МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»
(РУТ (МИИТ)

Рабочая программа дисциплины (модуля),
как компонент образовательной программы
базового высшего образования
по направлению подготовки
09.03.01 Информатика и вычислительная техника,
утвержденной директором ФГАОУ ВО РУТ
(МИИТ) Париновым Д.В.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Инженерия качества программного обеспечения

Направление подготовки: 09.03.01 Информатика и вычислительная техника

Направленность (профиль): Технологии разработки 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: 937226
Подписал: директор центра Проневич Ольга Борисовна
Дата: 12.11.2026

1. Общие сведения о дисциплине (модуле).

Дисциплина формирует у обучающихся способность отвечать за качество собственного продуктового инкремента в промышленной разработке бизнес-приложений. Обучение строится на учебном контуре индустриального партнёра (модуль расчёта штрафов и премий с ролевым согласованием операций на Java, React и TypeScript с базой данных PostgreSQL) в формате сквозного практического кейса, где результат каждого занятия становится входными данными следующего. Студент в роли младшего инженера по качеству выделяет риски инкремента, разрабатывает контрольные примеры и тестовые наборы данных, автоматизирует многоуровневые проверки, воспроизводит и устраняет дефекты, проводит инспекцию и рефакторинг кода и обосновывает готовность версии к выпуску. Итогом является защищённое досье качества версии, доказывающее освоение результатов обучения. Содержание дисциплины соответствует профессиональному стандарту «Специалист по тестированию в области информационных технологий» и отвечает отраслевому переходу от ручного тестирования к инженерии качества на отечественном и свободно распространяемом стеке.

Целью освоения дисциплины является формирование способности реализовывать риск-ориентированный план качества программного инкремента, разрабатывать контрольные примеры и тестовые данные, выполнять многоуровневые автоматизированные проверки, воспроизводить и устранять дефекты, проводить инспекцию и рефакторинг кода, оценивать готовность версии к выпуску на основании проверяемых свидетельств.

Для достижения поставленной цели в рамках дисциплины решается комплекс задач, направленных на формирование у обучающихся способности:

выделять риски инкремента и преобразовывать их в проверяемый план качества. Разрабатывать контрольные примеры и тестовые наборы данных для функциональных, интеграционных, сквозных и нагрузочных проверок. Автоматизировать проверки серверной части на Java, клиентской части на React и TypeScript, программных интерфейсов и базы данных. Воспроизводить дефект по свидетельствам, локализовать первопричину, исправлять код и вводить регрессионную проверку. Проводить инспекцию и рефакторинг кода с учётом читаемости, производительности и возможности отказа. Анализировать результаты проверок, дефекты и показатели качества, устанавливать контрольные шлюзы выпуска. Формировать инженерную рекомендацию о готовности версии и обосновывать остаточный риск.

2. Планируемые результаты обучения по дисциплине (модулю).

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

ДПК-БП-1 - Способен обеспечивать безопасность бизнес-приложения на протяжении жизненного цикла;

ОПК-6 - Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения;

ППК-Р6 - Способен участвовать в промышленной разработке программного обеспечения.

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

Знать:

- методы создания и документирования контрольных примеров и тестовых наборов данных;
- требования к структуре и форматам хранения тестовых наборов данных;
- правила, алгоритмы и технологии создания тестовых наборов данных;
- методы и средства проверки работоспособности компьютерного программного обеспечения;
- государственные стандарты испытания автоматизированных систем;
- руководящие документы по стандартизации требований к документам автоматизированных систем;
- типичные ошибки, возникающие при разработке компьютерного программного обеспечения, методы их диагностики и исправления;
- методы и приемы отладки программного кода;
- методы и средства рефакторинга и инспекции программного кода;
- нормативно-технические документы (стандарты и регламенты), регламентирующие требования к программному коду, порядок отражения изменений в системе управления версиями, порядок отражения результатов рефакторинга, оптимизации и инспекции в коллективной базе знаний;
- о возможности отказа частей системы в результате воздействий внутреннего и внешнего нарушителя (хакер, неосторожный пользователь, программист, поставщик компонентов);
- принципы чистого кода, SOLID, DRY, KISS и др.;
- принципы предметно-ориентированного проектирования (ПОП) программного обеспечения;
- паттерны проектирования и антипаттерны.

Уметь:

- разрабатывать и оформлять контрольные примеры для проверки работоспособности компьютерного программного обеспечения;
- готовить тестовые наборы данных в соответствии с выбранной методикой тестирования компьютерного программного обеспечения;
- применять методы и средства проверки работоспособности компьютерного программного обеспечения;
- интерпретировать диагностические данные проверки работоспособности компьютерного программного обеспечения;
- анализировать значения полученных характеристик компьютерного программного обеспечения;
- воспроизводить дефекты программного кода, зафиксированных в базе данных дефектов;
- выяснять причины возникновения дефектов программного кода;
- вносить изменений в программный код для устранения выявленных дефектов;
- анализировать программный код на соответствие требованиям по читаемости и производительности;
- проводить инспекцию программного кода для поиска не обнаруженных на ранних стадиях разработки компьютерного программного обеспечения ошибок и критических мест;
- применять методы и средства рефакторинга и инспекции программного кода;
- публиковать результаты рефакторинга и инспекции в коллективной базе знаний;
- проводить инспекцию программного кода для поиска мест возможных отказов из воздействий внутреннего или внешнего нарушителя;
- разрабатывать модульный и тестируемый программный код;
- выполнять модульное, интеграционное и нагрузочное тестирование;
- проводить рефакторинг для повышения качества кода;
- применять принципы ПОП при разработке программного обеспечения на языках программирования высокого уровня абстракций и в LowCode и NoCode системах.
- проектирует меры безопасности в архитектуре, программном коде, конвейере непрерывной интеграции и поставки, а также в инфраструктуре;
- автоматизирует статический анализ кода, анализ состава зависимостей, поиск секретов, проверку контейнеров и инфраструктуры как кода с контрольными условиями качества;

- формирует ведомость состава программного обеспечения (SBOM) и правила управления зависимостями, исключениями и происхождением артефактов;

- разрабатывает риск-ориентированный план проверки и воспроизводит уязвимости;

- приоритизирует находки, устраняет первопричину и добавляет регрессионную проверку;

- документирует доказательства исправления и остаточный риск.

3. Объем дисциплины (модуля).

3.1. Общая трудоемкость дисциплины (модуля).

Общая трудоемкость дисциплины (модуля) составляет 3 з.е. (108 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 №6
Контактная работа при проведении учебных занятий (всего):	48	48
В том числе:		
Занятия лекционного типа	16	16
Занятия семинарского типа	32	32

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 60 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	Качество программного обеспечения как инженерная ответственность разработчика Рассматриваемые вопросы: - место обеспечения качества в жизненном цикле программного обеспечения и переход от роли тестировщика к инженерии качества; - модели качества по ГОСТ Р ИСО/МЭК 25010, верификация и валидация, приемо-сдаточные испытания; - государственные стандарты испытания автоматизированных систем и руководящие документы серий ГОСТ 19 и ГОСТ 34; - доказательный подход к качеству: свидетельство, воспроизводимость, остаточный риск.
2	Риск-ориентированное планирование качества инкремента Рассматриваемые вопросы: - идентификация и классификация рисков программного продукта; - матрица «риск – сценарий – уровень проверки» и распределение усилий при ограниченном времени; - состав риск-ориентированного плана качества: критичные сценарии, виды проверок, критерии входа и выхода; - связь плана качества с контрольными шлюзами промышленного конвейера разработки.
3	Проектирование контрольных примеров и тестовых данных Рассматриваемые вопросы: - контрольный пример как документированный инженерный артефакт: структура, оформление, версионирование; - методы проектирования тестовых наборов: классы эквивалентности, анализ граничных значений, таблицы решений, попарные комбинации; - требования к структуре и форматам хранения тестовых наборов данных; - генерация синтетических и обезличенных данных, запрет использования производственных данных.
4	Многоуровневая автоматизированная проверка работоспособности Рассматриваемые вопросы: - уровни проверок: модульные, интеграционные, программно-интерфейсные, сквозные; пирамида тестирования; - методы и средства автоматизации проверок для серверной части на Java, клиентской части на React и TypeScript, программных интерфейсов и базы данных; - нагрузочный эксперимент: профиль нагрузки, метрики производительности, границы ёмкости; - интерпретация диагностических данных: журналы выполнения, отчёты о покрытии, анализ характеристик.
5	Жизненный цикл дефекта и его устранение Рассматриваемые вопросы: - база данных дефектов: атрибуты карточки, классификация по критичности, статусная модель; - типичные ошибки разработки и их корневые причины, связь дефекта с этапом внесения; - методы воспроизведения и локализации первопричины: минимальный пример, изоляция, приёмы отладки кода; - минимальное исправление и регрессионная проверка: состав регрессионного пакета, подтверждение отсутствия повторного проявления.
6	Инспекция и рефакторинг программного кода Рассматриваемые вопросы: - инспекция кода: контрольные списки, статический анализ, поиск не выявленных ранее ошибок и

№ п/п	Тематика лекционных занятий / краткое содержание
	критических мест; - возможность отказа частей системы от воздействий внутреннего и внешнего нарушителя, контроль авторизации; - рефакторинг: безопасные преобразования без изменения внешнего поведения, методы и средства; - принципы масштабируемого и поддерживаемого кода: чистый код, SOLID, DRY, KISS, паттерны и антипаттерны, публикация результатов в коллективной базе знаний.
7	Метрики качества и контрольные шлюзы выпуска Рассматриваемые вопросы: - показатели качества версии: плотность дефектов, покрытие, динамика открытия и закрытия; - ограничения метрик и риск подмены цели показателем; - контрольные шлюзы качества в конвейере CI/CD, критерии готовности версии; - роли управления изменением и выпуском, свидетельства качества и их полнота.
8	Инженерные решения о выпуске в условиях неполной определённости Рассматриваемые вопросы: - выпуск версии как инженерное решение: кто допускает систему к эксплуатации и на основании каких свидетельств; - асимметрия цены ошибок: цена выпуска версии с дефектом против цены задержки выпуска исправной версии; - принятие решения при неполных данных: отношение «остановить – продолжить – выпустить с условиями», выражение уверенности через остаточный риск; - конфликт интересов разработки, эксплуатации и заказчика при решении о выпуске, механизмы эскалации разногласий; - роли и регламенты управления изменением и выпуском в промышленной организации, пределы ответственности инженера по качеству.

4.2. Занятия семинарского типа.

Практические занятия

№ п/п	Тематика практических занятий/краткое содержание
1	Анализ инкремента и выделение критичных сценариев Студент изучает предоставленный программный инкремент модуля расчёта штрафов и премий и его предметные правила. Он выделяет критичные пользовательские и фоновые сценарии расчёта, включая выбор дат, нормативы, округление и шкальные ставки. Для каждого сценария фиксируется ожидаемое поведение и цена ошибки для заказчика. Итогом становится перечень критичных сценариев инкремента.
2	Матрица рисков инкремента Студент выявляет риски качества по каждому критичному сценарию из перечня предыдущего занятия. Он оценивает вероятность и влияние рисков и размещает их в матрице «риск – сценарий – уровень проверки». На основе матрицы определяется, какие сценарии требуют углублённой проверки, а какие допускают минимальную. Результатом является согласованная матрица рисков инкремента.
3	Риск-ориентированный план качества Студент преобразует матрицу рисков в план качества инкремента с составом проверок, критериями входа и выхода. Он распределяет усилия по видам проверок с учётом ограниченного времени до выпуска версии. В плане явно фиксируется, какой риск какой проверкой закрывается. Итоговый план качества становится управляющим документом всех последующих работ.
4	Декомпозиция предметных правил в проверяемые условия Студент раскладывает правила классического и балансового методов расчёта в полный набор

№ п/п	Тематика практических занятий/краткое содержание
	проверяемых условий. Он применяет классы эквивалентности и таблицы решений, устраняя неоднозначности исходного описания. Каждое условие получает однозначную формулировку и ожидаемый результат. Полученная таблица условий является спецификацией для будущих контрольных примеров.
5	Проектирование контрольных примеров и граничных данных Студент разрабатывает контрольные примеры по таблице условий, добавляя граничные и отрицательные наборы. Он оформляет примеры по принятому шаблону с входными данными, ожидаемым результатом и обоснованием. Набор размещается в системе управления версиями в формате, пригодном для автоматизации. Итогом является версионируемый комплект контрольных примеров и тестовых данных.
6	Стратегия многоуровневых проверок и критерии покрытия Студент проектирует стратегию проверок инкремента по пирамиде тестирования, распределяя контрольные примеры между модульным, интеграционным, программно-интерфейсным и сквозным уровнями. Он выбирает инструменты проверки для серверной части, клиентской части и базы данных. Устанавливаются целевые показатели покрытия критичных сценариев. Документ стратегии завершает фазу планирования качества.
7	Автоматизация модульных и интеграционных проверок расчёта Студент переносит контрольные примеры из комплекта предыдущей фазы в автоматизированные модульные проверки расчётного ядра. Он реализует интеграционные проверки взаимодействия расчёта с базой данных. По результатам запуска анализируются расхождения и уточняется набор данных. Итогом является работающий набор автоматизированных проверок с журналом выполнения.
8	Проверка программного интерфейса и сквозного сценария Студент автоматизирует проверки программного интерфейса согласования операции по контракту сервиса. Он строит сквозную проверку сценария от клиентской формы до сохранённого результата расчёта. Выявленные расхождения между уровнями фиксируются как потенциальные дефекты. Результатом является комплект проверок API и сквозного сценария с перечнем находок.
9	Нагрузочный эксперимент фонового расчёта Студент формирует профиль нагрузки для повторного фонового расчёта по выбранному инструменту нагрузочных испытаний. Он проводит измерения на согласованных объёмах данных и фиксирует метрики производительности. По результатам определяются границы ёмкости и время выполнения массовой обработки. Итогом является нагрузочный отчёт с интерпретацией характеристик.
10	Воспроизведение и локализация намеренных дефектов Студент воспроизводит намеренные дефекты серверной и клиентской частей по свидетельствам из журналов и результатам проверок предыдущих занятий. Он локализует первопричину каждого дефекта, отделяя её от симптомов и следствий. Дефекты регистрируются в базе дефектов с полной карточкой и минимальным воспроизводящим примером. Результатом являются карточки дефектов с установленными первопричинами.
11	Исправление дефекта авторизации и регрессионная проверка Студент воспроизводит обращение к чужому объекту в роли куратора и устанавливает серверную первопричину дефекта авторизации. Он вносит минимальное исправление проверки полномочий и добавляет автоматизированную регрессионную проверку. Смежные операции согласования проверяются на отсутствие аналогичного отказа. Итогом является исправленный код с регрессионной проверкой и доказательствами устранения.
12	Инспекция и рефакторинг кода расчётного модуля Студент проводит инспекцию кода расчётного модуля по контрольному списку на читаемость, производительность и места возможных отказов. Он формулирует и обосновывает план рефакторинга, не меняющего внешнее поведение, и выполняет его под защитой регрессионных

№ п/п	Тематика практических занятий/краткое содержание
	проверок. Результаты инспекции и рефакторинга публикуются в коллективной базе знаний. Итогом являются протокол инспекции, улучшенный код и запись в базе знаний.
13	Реестр свидетельств качества версии Студент собирает в единый реестр результаты всех проверок, нагрузочного эксперимента, карточки дефектов и протоколы инспекции, полученные на предыдущих занятиях. Он проверяет полноту и непротиворечивость свидетельств архива версии. Каждое свидетельство связывается с риском из плана качества. Итогом является согласованный реестр свидетельств версии.
14	Сопоставление результатов с критериями выпуска Студент сопоставляет показатели из реестра свидетельств с критериями готовности и контрольными шлюзами выпуска. Он оценивает влияние незакрытых дефектов на критичные сценарии заказчика. По итогам сопоставления формируется предварительное решение «готов – готов с условиями – не готов». Результатом является таблица соответствия критериям и перечень невыполненных условий.
15	Отчёт о качестве и остаточный риск Студент формирует отчёт о качестве версии по принятому шаблону на основе таблицы соответствия критериям. Он формулирует остаточный риск в терминах последствий для заказчика и условия безопасного выпуска. Отчёт оформляется в офисном пакете и готовится к представлению разным адресатам. Итогом является согласованный отчёт о качестве версии.
16	Инженерная рекомендация и защита досье качества Студент формирует итоговую инженерную рекомендацию о готовности версии с явным остаточным риском и условиями. Он пересматривает решение после регрессии и пробного развёртывания, документируя изменение позиции. На защите студент обосновывает собранное досье качества перед техническими и нетехническими участниками. Итогом является защищённое досье качества версии как итоговое портфолио дисциплины.

4.3. Самостоятельная работа обучающихся.

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

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

№ п/п	Библиографическое описание	Место доступа
1	Дукельский, К. В. Управление качеством программного обеспечения : учебное пособие / К. В. Дукельский, И. Б. Бондаренко. — Санкт-Петербург : СПбГУТ им. М.А. Бонч-Бруевича, 2021. — 52 с.	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/279632 (дата обращения: 28.10.2026).

2	Игнатьев, А. В. Тестирование программного обеспечения : Учебное пособие для СПО / А. В. Игнатьев. — Санкт-Петербург : Лань, 2021. — 56 с. — ISBN 978-5-8114-8073-9.	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/183199 (дата обращения: 28.10.2026).
3	Игнатьев, А. В. Тестирование программного обеспечения / А. В. Игнатьев. — 2-е изд., стер. — Санкт-Петербург : Лань, 2022. — 56 с. — ISBN 978-5-8114-9936-6.	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/201188 (дата обращения: 28.10.2026).

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

ЭБС «Лань» – https://e.lanbook.com/ (основные и дополнительные издания дисциплины).

ГОСТ Р ИСО/МЭК 25010-2015 «Системная и программная инженерия. Требования и оценка качества систем и программных продуктов (SQuaRE). Модели качества систем и программных продуктов» – текст в Федеральном информационном фонде технических регламентов и стандартов – https://docs.cntd.ru/ (дата обращения: 17.10.2026).

Стандартный глоссарий терминов, используемых при тестировании (ISTQB, русская редакция) – https://rstqb.org/ (дата обращения: 17.10.2026).

JUnit 5 User Guide – официальная документация – https://junit.org/junit5/docs/current/user-guide/ (дата обращения: 17.10.2026).

Pytest Documentation – https://docs.pytest.org/ (дата обращения: 17.10.2026).

Playwright for Java – официальная документация – https://playwright.dev/java/ (дата обращения: 17.10.2026).

Apache JMeter User's Manual – https://jmeter.apache.org/usermanual/ (дата обращения: 17.10.2026).

Testcontainers – официальная документация – https://testcontainers.com/ (дата обращения: 17.10.2026).

Документация Test IT (российская система управления тестированием) – https://docs.testit.software/ (дата обращения: 17.10.2026).

OWASP Web Security Testing Guide – https://owasp.org/www-project-web-security-testing-guide/ (дата обращения: 17.10.2026).

Документация GitLab CI/CD – https://docs.gitlab.com/ci/ (дата обращения: 17.10.2026).

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

Операционные системы: Astra Linux / РЕД ОС / ALT Linux (в реестре российского ПО).

Офисные пакеты: Р7-Офис / МойОфис (план качества, протоколы инспекции, отчёт о качестве по ГОСТ-ориентированным шаблонам).

Платформа совместной разработки и база дефектов: GitFlic (ГК «Астра») / GitVerse (Сбер) – ветвление, пул-реквесты, трекер дефектов, CI/CD, вики-база знаний.

Модульные и интеграционные проверки: JUnit 5, AssertJ, Testcontainers (проверки с PostgreSQL в изолированных контейнерах).

Проверка программных интерфейсов: REST Assured, Postman / Hoppscotch (открытый аналог), curl.

Сквозные проверки клиентской части: Playwright (Java и TypeScript).

Нагрузочные эксперименты: Apache JMeter (основной), Gatling; демонстрация промышленных отечественных платформ Boomq и Yandex Load Testing.

Управление тестированием (TMS): Test IT (реестр ПО РФ, тариф Lite) / ТестОпс (Qameta Software).

Отчётность о проверках и покрытии: Allure Report, JaCoCo (Java), Istanbul/nyc (TypeScript).

Статический анализ и инспекция кода: SonarQube Community Edition; демонстрация промышленного отечественного уровня – Solar appScreener (реестр ПО РФ, сертификат ФСТЭК, SAST/DAST/SCA, генерация SBOM).

Проверка защищённости: OWASP ZAP (воспроизведение нарушений контроля доступа).

СУБД: PostgreSQL / Postgres Pro (в реестре ПО РФ).

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 6 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»
Института железнодорожного
транспорта И.С. Разживайкин

Согласовано:

Заведующий кафедрой ЦТУТП В.Е. Нутович

Директор ЦС IT - программ ВИШ О.Б. Проневич

Председатель учебно-методической
комиссии Н.А. Андриянова