Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 30.09.2023 14:25:18
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение
высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
Факультет информационных технологий

УТВЕРЖДЕНО
Декан факультета
Информационных технологий
_____ / Д.Г. Демидов /
«16» 02 2023 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Системы управления разработкой программного обеспечения

Направление подготовки/специальность
09.03.02 Информационные системы и технологии

Профиль/специализация
Программное обеспечение игровой компьютерной индустрии

Квалификация
Бакалавр

Формы обучения
Очная

Москва, 2023 г.

Разработчик(и):

ст. преподаватель  / М.В. Алпатова /

Согласовано:
Заведующий кафедрой
«Информатики и информационных технологий»,
к.т.н.  / Е.В. Булатников/

Содержание

1 Цели, задачи и планируемые результаты обучения по дисциплине 4
2 Место дисциплины в структуре образовательной программы 5
3 Структура и содержание дисциплины 5
3.1 Виды учебной работы и трудоемкость 5
3.2 Тематический план изучения дисциплины 6
3.3 Содержание дисциплины 6
3.4 Тематика семинарских/практических и лабораторных занятий 7
4 Учебно-методическое и информационное обеспечение 8
4.1 Основная литература 8
4.2 Дополнительная литература 9
4.3 Электронные образовательные ресурсы 9
4.4 Лицензионное и свободно распространяемое программное обеспечение 9
5 Материально-техническое обеспечение 9
6 Методические рекомендации 9
6.1 Методические рекомендации для преподавателя по организации обучения 9
6.2 Методические указания для обучающихся по освоению дисциплины 10
7 Фонд оценочных средств 10
7.1 Методы контроля и оценивания результатов обучения 10
7.2 Шкала и критерии оценивания результатов обучения 10
7.3 Оценочные средства 11

1 Цели, задачи и планируемые результаты обучения по дисциплине

Целью изучения дисциплины "Системы управления разработкой программного обеспечения" является формирование у студентов глубоких знаний и практических навыков в области современных систем управления версиями, в том числе с использованием Git. Это направлено на обеспечение студентов способностью эффективно применять системы контроля версий в профессиональной деятельности, проектировать программное обеспечение с учетом требований к версионированию и интеграции.

В рамках дисциплины ставятся следующие задачи, соответствующие задачам профессиональной деятельности, указанным в ФГОС ВО:

- Освоение основных понятий и принципов работы систем управления версиями, в частности, Git.
- Приобретение навыков настройки и работы с локальными и удаленными репозиториями.
- Освоение методик и инструментов для эффективного ветвления, слияния и решения конфликтов в коде.
- Формирование навыков применения продвинутых методов работы в Git, таких как интерактивное перебазирование, выборочное применение изменений и тегирование релизов.
- Изучение и практическое применение стандартов и лучших практик работы в системах управления версиями.

По завершении изучения дисциплины студент:

- Знает способы разработки требований и проектирования программного обеспечения
- Обладает навыками проектирования программного обеспечения с использованием современных инструментальных средств, особенно в контексте систем управления версиями
- Знает методы интеграции программных модулей
- Способен проводить верификацию версий используемых ИТ
- Применяет программное обеспечение для верификации версий в информационных системах

Обучение по дисциплине «Системы управления разработкой программного обеспечения» направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенций	Индикаторы достижения компетенции
ПК-1. Способен разрабатывать требования и проектировать программное обеспечение.	ИПК-1.1. Знает способы разработки требований и проектирования программного обеспечения для обработки и автоматизации информации и систем управления ИПК-1.2. Умеет проектировать программное обеспечение и системы управления с применением современных инструментальных средств ИПК-1.3. Имеет навыки разработки требований и проектирования информационных систем управления и автоматизированных систем обработки информационных потоков и систем

ПК-4. Способен проводить интеграцию программных модулей и компонент и верификацию выпусков программного продукта.	ИПК-4.1. Знает методы и способы интеграции программных модулей информационных и автоматизированных систем по обработке информации ИПК-4.2. Умеет проводить верификацию выпусков информационных и автоматизированных систем по обработке информации и их систем управления ИПК-4.3. Имеет навыки применения программного обеспечения для верификации версий используемых ИТ в информационных и автоматизированных систем по обработке информации и их систем управления в виде готовых продуктов

2 Место дисциплины в структуре образовательной программы

Дисциплина относится к обязательной части блока Б1 «Дисциплины (модули)».

Дисциплина взаимосвязана логически и содержательно-методически со следующими дисциплинами и практиками ОПОП:

- Операционные системы
- Базы данных
- Системы управления разработкой программного обеспечения
- Информационная безопасность и защита информации
- Введение в программирование
- Технологии прикладного программирования
- Объектно-ориентированное программирование
- Веб-программирование и дизайн
- Проектная деятельность
- Управление проектами
- Методология DevOps
- Тестирование программного обеспечения
- Шаблоны проектирования
- BackEnd-разработка
- Рефакторинг
- Разработка технической документации

3 Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 2 зачетные единицы, т.е. 72 академических часа (из них 36 часов – аудиторные занятия и 36 часов – самостоятельная работа студентов).

Разделы дисциплины изучаются на 2 курсе в 3 семестре, форма промежуточной аттестации – зачет.

3.1 Виды учебной работы и трудоемкость (по формам обучения)

3.1.1 Очная форма обучения

	Вид учебной работы		**Семестры**

№ п/п		Количество часов	Семестр	Неделя семестра
1	**Аудиторные занятия**	**36**	3	1-8
	В том числе:			
1.1	Лекции	18		
1.2	Семинарские/практические занятия	-		
1.3	Лабораторные занятия	18		
2	**Самостоятельная работа**	**36**	3	1-8
3	**Промежуточная аттестация**		3	19-21
	Экзамен			
	Итого:	**72**		

3.2 Тематический план изучения дисциплины (по формам обучения)

3.2.1 Очная форма обучения

№ п/п	Разделы/темы дисциплины	Трудоемкость, час					
		Всего	Аудиторная работа				Самостоятельная работа
			Лекции	Семинарские/ практические занятия	Лабораторные занятия	Практическая подготовка	
1	Введение в системы контроля версий (VCS)		2		2		4
2	Начало работы с Git		2		2		4
3	Подробно: Внутреннее устройство Git		2		2		4
4	Ветвление и слияние в Git		2		2		4
5	Разрешение конфликтов и перебазирование		2		2		4
6	Удаленные репозитории и коллаборация в Git		2		2		4
7	Продвинутые методы работы с Git		2		2		4
8	Метки, релизы и хуки в Git		2		2		4
9	Лучшие практики и реальные сценарии использования		2		2		4

3.3 Содержание дисциплины

1. Введение в системы контроля версий (VCS):

Обзор принципов учета изменений в коде, исторический контекст появления VCS, основное различие между централизованными и децентрализованными системами, краткий обзор популярных инструментов.

2. Начало работы с Git:

Процесс установки Git на различных ОС, первоначальная настройка и конфигурация, основные команды для работы с репозиториями и их коммитами.

3. Внутреннее устройство Git:

Глубокое погружение в то, как Git хранит объекты и изменения, структура каталога .git, принципы создания и связывания объектов в Git.

4. Ветвление и слияние в Git:

Понимание концепции ветвления, методы создания, переключения и слияния веток, основные стратегии слияния и их применение.

5. Разрешение конфликтов и перебазирование:

Причины возникновения конфликтов при слиянии, методы и инструменты их разрешения, понимание и практика перебазирования, его преимущества и риски.

6. Удаленные репозитории и коллабораиця в Git:

Процесс настройки удаленных репозиториев и взаимодействие с ними, основы работы в команде, популярные сценарии и рабочие процессы в Git.

7. Продвинутые методы работы с Git:

Глубокое изучение продвинутых возможностей Git, таких как интерактивное перебазирование, выборочное применение изменений и поиск дефектов с помощью bisect.

8. Метки, релизы и хуки в Git:

Значимость и методы работы с метками, создание и управление релизами, автоматизация рабочих процессов с помощью хуков в Git.

9. Лучшие практики и реальные сценарии использования:

Стандарты качества коммитов, рекомендации по работе с большими репозиториями и файлами, интеграция Git в процессы непрерывной интеграции и доставки.

3.4 Тематика семинарских/практических и лабораторных занятий

Установка и первый коммит:

Цель: Ознакомление с процессом установки Git и созданием первого репозитория.

Задания: Установите Git, создайте свой первый репозиторий, сделайте первые изменения и зафиксируйте их с помощью коммита. Посмотрите историю коммитов.

Рекомендации: Проверьте корректность установки Git, обратите внимание на конфигурацию автора коммитов.

Изучение внутренностей Git:

Цель: Глубокое погружение во внутреннее устройство Git.

Задания: Исследуйте содержимое каталога .git, создайте объекты вручную (blobs, trees) и изучите, как они связаны.

Рекомендации: Используйте команды низкого уровня Git для работы с объектами.

Ветвление:

Цель: Понимание и применение концепции ветвления в Git.

Задания: Создайте несколько веток, переключайтесь между ними, делайте параллельные изменения для наблюдения разветвления.

Рекомендации: После создания веток, проверьте историю коммитов, чтобы увидеть различия.

Слияние и разрешение конфликтов:

Цель: Научиться сливать ветки и решать возникающие при этом конфликты.

Задания: Слейте ветки с конфликтующими изменениями, решите возникающие конфликты вручную.

Рекомендации: Используйте визуальные инструменты для разрешения конфликтов, если это необходимо.

Взаимодействие с удаленными репозиториями:

Цель: Овладеть базовыми навыками работы с удаленными репозиториями.

Задания: Создайте аккаунт на GitHub (или другой платформе), отправьте (push) свой репозиторий на удаленный сервер и сотрудничайте с коллегой.

Рекомендации: Познакомьтесь с понятиями fork, pull request и collaboration в контексте работы с платформами вроде GitHub.

Перебазирование и продвинутое слияние:

Цель: Глубокое изучение перебазирования и сложных сценариев слияния.

Задания: Возьмите ветку и перебазируйте ее на master, разрешите возникающие конфликты.

Рекомендации: Обратите внимание на различия между слиянием и перебазированием и на случаи их применения.

Продвинутые операции:

Цель: Изучение дополнительных инструментов и возможностей Git.

Задания: Используйте bisect для поиска коммита с ошибкой, перенесите коммит из одной ветки в другую с помощью cherry-pick.

Рекомендации: Процесс bisect может быть автоматизирован с помощью тестов.

Метки и релизы:

Цель: Освоение принципов версионирования и управления релизами с использованием Git.

Задания: Создайте метки для конкретных коммитов, отражающих версии или важные этапы; создайте релиз на платформе, такой как GitHub.

Рекомендации: Изучите принципы семантического версионирования для правильного присвоения версий вашим релизам.

4 Учебно-методическое и информационное обеспечение

4.1 Основная литература

1. Влацкая, И. В. Проектирование и реализация прикладного программного обеспечения : учебное пособие / И. В. Влацкая, Н. А. Заельская, Н. С. Надточий. — Оренбург : Оренбургский государственный университет, ЭБС АСВ, 2015. — 119 с. — ISBN 978-5-7410-1238-3. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/54145.html (дата обращения: 22.08.2023). — Режим доступа: для авторизир. пользователей

2. Долженко, А. И. Технологии командной разработки программного обеспечения информационных систем : курс лекций / А. И. Долженко. — 3-е изд. — Москва : Интернет-

Университет Информационных Технологий (ИНТУИТ), Ай Пи Эр Медиа, 2019. — 300 с. — ISBN 978-5-4486-0525-3. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/79723.html (дата обращения: 22.08.2023). — Режим доступа: для авторизир. пользователей

3. Зубкова, Т. М. Технология разработки программного обеспечения : учебное пособие для СПО / Т. М. Зубкова. — Саратов : Профобразование, 2019. — 468 с. — ISBN 978-5-4488-0354-3. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/86208.html (дата обращения: 22.08.2023). — Режим доступа: для авторизир. пользователей

4.2 Дополнительная литература

1. Git tutorial URL: https://git-scm.com/
2. Learn Git Branching URL: http://learngitbranching.js.org/
3. Try Git URL: https://try.github.io/levels/1/challenges/1
4. Git wiki URL: https://git.wiki.kernel.org/index.php/Main_Page

4.3 Электронные образовательные ресурсы

1. Системы управления разработкой программного обеспечения. LMS Московского политеха. URL: https://online.mospolytech.ru/course/view.php?id=12723

4.4 Лицензионное и свободно распространяемое программное обеспечение

1. VS Code
2. SourceTree

5 Материально-техническое обеспечение

Лабораторные работы и самостоятельная работа студентов должны проводиться в специализированной аудитории, оснащенной современной оргтехникой и персональными компьютерами с программным обеспечением в соответствии с тематикой изучаемого материала. Число рабочих мест в аудитории должно быть достаточным для обеспечения индивидуальной работы студентов. Рабочее место преподавателя должно быть оснащены современным компьютером с подключенным к нему проектором на настенный экран, или иным аналогичным по функциональному назначению оборудованием.

6 Методические рекомендации

6.1 Методические рекомендации для преподавателя по организации обучения

1. При подготовке к занятиям следует предварительно проработать материал занятия, предусмотрев его подачу точно в отведенное для этого время занятия. Следует подготовить необходимые материалы – теоретические сведения, задачи и др. При проведении занятия следует контролировать подачу материала и решение заданий с учетом учебного времени, отведенного для занятия.

2. При проверке работ и отчетов следует учитывать не только правильность выполнения заданий, но и оптимальность выбранных методов решения, правильность выполнения всех его шагов.

6.2 Методические указания для обучающихся по освоению дисциплины

В процессе самостоятельной работы студенты закрепляют и углубляют знания, полученные во время аудиторных занятий, дорабатывают конспекты и записи, готовятся к промежуточной аттестации, а также самостоятельно изучают отдельные темы учебной программы.

На занятиях студентов, в том числе предполагающих практическую деятельность, осуществляется закрепление полученных, в том числе и в процессе самостоятельной работы, знаний. Особое внимание обращается на развитие умений и навыков установления связи положений теории с профессиональной деятельностью будущего специалиста.

Самостоятельная работа осуществляется индивидуально. Контроль самостоятельной работы организуется в двух формах:

- самоконтроль и самооценка студента;
- контроль со стороны преподавателей (текущий и промежуточный).

Текущий контроль осуществляется на аудиторных занятиях, промежуточный контроль осуществляется на зачете в письменной (устной) форме.

Критериями оценки результатов самостоятельной работы студента являются:

- уровень освоения студентом учебного материала;
- умения студента использовать теоретические знания при выполнении практических задач;
- сформированность компетенций;
- оформление материала в соответствии с требованиями.

7 Фонд оценочных средств

7.1 Методы контроля и оценивания результатов обучения

В процессе обучения используются следующие оценочные формы самостоятельной работы студентов, оценочные средства текущего контроля успеваемости и промежуточных аттестаций:

- Выполнение лабораторных работ
- Промежуточное тестирование (посредством изучения теоретических материалов в системе LMS)
- Итоговое тестирование

7.2 Шкала и критерии оценивания результатов обучения

Итоговая оценка по дисциплине рассчитывается как среднее взвешенное всех оценок в соответствующем курсе LMS Московского политеха с применением весовых коэффициентов, представленных ниже:

- Ознакомление с теорией → 0.05
- Прохождение интерактива → 0.15
- Лабораторные работы → 0.5
- Тестирование → 0.3 (0.7 * Итоговое тестирование, 0.3 * среднее по промежуточным)

Оценка за каждую лабораторную работу выставляется исходя из фактического выполнения всех поставленных задач с учётом сроков исполнения: за каждую 1 неделю просрочки задания из оценки вычитается 10 баллов.

Для получения положительной экзаменационной оценки студенту необходимо набрать всего минимально 55 баллов по дисциплине и завершить итоговый тест с результатом не менее 55%.

Шкала оценивания	Диапазон баллов	Описание
Неудовлетворительно	0-54	Не достигнуто пороговое значение хотя бы для одного уровня формируемых на момент проведения аттестации компетенций. Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателей, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.
Удовлетворительно	55-69	Среднее значение для всех формируемых на момент проведения аттестации уровней компетенций – 3. Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, в котором освещена основная, наиболее важная часть материала, но при этом допущена одна значительная ошибка или неточность.
Хорошо	70-84	Среднее значение для всех формируемых на момент проведения аттестации уровней компетенций – 4. Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует неполное, правильное соответствие знаний, умений, навыков приведенным в таблицах показателей, либо если при этом были допущены 2-3 несущественные ошибки.
Отлично	85-100	Среднее значение для всех формируемых на момент проведения аттестации уровней компетенций – 5. Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателей, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.

7.3 Оценочные средства

7.3.1 Вопросы к зачету

1. Что такое система контроля версий (Version Control System, VCS) и для чего она необходима?
2. В чем основные различия между централизованными и распределенными системами контроля версий?

3. Какие команды Git используются для базовых операций: инициализации, добавления изменений и фиксации (коммита) этих изменений?
4. Что представляют из себя объекты в Git (blobs, trees, commits, и tags) и как они связаны между собой?
5. Какие стратегии слияния существуют в Git (fast-forward, three-way merge)?
6. В чем заключается процесс перебазирования (rebase) и как он отличается от процесса слияния (merge)?
7. Какие проблемы могут возникнуть при слиянии веток и как их разрешить?
8. В чем преимущества и недостатки работы с удаленными репозиториями, такими как GitHub, GitLab, Bitbucket?
9. Какие подходы к сотрудничеству существуют в Git (feature branching, GitFlow и др.)?
10. В чем заключается интерактивное перебазирование и какие сценарии его применения вы знаете?
11. Что такое cherry-picking и в каких случаях он может быть полезен?
12. Зачем нужно использовать тегирование в Git и какова роль семантического версионирования?
13. Что такое хуки в Git (hooks) и какие типы хуков вы можете назвать?
14. Какие стандарты записи сообщений коммитов существуют и почему они важны?
15. В чем особенности работы с большими репозиториями и большими файлами в Git?
16. Как интегрировать Git с системами непрерывной интеграции/непрерывной доставки (CI/CD)?
17. Что такое bisect в Git и каков его основной сценарий использования?
18. Как создать и управлять ветками в Git?
19. Что такое удаленные репозитории и какова их роль в процессе сотрудничества?
20. Как правильно настроить и конфигурировать Git для оптимальной работы в команде?