура? Где она применяется?
63. Как работает шардирование баз данных? Приведите пример.
64. Как организуется репликация данных в программных системах?
65. Какие подходы используются для минимизации времени простоя системы?
66. Как организуется параллельная обработка данных в команде разработчиков?
67. Какие инструменты используются для анализа производительности системы?
68. Как организуется оптимизация кода в коллективной разработке?
69. Какие подходы используются для снижения затрат на разработку?
70. Как организуется рефакторинг кода в команде?

8. Перспективы развития

71. Как искусственный интеллект меняет подходы к коллективной разработке?
72. Какие тренды наблюдаются в развитии технологий коллективной разработки?
73. Как блокчейн может быть интегрирован в программные системы?
74. Какие перспективы у использования IoT в коллективной разработке?
75. Как развивается использование облачных технологий в разработке?
76. Как меняются требования к программным системам с развитием технологий?
77. Какие вызовы стоят перед разработчиками программных систем?

78. Как искусственный интеллект влияет на управление бизнес-процессами?
79. Какие инновации ожидаются в области коллективной разработки?
80. Как глобализация влияет на развитие технологий коллективной разработки?
9. Дополнительные вопросы
81. Как мобильные технологии влияют на коллективную разработку?
82. Какие подходы используются для создания адаптивных программных систем?
83. Как строится система логирования в программных системах?
84. Как организуется работа с большими данными (Big Data) в коллективной разработке?
85. Какие методы анализа данных применяются в современных программных системах?
86. Как внедряется машинное обучение в программные системы?
87. Как организуется работа с облачными сервисами в коллективной разработке?
88. Какие подходы используются для масштабирования программных систем?
89. Какие инструменты используются для защиты от DDoS-атак в программных системах?
90. Какие подходы используются для минимизации затрат на разработку и поддержку программных систем?

2.2 Темы письменных работ (эссе, рефераты, курсовые работы и др.)

Эссе (20 тем)

1. Роль Agile-методологий в современной коллективной разработке.
2. Этические аспекты совместной работы над программными системами.
3. Преимущества и ограничения использования Git в коллективной разработке.
4. Будущее коллективной разработки: тренды и прогнозы развития.
5. Как искусственный интеллект меняет подходы к коллективной разработке?
6. Сравнение Scrum и Kanban: что эффективнее для команды разработчиков?
7. Влияние удаленной работы на коллективную разработку.
8. Как обеспечить универсальную доступность (Accessibility) в программных системах?
9. Роль цвета и типографики в интерфейсах, создаваемых командой разработчиков.
10. Безопасность данных в контексте коллективной разработки.
11. Как блокчейн-технологии могут быть интегрированы в коллективную разработку?
12. Коллективная разработка как инструмент повышения эффективности бизнеса.
13. Проблемы цифрового неравенства в контексте использования программных систем.
14. Как персонализация влияет на успех программных продуктов?
15. Роль аналитики данных в улучшении пользовательского опыта через коллективную разработку.
16. Как коллективная разработка поддерживает глобализацию продуктов?
17. Проблемы правового регулирования использования программных систем.

18. Как коллективная разработка помогает малому бизнесу выйти на международный уровень?

19. Роль CRM-систем в управлении клиентской базой через коллективную разработку.

20. Этические вопросы использования больших данных в программных системах.

Рефераты (20 тем)

21. История развития методологий коллективной разработки и их влияние на современные технологии.

22. Основные принципы коллективной разработки: удобство, эстетика, функциональность.

23. Принципы создания адаптивных и отзывчивых программных систем.

24. Принципы разработки программных систем для мобильных устройств.

25. Как работает система взаимодействия пользователя с программным продуктом (UX/UI)?

26. Алгоритмы анализа пользовательского поведения в программных системах.

27. Принципы работы систем лояльности через программные системы.

28. Как используются облачные технологии в коллективной разработке?

29. Принципы работы систем управления заказами через программные системы.

30. Роль маркетинга в продвижении программных продуктов.

31. Принципы работы систем управления складскими запасами через программные системы.

32. Методы защиты персональных данных в программных системах.

33. Принципы работы систем управления цепями поставок через программные системы.

34. Как работают системы анализа пользовательского поведения через программные системы?

35. Принципы работы систем управления финансами в бизнесе через программные системы.

36. Принципы работы систем управления рисками через программные системы.

37. Как используются технологии IoT в программных системах?

38. Принципы работы систем управления документооборотом через программные системы.

39. Принципы работы систем управления качеством продукции через программные системы.

40. Принципы работы систем управления маркетинговыми кампаниями через программные системы.

Курсовые работы (20 тем)

41. Разработка программной системы для управления проектами.

42. Создание системы рекомендаций для клиентов интернет-магазина через программное обеспечение.

43. Проектирование программной системы для управления рисками.

44. Разработка программной системы для аналитики данных о клиентах.

45. Создание программной системы для управления складскими запасами.

46. Разработка программной системы для системы электронных платежей.

47. Проектирование программной системы для управления лояльностью клиентов.

48. Создание программной системы для управления маркетинговыми

кампаниями.

49. Разработка программной системы для управления цепями поставок.
50. Проектирование программной системы для управления рисками в бизнесе.
51. Создание программной системы для управления финансами компании.
52. Разработка программной системы для управления документооборотом.
53. Проектирование программной системы для управления качеством продукции.
54. Создание программной системы для управления портфелем инвестиций.
55. Разработка программной системы для управления экологическими

показателями.

56. Проектирование программной системы для управления фермерским хозяйством.

57. Создание программной системы для управления медицинскими данными.
58. Разработка программной системы для управления учебным процессом онлайн.
59. Проектирование программной системы для управления задачами удаленных

команд.

60. Создание программной системы для управления энергопотреблением здания.

1. Текущий контроль знаний

Тестовые вопросы

1. Что такое Git? Какую роль он играет в коллективной разработке?
2. Какие основные методологии разработки программных систем существуют? (например, Agile, Scrum)
3. В чем заключается разница между централизованной и распределенной системой контроля версий?
4. Что такое CI/CD? Где это применяется?
5. Как организуется автоматизация тестирования в коллективной разработке?
6. Что такое Product Backlog? Как он используется в Scrum?
7. Как проводится планирование проекта в Agile-командах?
8. Что такое Pull Request? Как он используется в Git?
9. Как работает система ветвления в Git? Приведите пример.
10. Какие инструменты используются для управления задачами в команде?

Практические задания

11. Создайте репозиторий на GitHub и добавьте в него файл с кодом.
12. Напишите скрипт для автоматизации сборки проекта с использованием Jenkins.
13. Разработайте план управления задачами для команды разработчиков.
14. Реализуйте систему тестирования для простого приложения.
15. Настройте Docker-контейнер для запуска приложения.

Открытые вопросы

16. Как искусственный интеллект меняет подходы к коллективной разработке?
17. Какие ошибки чаще всего допускаются при управлении проектами в команде?
18. Как коллективная разработка помогает в снижении затрат на бизнес?
19. Какие факторы следует учитывать при выборе архитектуры программной системы?
20. Как удаленная работа влияет на коллективную разработку?

2. Промежуточная аттестация

Тестовые вопросы

21. Что такое микросервисная архитектура? Где она применяется?
22. Какие методы защиты данных используются в коллективной разработке?
23. В чем заключается концепция "DevOps"?
24. Как коллективная разработка поддерживает глобализацию технологий?
25. Что такое KPI? Для чего они используются в коллективной разработке?

Практические задания

26. Разработайте систему KPI для анализа эффективности команды разработчиков.
27. Создайте план управления рисками для проекта через коллективную разработку.

28. Реализуйте систему уведомлений для команды проекта.
29. Настройте интеграцию облачного хранилища с программной системой.
30. Разработайте план миграции данных из старой системы в новую.

Ситуационные задачи

31. Компания хочет автоматизировать процесс обработки данных через коллективную разработку. Как вы будете строить модель системы?

32. У вас есть набор данных о поведении пользователей. Как вы будете их анализировать через программную систему?

33. Вам нужно разработать систему для прогнозирования рыночных трендов. Какие технологии вы выберете?

34. Вы работаете над проектом управления рисками. Как коллективная разработка может быть применена в этой задаче?

35. Компания хочет автоматизировать процесс анализа данных. Как вы будете использовать коллективную разработку для решения этой задачи?

3. Итоговый контроль

Защита курсового проекта

36. Разработка программной системы для управления проектами.
37. Проектирование системы анализа данных для принятия решений.
38. Создание системы управления складскими запасами через коллективную разработку.

39. Разработка системы рекомендаций для клиентов интернет-магазина.

40. Проектирование системы управления качеством продукции через коллективную разработку.

Экзамен

41. Теоретический блок: тестовые вопросы по основным темам курса.

42. Практический блок: выполнение задания на проектирование программной системы.

43. Анализ предоставленного кода и исправление ошибок.

Оценка портфолио

44. Сборник выполненных лабораторных работ.
45. Документация по разработанным проектам.
46. Отчеты по практическим заданиям.

4. Дополнительные виды оценочных средств

47. Устное собеседование по ключевым темам курса.

48. Анализ кода и исправление ошибок в предоставленном проекте.

49. Выполнение заданий на оптимизацию производительности программной системы.

50. Разработка прототипа системы управления процессом с использованием современных технологий.

1. Текущий контроль знаний:

- Тесты на знание основ коллективной разработки (например, методологии Agile, Scrum, Git).
- Практические задания по работе с системами контроля версий (например, создание репозитория, работа с ветками в Git).
- Написание скриптов для автоматизации процессов разработки (например, CI/CD-конвейеры).
- Лабораторные работы по настройке инструментов совместной разработки (например, Docker, Jenkins).

2. Промежуточная аттестация:

- Выполнение контрольной работы (например, разработка плана управления проектом с использованием Agile-методологий).
- Рефераты или эссе на заданные темы (например, "Роль DevOps в современной разработке").
- Мини-проект (например, разработка прототипа системы управления задачами для команды разработчиков).

3. Итоговый контроль:

- Защита курсового проекта (разработка полноценной программной системы с использованием технологий коллективной разработки).
- Экзамен в виде теста или выполнения практического задания.
- Оценка портфолио выполненных лабораторных и практических работ.

4. Дополнительные виды оценочных средств:

- Устное собеседование по ключевым темам курса.
- Анализ кода и исправление ошибок в предоставленном проекте.
- Выполнение заданий на оптимизацию производительности программной системы.