

АННОТАЦИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

Б1.О.32 «Технологии создания и поддержки программного обеспечения»

Сокращенное наименование дисциплины: ТСиППО

Направление подготовки 09.03.03 «Прикладная информатика»

Уровень образования: бакалавриат

Объем трудоемкости: 6 з.е.

Цели изучения дисциплины определены государственным образовательным стандартом высшего образования и соотнесены с общими целями ОПОП ВО по направлению подготовки 09.03.03 «Прикладная информатика», в рамках которой преподается дисциплина.

Цель дисциплины: «Технологии создания и поддержки программного обеспечения»: формирование у студентов комплексных знаний и практических навыков в области современных технологий создания и поддержки программного обеспечения на протяжении всего жизненного цикла разработки (ЖЦР).

Задачи дисциплины:

- изучить основные модели ЖЦР ПО и их особенности;
- освоить современные методологии разработки ПО, включая Agile-подходы;
- научиться использовать системы контроля версий для совместной разработки;
- приобрести навыки тестирования программного обеспечения на различных уровнях;
- познакомиться с принципами автоматизации сборки, развертывания и мониторинга ПО (CI/CD);
- освоить базовые концепции DevOps и их применение в практике разработки.

Пререквизиты: информатика, основы программирование, операционные системы и сети, Технологии программирования.

Место дисциплины в структуре образовательной программы

Дисциплина «Технологии создания и поддержки программного обеспечения» относится к профильным дисциплинам, обеспечивающим формирование профессиональных компетенций выпускника по направлению подготовки 09.03.03 – Прикладная информатика. Является основой для изучения следующих дисциплин.

Постреквизиты: Б1.О.29 «Анализ, проектирование и разработка БД», Б1.В.05 «Облачные технологии и бэкэндразработка», Б1.В.07 «Разработка мобильных приложений», а также других дисциплин связанных с: разработкой, тестированием, поддержкой программного обеспечения, разработкой архитектур информационных систем. Углубляет знания, полученные в рамках дисциплин Б1.О.10 «Основы программирования», Б1.О.25 «Объектно-ориентированное программирование», Б1.О.27 «Разработка пользовательского WEB интерфейса», Б1.В.01 «Основы программирования на Python».

Компетенции, формируемые в результате освоения дисциплины: изучение данной учебной дисциплины направлено на формирование у обучающихся следующих компетенций:

КОД компетенции	Результат
УК-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач.
УК-2	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений.
ПК-1	Способен решать актуальные и значимые задачи прикладной информатики.
ОПК-2	Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства и использовать их при решении задач профессиональной деятельности.
ОПК-3	Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.
ОПК-4	Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью.
ОПК-8	Способен принимать участие в управлении проектами создания информационных систем на стадиях жизненного цикла.
ПК-3	Способен разрабатывать и адаптировать прикладное программное обеспечение.
ПК-5	Способен выявить естественнонаучную сущность проблем, возникающих в ходе профессиональной деятельности в области моделирования и анализа сложных естественных и искусственных систем.
ПК-6	Способен моделировать прикладные (бизнес) процессы и предметную область.
ПК-8	

Содержание дисциплины:

1. **Жизненный цикл разработки программного обеспечения (ЖЦР)**
 - Классические модели их преимущества и недостатки, области применения: Waterfall, V-model, Spiral model.
 - Итеративные и инкрементные модели: Rational Unified Process (RUP).
 - Agile методологии их принципы, роли, артефакты, церемонии: Scrum, Kanban, Extreme Programming (XP).
 - Сравнение Agile с традиционными подходами.
2. **Методологии разработки ПО**
 - Принципы SOLID и DRY.
 - Паттерны проектирования: основные типы паттернов (создающие, структурные, поведенческие).
 - Code Review – принципы проведения и инструменты.
3. **Системы контроля версий (Git)**
 - Основные понятия: репозиторий, коммит, ветка, и т.д.
 - Команды Git: init, clone, add, commit, push, pull, branch, merge, rebase.
 - Работа с удаленными репозиториями (GitHub, GitLab), создание репозитория, клонирование, отправка изменений, работа с issue tracker.
 - Стратегии ветвления: Gitflow, GitHub Flow.
4. **Тестирование программного обеспечения**
 - Уровни тестирования: модульное тестирование (unit testing), интеграционное тестирование, системное тестирование, приемочное тестирование.
 - Виды тестирования: функциональное тестирование, нефункциональное тестирование (нагрузочное, стрессовое, security).
 - Методики тестирования: black-box testing, white-box testing.
 - Инструменты автоматизированного тестирования.
5. **Автоматизация сборки и развертывания ПО (CI/CD)**
 - Принципы CI/CD.
 - Инструменты CI/CD: Jenkins, Travis CI, CircleCI, GitLab CI.
 - Настройка пайплайнов CI/CD для автоматической сборки, тестирования и развертывания приложений.
6. **Основы DevOps**
 - Культура DevOps: Collaboration, Automation, Measurement, Sharing (CAMS).
 - Инфраструктура как код (IaC).
 - Мониторинг и логирование.
7. **Контейнеризация (Docker)**
 - Основные понятия: образы, контейнеры, Dockerfile.
 - Создание и запуск контейнеров.
 - Docker Compose для управления многоконтейнерными приложениями.
 - Разработка Dockerfile для простого приложения.
8. **Оркестрация контейнеров (Kubernetes)**
 - Основные понятия: поды, сервисы, деплойменты, namespace.
 - Развертывание приложений в Kubernetes.
 - Масштабирование и обновление приложений в Kubernetes.
 - Практическое развертывание простого приложения в Kubernetes.

Формы организации обучения: лекции: 50 часов, лабораторные работы: 68 часов, самостоятельная работа: 97,6 часов. Всего: 216 часов

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.

№	Наименование разделов (тем)	Количество часов				
		Всего	Аудиторная Работа			Внеаудиторная работа
			Л	ПЗ	ЛР	
1.	Жизненный цикл разработки программного обеспечения (ЖЦР)	24,2	4		8	12,2
2.	Методологии разработки ПО	28,2	8		8	12,2
3.	Системы контроля версий (Git)	26,2	6		8	12,2
4.	Тестирование программного обеспечения	26,2	6		8	12,2
5.	Автоматизация сборки и развертывания ПО (CI/CD)	26,2	6		8	12,2
6.	Основы DevOps	24,2	4		8	12,2
7.	Контейнеризация (Docker)	30,2	8		10	12,2
8.	Оркестрация контейнеров (Kubernetes)	30,2	8		10	12,2
	<i>ИТОГО по разделам дисциплины</i>	215,6	50		68	97,6
	Промежуточная аттестация (ИКР)	0,4				
	Подготовка к текущему контролю					
	Общая трудоемкость по дисциплине	216				

Курсовые работы: *не предусмотрены.*

Оценочные средства: выполнение лабораторных работ с защитой. Тесты с теоретическими вопросами по темам. Практическое задание: мини-проект с использованием изученных технологий.

Критерии оценивания – полнота и правильность ответов на вопросы, качество выполненного практических заданий, соблюдение требований к оформлению отчетности.

Учебно-методическое обеспечение: презентации лекций. Методические указания к лабораторным работам. Электронные образовательные ресурсы (доступ к онлайн-курсам и документации по используемым технологиям). Примеры кода и конфигурационные файлы для практических заданий.

Форма проведения аттестации по дисциплине: *зачет.*

Автор: Вишняков Р.Ю. – канд., техн., наук, доцент каф. КАДИИ.