

МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»
(РУТ (МИИТ))



Рабочая программа дисциплины (модуля),
как компонент образовательной программы
специализированного высшего образования
по направлению подготовки
09.04.01 Информатика и вычислительная техника,
утвержденной первым проректором ФГАОУ ВО
РУТ (МИИТ) Тимониным В.С.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Управление жизненным циклом исследовательского проекта

Направление подготовки: 09.04.01 Информатика и вычислительная техника

Направленность (профиль): Технологии проектирования программного обеспечения

Форма обучения: Очная

Рабочая программа дисциплины (модуля) в виде
электронного документа выгружена из единой
корпоративной информационной системы управления
университетом и соответствует оригиналу

Простая электронная подпись, выданная ФГАОУ ВО РУТ
(МИИТ)

ID подписи: 5665

Подписал: заведующий кафедрой Нутович Вероника
Евгеньевна

Дата: 01.09.2026

1. Общие сведения о дисциплине (модуле).

Дисциплина ориентирована на подготовку инженеров и технических лидеров, способных выстраивать передовые процессы управления жизненным циклом и обеспечения качества корпоративного программного обеспечения в условиях высокой неопределенности исследовательских проектов. Студенты погружаются в современные парадигмы инженерии качества, эволюционной архитектуры, наблюдаемости и безопасной разработки, выходя за рамки базового тестирования к стратегическому проектированию потоков создания ценности. В рамках курса обучающиеся работают со сквозным практическим кейсом, формируя комплексный паспорт архитектуры качества продукта с использованием отечественных и открытых индустриальных стандартов.

Целью освоения дисциплины является формирование у обучающихся системных компетенций в области проектирования, внедрения и управления интегрированными процессами обеспечения качества и инженерии надежности на всех этапах жизненного цикла корпоративного программного продукта.

Для достижения поставленной цели в рамках дисциплины решается комплекс задач, направленных на формирование у обучающихся способности: анализировать и оптимизировать потоки создания ценности продуктовой команды; проектировать гибридные модели жизненного цикла и эволюционной архитектуры; внедрять стратегии превентивного тестирования и прогрессивной доставки; интегрировать практики наблюдаемости и безопасной разработки в конвейеры непрерывной интеграции; оценивать и управлять техническим долгом на основе метрик инженерной эффективности.

2. Планируемые результаты обучения по дисциплине (модулю).

Перечень формируемых результатов освоения образовательной программы (компетенций) в результате обучения по дисциплине (модулю):

ПК-4 - Способен осуществлять руководство процессом обеспечения качества разрабатываемого программного продукта для корпоративного рынка.

Обучение по дисциплине (модулю) предполагает, что по его результатам обучающийся будет:

Знать:

- концепции разделения исследовательского и производственного треков в жизненном цикле продукта;
- метрики потока команды и методы вероятностного прогнозирования релизов;
- принципы эволюционной архитектуры и фиксации решений через записи ADR;
- подходы к контрактному проектированию микросервисов и API-First дизайну;
- стратегии контрактного и мутационного тестирования для оценки качества тестов;
- спецификации поведенческой разработки для автоматизации приемочного тестирования;
- принципы разработки на основе магистральной ветки и управления функциональными флагами;
- стратегии прогрессивной доставки и канареечных релизов в корпоративном контуре;
- концепции наблюдаемости и определения индикаторов уровня обслуживания;
- механизмы балансировки скорости разработки и стабильности через бюджеты ошибок;
- методологии моделирования угроз на уровне архитектуры продукта;
- подходы к автоматизации анализа уязвимых зависимостей и поиска секретов;
- классификацию технического долга и стратегии его постепенного рефакторинга;
- факторы влияния когнитивной нагрузки команды на качество кода;
- возможности аугментации процессов жизненного цикла с помощью языковых моделей;
- методы автоматизации генерации документации из кодовой базы.

Уметь:

- разделять продуктовые гипотезы на исследовательские и производственные треки при помощи регламентов проведения технических «спайков» в условиях высокой неопределенности R&D-задач;
- рассчитывать метрики потока команды и прогнозировать сроки релиза при помощи закона Литтла и дашбордов визуализации лимитов незавершенной работы;
- проектировать контракты взаимодействия микросервисов при помощи спецификаций OpenAPI и оформлять архитектурные решения в виде записей ADR;

- разрабатывать стратегии контрактного и мутационного тестирования при помощи спецификаций Gherkin для автоматизации приемочного тестирования сложных сценариев;
- проектировать стратегию непрерывной поставки при помощи функциональных флагов и канареечных релизов в условиях изоляции тестовых данных;
- формулировать индикаторы и цели уровня обслуживания при помощи схем сбора телеметрии и расчета бюджета ошибок для ключевого пользовательского сценария;
- проводить моделирование угроз по методологии STRIDE при помощи процессов автоматической блокировки сборок с уязвимыми зависимостями;
- классифицировать накопленный технический долг при помощи матрицы квадратов и разрабатывать план постепенного рефакторинга внутри регулярных спринтов.

3. Объем дисциплины (модуля).

3.1. Общая трудоемкость дисциплины (модуля).

Общая трудоемкость дисциплины (модуля) составляет 4 з.е. (144 академических часа(ов)).

3.2. Объем дисциплины (модуля) в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении учебных занятий:

Тип учебных занятий	Количество часов	
	Всего	Семестр №1
Контактная работа при проведении учебных занятий (всего):	32	32
В том числе:		
Занятия лекционного типа	16	16
Занятия семинарского типа	16	16

3.3. Объем дисциплины (модуля) в форме самостоятельной работы обучающихся, а также в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении промежуточной аттестации составляет 112 академических часа (ов).

3.4. При обучении по индивидуальному учебному плану, в том числе при ускоренном обучении, объем дисциплины (модуля) может быть реализован

полностью в форме самостоятельной работы обучающихся, а также в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении промежуточной аттестации.

4. Содержание дисциплины (модуля).

4.1. Занятия лекционного типа.

№ п/п	Тематика лекционных занятий / краткое содержание
1	Управление жизненным циклом R&D-продукта и Dual-Track Agile Рассматриваемые вопросы: - разделение исследовательского и производственного треков (Discovery и Delivery) в продуктовом цикле; - управление техническими «спайками» и таймбоксинг научных гипотез; - критерии перехода прототипа в бэклог и механизмы fail-fast.
2	Метрики потока команды и вероятностное прогнозирование Рассматриваемые вопросы: - анализ метрик Cycle Time, Lead Time и WIP для оценки эффективности команды; - применение закона Литтла для прогнозирования сроков релиза; - выявление системных потерь и узких мест в процессе code review.
3	Эволюционная архитектура и API-First дизайн Рассматриваемые вопросы: - фиксация архитектурных решений через записи ADR (Architecture Decision Records); - контрактное проектирование микросервисов и спецификации OpenAPI; - влияние предметно-ориентированного проектирования (DDD) на качество кода.
4	Продвинутые стратегии Shift-Left и превентивного тестирования Рассматриваемые вопросы: - контрактное тестирование (Contract Testing) и предотвращение ошибок интеграции; - мутационное тестирование для оценки качества самих тестовых наборов; - поведенческая разработка (BDD) и спецификации Gherkin для сложных сценариев.
5	Инженерия релизов и стратегии прогрессивной доставки Рассматриваемые вопросы: - разработка на основе магистральной ветки (Trunk-Based Development); - управление функциональными флагами (Feature Toggles) и скрытие кода; - канареечные релизы и сине-зеленое развертывание в корпоративном контуре.
6	Наблюдаемость (Observability) и инженерия надежности сервиса Рассматриваемые вопросы: - определение индикаторов и целей уровня обслуживания (SLI/SLO); - расчет и управление бюджетом ошибок (Error Budget); - распределенная трассировка и анализ телеметрии в продакшене.
7	Моделирование угроз и интеграция DevSecOps в команду Рассматриваемые вопросы: - практическое применение методологии STRIDE для архитектуры продукта; - автоматизация анализа уязвимых зависимостей (SCA) и поиска секретов; - управление лицензионной чистотой Open Source компонентов.

№ п/п	Тематика лекционных занятий / краткое содержание
8	Экономика технического долга и AI-аугментация процессов Рассматриваемые вопросы: - классификация технического долга по матрице квадратов и оценка его стоимости; - стратегии постепенного рефакторинга внутри регулярных спринтов; - применение LLM для генерации тестов, код-ревью и автоматизации документации.

4.2. Занятия семинарского типа.

Практические занятия

№ п/п	Тематика практических занятий/краткое содержание
1	Разделение треков Discovery и Delivery Студент анализирует продуктовые гипотезы и разделяет их на исследовательские и производственные треки. Обучающийся формирует регламент проведения технических «спайков» с жестким таймбоксингом для проверки научных допущений. На основе результатов смоделированного исследования студент принимает решение о переходе прототипа в бэклог продукта. Итоговый документ с критериями валидации гипотез вшивается в паспорт жизненного цикла.
2	Оптимизация потока команды и прогнозирование Студент рассчитывает метрики потока команды – Cycle Time и Lead Time – на основе смоделированных данных системы трекинга задач. Обучающийся применяет закон Литтла для прогнозирования сроков релиза и выявления узких мест в процессе code review. Студент проектирует дашборд для визуализации лимитов незавершенной работы и эффективности потока. Схема оптимизации процесса поставки добавляется в мастер-документ паспорта проекта.
3	Архитектурные контракты и API-First спецификации Обучающийся проектирует контракт взаимодействия микросервисов разрабатываемого продукта в формате OpenAPI. Студент оформляет ключевые архитектурные решения в виде записей ADR, обосновывая выбор технологического стека. В работе моделируется процесс синхронизации спецификаций между командами фронтенда и бэкенда. Полученные артефакты интегрируются в раздел архитектурного контроля паспорта качества.
4	Стратегия контрактного и мутационного тестирования Студент разрабатывает стратегию контрактного и мутационного тестирования для критических компонентов исследовательского ПО. Обучающийся составляет спецификации в формате Gherkin для автоматизации приемочного тестирования сложных бизнес-сценариев. Студент описывает процесс интеграции этих проверок в локальные хуки и пре-коммит проверки разработчиков. Стратегия превентивного контроля вшивается в инфраструктурный раздел паспорта.
5	Инженерия безопасных релизов и Feature Toggles Обучающийся проектирует стратегию непрерывной поставки с использованием функциональных флагов и канареечных релизов. Студент разрабатывает матрицу управления конфигурациями для безопасного отката проблемных версий корпоративного продукта. В работе описывается процесс изоляции тестовых данных и управления секретами в конвейере CI/CD. Регламент безопасного релиза добавляется в соответствующий раздел мастер-документа
6	Формирование SLI/SLO и бюджета ошибок Студент формулирует индикаторы и цели уровня обслуживания (SLI/SLO) для ключевого пользовательского сценария продукта. Обучающийся проектирует схему сбора телеметрии и распределенной трассировки для быстрого выявления аномалий в продакшене. Студент рассчитывает бюджет ошибок (Error Budget) и определяет политики реагирования на его исчерпание. Документ с метриками наблюдаемости становится частью раздела эксплуатации паспорта проекта.

№ п/п	Тематика практических занятий/краткое содержание
7	Моделирование угроз продукта (STRIDE) и SCA Обучающийся проводит моделирование угроз по методологии STRIDE для архитектуры разрабатываемого корпоративного сервиса. Студент проектирует процесс автоматической блокировки сборок при обнаружении уязвимых зависимостей и жестко заданных секретов в коде. В работе формируется чек-лист проверки лицензионной чистоты Open Source библиотек перед релизом. Регламент безопасности продукта интегрируется в итоговый паспорт качества.
8	Управление техническим долгом и финализация паспорта Студент классифицирует накопленный в проекте технический долг по матрице квадратов и оценивает его влияние на скорость доставки новых функций. Обучающийся разрабатывает план постепенного рефакторинга, интегрируя задачи по погашению долга в регулярные спринты команды. Студент консолидирует все артефакты в итоговый паспорт, готовя резюме для защиты перед продуктовым стейкхолдером. Финальный пакет документов оформляется в отечественном офисном пакете.

4.3. Самостоятельная работа обучающихся.

№ п/п	Вид самостоятельной работы
1	Изучение рекомендованной литературы.
2	Подготовка к промежуточной аттестации.
3	Подготовка к текущему контролю.

5. Перечень изданий, которые рекомендуется использовать при освоении дисциплины (модуля).

№ п/п	Библиографическое описание	Место доступа
1	Управление жизненным циклом информационных систем : методические указания / составитель И. А. Цыганова. — Оренбург : ОГУ, 2024. — 63 с. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/502503 (дата обращения: 18.06.2026)
2	Аншина, М. Л. Управление жизненным циклом информационных систем : учебное пособие / М. Л. Аншина. — Москва : РТУ МИРЭА, 2024. — 169 с. — ISBN 978-5-7339-2318-5. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/448937 (дата обращения: 18.06.2026)

6. Перечень современных профессиональных баз данных и информационных справочных систем, которые могут использоваться при освоении дисциплины (модуля).

Официальная документация и спецификации промышленных стандартов API-First дизайна и контрактного тестирования.

Электронно-библиотечные системы (ЭБС) «Лань» и Znanium для доступа к академической литературе по стандартизации и инженерии требований.

Открытые базы знаний и отчеты исследовательских консорциумов по метрикам инженерной эффективности и DevOps-практикам.

Официальные порталы стандартов ЕСПД и ГОСТ для оформления проектной документации.

7. Перечень лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства, необходимого для освоения дисциплины (модуля).

Операционные системы – отечественные дистрибутивы Linux, включенные в Единый реестр российского ПО.

Офисные пакеты – российские решения для корпоративного документооборота и подготовки отчетов по ГОСТ.

Системы управления жизненным циклом (ALM) и трекинга задач – отечественные корпоративные платформы и открытые решения для расчета метрик потока.

Инструменты контрактного тестирования и спецификации API – открытые промышленные стандарты и фреймворки для автоматизации проверки интеграций.

Платформы наблюдаемости и мониторинга – вендор-независимые фреймворки сбора телеметрии и связки для визуализации временных рядов.

Средства статического и динамического анализа безопасности – российские платформы безопасной разработки и открытые сканеры уязвимостей зависимостей.

Конвейеры непрерывной интеграции – открытые системы автоматизации сборки и управления функциональными флагами.

Системы контроля версий и среды разработки – стандартные отраслевые инструменты с поддержкой локальных хуков и пре-коммит проверок.

8. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю).

Учебные аудитории для проведения учебных занятий, оснащенные компьютерной техникой и наборами демонстрационного оборудования.

Для практических занятий – наличие персональных компьютеров вычислительного класса.

9. Форма промежуточной аттестации:

Зачет в 1 семестре.

10. Оценочные материалы.

Оценочные материалы, применяемые при проведении промежуточной аттестации, разрабатываются в соответствии с локальным нормативным актом РУТ (МИИТ).

Авторы:

старший преподаватель кафедры
«Цифровые технологии управления
транспортными процессами»

И.С. Разживайкин

Согласовано:

Заведующий кафедрой ЦТУТП

В.Е. Нутович

Председатель учебно-методической
комиссии

Н.А. Андриянова