

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



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

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**

**Технология разработки программного обеспечения**

Направление подготовки: 09.04.01 Информатика и вычислительная техника

Направленность (профиль): Геоинформационные и кадастровые автоматизированные системы

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 РУТ (МИИТ)  
ID подписи: 72156  
Подписал: заведующий кафедрой Розенберг Игорь Наумович  
Дата: 30.06.2024

## 1. Общие сведения о дисциплине (модуле).

Целями освоения учебной дисциплины «Технология разработки программного обеспечения» являются усвоение знаний, развитие профессиональных умений и навыков, необходимых в области разработки программного обеспечения (ПО), знакомство с современными направлениями программирования.

В задачи освоения дисциплины «Технология разработки программного обеспечения» входят:

1. знакомство с современными языками и системами программирования;
2. изучение процессов, моделей и стадий жизненного цикла ПО, соответствующих принятым международным стандартам;
3. овладение структурным и объектно-ориентированным подходами к проектированию ПО;
4. знакомство с CASE-средствами, поддерживающими как структурный, так и объектно-ориентированный подходы к проектированию ПО, а также промышленными технологиями проектирования ПО;
5. выработка научного подхода к практике применения теоретических знаний в области технологий разработки ПО;
6. повышение мотивации к процессу изучения учебной дисциплины и научной деятельности.

## 2. Планируемые результаты обучения по дисциплине (модулю).

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

**ОПК-1** - Способен самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте;

**ОПК-5** - Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем;

**ОПК-8** - Способен осуществлять эффективное управление разработкой программных средств и проектов.;

**УК-6** - Способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки.

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

**Знать:**

- современное состояние, проблемы и перспективы развития информационных и автоматизированных систем;
- современные средства программно-аппаратного обеспечения информационных и автоматизированных систем;
- перспективные подходы к разработке программных средств и проектов;
- методы эффективного управления разработкой программных средств и проектов;
- нормативную базу проектирования и разработки программного обеспечения.

**Уметь:**

- проводить обоснованный выбор методологии разработки программного обеспечения;
- применять современные подходы, методы и методики к разработке и модернизации программно-аппаратного обеспечения информационных и автоматизированных систем;
- осуществлять эффективное управление разработкой программных средств и проектов;
- составлять документацию на разработку и проектирование, оценивать наиболее перспективные технические решения.

**Владеть:**

- современными средствами разработки программного обеспечения;
- навыками управления разработкой проектов и программных средств;
- методами сравнительного анализа проектов;
- методами принятия технических решений.

**3. Объем дисциплины (модуля).****3.1. Общая трудоемкость дисциплины (модуля).**

Общая трудоемкость дисциплины (модуля) составляет 6 з.е. (216 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	№2
Контактная работа при проведении учебных занятий (всего):	60	32	28
В том числе:			
Занятия лекционного типа	30	16	14
Занятия семинарского типа	30	16	14

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 156 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	Эволюция и современные модели жизненного цикла ПО Рассматриваемые вопросы: Обзор от каскадных моделей (Waterfall, V-Model) к гибким (Agile, Scrum, Kanban, Lean) и DevOps; Критерии выбора модели для проекта.
2	Гибкие методологии (Agile) и фреймворк Scrum Рассматриваемые вопросы: Детальный разбор ценностей и принципов Agile; Роли, артефакты, события в Scrum; Спринт как основная единица работы.
3	Управление требованиями в гибкой и традиционной разработке Рассматриваемые вопросы: Техники выявления, анализа и спецификации требований (User Stories, Use Cases); Приоритизация (MoSCoW, RICE); Управление изменениями.
4	Оценка и планирование проектов. Метрики в разработке ПО Рассматриваемые вопросы: Методы оценки трудозатрат; Ключевые метрики; Управление рисками.

№ п/п	Тематика лекционных занятий / краткое содержание
5	<p>Архитектура ПО: стили, паттерны, принятие решений</p> <p>Рассматриваемые вопросы:</p> <p>Модульность и связность;</p> <p>Обзор ключевых стилей;</p> <p>Паттерны архитектурные и проектирования.</p>
6	<p>Современные практики разработки: CI/CD, DevOps культура</p> <p>Рассматриваемые вопросы:</p> <p>Понятие DevOps;</p> <p>Автоматизация сборки, тестирования и развертывания;</p> <p>Конвейер непрерывной интеграции и доставки (CI/CD);</p> <p>Инфраструктура как код (IaC).</p>
7	<p>Системы контроля версий: стратегии работы в команде</p> <p>Рассматриваемые вопросы:</p> <p>Углубленное изучение Git: branching strategies (GitFlow, GitHub Flow, Trunk-Based Development);</p> <p>Code Review, Pull Requests.</p>
8	<p>Инженерные практики качества: TDD, BDD, рефакторинг</p> <p>Рассматриваемые вопросы:</p> <p>Развитие через тестирование (TDD): красный-зеленый-рефакторинг;</p> <p>Поведенческо-ориентированная разработка (BDD);</p> <p>Принципы чистого кода (Clean Code).</p>
9	<p>Тестирование ПО: уровни, пирамида тестов, автоматизация</p> <p>Рассматриваемые вопросы:</p> <p>Пирамида тестов: unit, интеграционные, системные, e2e;</p> <p>Стратегия тестирования;</p> <p>Обзор инструментов (JUnit, Selenium, PyTest).</p>
10	<p>Работа в распределенных командах и инструменты collaboration</p> <p>Рассматриваемые вопросы:</p> <p>Особенности коммуникации в распределенных командах;</p> <p>Обзор инструментов (Jira, Confluence, Miro, Slack);</p> <p>Управление знаниями в проекте.</p>
11	<p>Управление качеством ПО и техническим долгом</p> <p>Рассматриваемые вопросы:</p> <p>Процессы обеспечения качества (QA, QC);</p> <p>Стратегии управления техническим долгом: выявление, оценка, приоритизация, «погашение».</p>
12	<p>Безопасность в жизненном цикле разработки (DevSecOps)</p> <p>Рассматриваемые вопросы:</p> <p>Основные уязвимости (OWASP Top 10);</p> <p>Принцип «Security by Design»;</p> <p>Интеграция проверок безопасности в CI/CD пайплайн.</p>
13	<p>Современные тренды: Low-Code, MLOps, Cloud-Native разработка</p> <p>Рассматриваемые вопросы:</p> <p>Обзор актуальных направлений;</p> <p>Разработка с низким кодом;</p> <p>Жизненный цикл ML-моделей (MLOps);</p> <p>Принципы Cloud-Native (контейнеры, оркестрация).</p>
14	<p>Эффективная презентация и документирование архитектурных решений</p> <p>Рассматриваемые вопросы:</p> <p>Принципы создания документов по архитектуре;</p> <p>Подготовка и проведение технических презентаций.</p>

№ п/п	Тематика лекционных занятий / краткое содержание
15	Финальная лекция: От магистра к инженеру/архитектору. Roadmap развития Рассматриваемые вопросы: Обзор карьерных треков (Tech Lead, Software Architect, DevOps Engineer); Рекомендации по непрерывному обучению.

## 4.2. Занятия семинарского типа.

### Лабораторные работы

№ п/п	Наименование лабораторных работ / краткое содержание
1	Анализ кейсов: Выбор модели ЖЦ для различных проектов Разбор реальных сценариев (стартап, гос. контракт, корпоративная система). Работа в группах по обоснованию выбора модели.
2	Workshop «Планирование первого спринта» На основе сформулированного продуктного видения (Product Vision) студенты формируют бэклог продукта, проводят планирование спринта, создают бэклог спринта.
3	Разработка и декомпозиция требований Практикум по написанию качественных пользовательских историй (INVEST-критерии), их декомпозиция на задачи. Создание диаграмм вариантов использования.
4	Покер планирования и анализ метрик Проведение сессии планирования покера на основе бэклога. Анализ примеров диаграмм сгорания задач (Burndown Chart), расчет и интерпретация метрик.
5	Проектирование архитектуры учебного приложения Групповая работа: выбор и обоснование архитектурного стиля для заданной предметной области. Созчение компонентной диаграммы.
6	Настройка базового пайплайна CI/CD (на примере GitLab CI/GitHub Actions) Создание простого конвейера для автоматической сборки, статического анализа кода и запуска unit-тестов для репозитория с кодом.
7	Командная работа с Git: модели ветвления и code review Имитация командной разработки: создание feature-веток, разрешение конфликтов слияния, проведение peer-to-peer code review.
8	Практикум по TDD и рефакторингу Написание кода «с нуля» по методологии TDD. Анализ «плохого» кода (code smells) и его рефакторинг для улучшения читаемости и поддержки.
9	Разработка многоуровневой стратегии тестирования Создание набора unit-тестов. Написание интеграционных тестов для REST API. Обсуждение плана e2e-тестирования для приложения.
10	Организация рабочего пространства проекта в Jira/Confluence Создание проекта в Jira, настройка workflows, бэклога, спринта. Оформление документации и решений (ADRs) в Confluence.
11	Аудит кода и оценка технического долга Использование инструментов статического анализа (SonarQube) для оценки качества кода. Составление отчета и плана по устранению критических замечаний.
12	Статический и динамический анализ безопасности кода Использование SAST- и DAST-инструментов для поиска уязвимостей в примерах приложений. Интерпретация результатов.

№ п/п	Наименование лабораторных работ / краткое содержание
13	Контейнеризация приложения с использованием Docker Создание Dockerfile для учебного приложения. Сборка и запуск контейнера. Обсуждение преимуществ и сценариев использования.
14	Подготовка итоговой презентации проекта Структурирование и оформление презентации для защиты проекта: обоснование выбора технологий, демонстрация архитектуры, процессов и результатов.
15	Защита командных проектов Презентация и защита итоговых проектов, выполненных в течение семестра. Оценка полноты применения изученных процессов, методов и инструментов.

#### 4.3. Самостоятельная работа обучающихся.

№ п/п	Вид самостоятельной работы
1	1. Проработка с конспектом лекций. 2. Изучение учебной литературы из приведенных источников 3. Изучение методических рекомендаций, проработка соответствующих разделов учебника.
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	Зубкова, Т. М. Технология разработки программного обеспечения : учебное пособие / Т. М. Зубкова. — Санкт-Петербург : Лань, 2022. — 324 с. — ISBN 978-5-8114-3842-6	Текст : электронный // Лань : электронно-библиотечная система. — URL: <a href="https://e.lanbook.com/book/206882">https://e.lanbook.com/book/206882</a> . — Режим доступа: для авториз. пользователей.
2	Дубровский, А. В. Геоинформационные системы: базы и банки пространственных данных для целей кадастра и землеустройства : учебно-методическое пособие / А. В. Дубровский. — Новосибирск : СГУГиТ, 2022. — 71 с. — ISBN 978-5-907513-59-4	Текст : электронный // Лань : электронно-библиотечная система. — URL: <a href="https://e.lanbook.com/book/317432">https://e.lanbook.com/book/317432</a> . — Режим доступа: для авториз. пользователей.
3	Ивановский, М. А. Методы и средства проектирования информационных систем и технологий : учебное пособие / М. А. Ивановский, И. А. Глазкова. — Тамбов : ТГТУ, 2024. — 130 с. — ISBN 978-5-8265-2787-0	Текст : электронный // Лань : электронно-библиотечная система. — URL: <a href="https://e.lanbook.com/book/472319">https://e.lanbook.com/book/472319</a> . — Режим доступа: для авториз. пользователей.

4	Игнатьев, А. В. Тестирование программного обеспечения : учебное пособие для вузов / А. В. Игнатьев. — 4-е изд., стер. — Санкт-Петербург : Лань, 2025. — 56 с. — ISBN 978-5-507-50858-7	Текст : электронный // Лань : электронно-библиотечная система. — URL: <a href="https://e.lanbook.com/book/481331">https://e.lanbook.com/book/481331</a> . — Режим доступа: для авториз. пользователей.
---	--	--

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

1. <http://library.miit.ru/> — электронно-библиотечная система Научно-технической библиотеки МИИТ.
2. <http://rzd.ru/> — сайт ОАО «РЖД».
3. <http://elibrary.ru/> — научно-электронная библиотека.
4. <http://www.e-heritage.ru> — Электронная Библиотека «Научное Наследие России».
5. <http://citforum.ru> — on-line библиотека свободно доступных материалов по информационным технологиям на русском языке.
6. <http://www.rushelp.com> — компьютерная документация от А - Z.
7. <http://www.emanual.ru> — сайт, посвящённый всем значимым событиям в IT-индустрии: новейшие разработки, уникальные методы и горячие новости.
8. Поисковые системы: Nigma, Yandex, Mail.

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

Компьютеры должны быть обеспечены стандартными лицензионными программными продуктами и обязательно программным продуктом Microsoft Office не ниже Microsoft Office 2010, Microsoft Visual Studio версии не ниже 2015.

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

Для проведения лекционных занятий необходима аудитория с мультимедиа аппаратурой. Для проведения практических занятий требуется аудитория, оснащенная мультимедиа аппаратурой и ПК с необходимым программным обеспечением и подключением к сети интернет.



9. Форма промежуточной аттестации:

Зачет в 1 семестре.

Экзамен во 2 семестре.

10. Оценочные материалы.

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

Авторы:

доцент, к.н. кафедры «Геодезия,  
геоинформатика и навигация»

А.С. Матвеев

Согласовано:

Заведующий кафедрой ГГН

И.Н. Розенберг

Председатель учебно-методической  
комиссии

М.Ф. Гуськова