Федеральное государственное бюджетное учреждение науки
Институт машиноведения им. А.А. Благонравова
Российской академии наук
(ИМАШ РАН)

Комплект оценочных материалов
по дисциплине «Управление производственным процессом разработки»

Направление подготовки:	02.04.03 Математическое обеспечение и администрирование информационных систем
Профиль подготовки:	Искусственный интеллект и интеллектуальная обработка данных
Квалификация:	Магистр
Форма обучения	Очная

Москва

Содержание

1. Общие положения....3
2. Результаты освоения дисциплины, подлежащие проверке....3
3. Контроль и оценка освоения учебной дисциплины....4
4. Оценочные материалы для проведения текущего контроля успеваемости....4
5. Оценочные материалы для проведения промежуточной аттестации....6

1. Общие положения

Комплект оценочных материалов (далее - ОМ) предназначен для контроля и оценки образовательных достижений обучающихся, освоивших программу дисциплины «Управление производственным процессом разработки».

ОМ включают контрольные материалы для проведения текущего контроля успеваемости и промежуточной аттестации в форме зачета.

ОМ разработаны на основании:

- основной профессиональной образовательной программы по направлению подготовки 02.04.03 Математическое обеспечение и администрирование информационных систем;
- рабочей программы дисциплины «Управление производственным процессом разработки».

2. Результаты освоения дисциплины, подлежащие проверке

В результате освоения дисциплины обучающийся должен обладать следующими компетенциями

ПК-2	**Управление работами системных аналитиков**
ПК-2.1	Разработка и внедрение новых методов и технологий исследования больших данных
ПК-2.2	Контроль и координация работ, выполняемых подчиненными системными аналитиками

Код индикатора компетенции	Планируемые результаты обучения по дисциплине
ПК-2.1	*Знать:* - принципы организации исследовательского процесса (R&D) в рамках стандартных производственных циклов разработки; - современные подходы к версионированию данных и экспериментов (DVC, MLflow). - метрики оценки качества и стабильности моделей машинного обучения в промышленной эксплуатации. *Уметь:* - применять методологии управления проектами для эффективного внедрения новых алгоритмов и технологий. - интегрировать новые методы исследования данных в существующий конвейер обработки (data pipeline). - аргументированно выбирать технологии для решения нестандартных аналитических задач. *Иметь практический опыт:* - участия в полном цикле разработки и внедрения нового компонента на основе машинного обучения: от постановки задачи до вывода в продакшн и последующего мониторинга.
ПК-2.2	*Знать:* - основы распределения задач и делегирования полномочий в проектной команде; - принципы и стандарты документирования требований и архитектурных решений; - ключевые метрики производительности команды (скорость, пропускная способность) и методы их отслеживания. *Уметь:* - проводить оценку и приемку выполненных работ у подчиненных аналитиков; - организовывать и фасилитировать рабочие встречи (ежедневные стендапы, планирование, ретроспективы);

	- выявлять блокирующие факторы и риски на ранних стадиях и координировать их устранение. *Иметь практический опыт:* - координации работы небольшой группы аналитиков или разработчиков в рамках учебного проекта; - анализа хода выполнения проекта, выявления отклонений от плана и составления отчетности о статусе работ.

3. Контроль и оценка освоения учебной дисциплины

Элемент учебной дисциплины	Текущий контроль		Промежуточная аттестация	
	Форма контроля	Проверяемые идк	Форма контроля	Проверяемые идк
Тема 1. Введение в управление разработкой ПО	Отчет по практикуму по решению задач	ПК-2.1 ПК-2.2		
Тема 2. Планирование и оценка проектов. Управление требованиями и бэклогом	Отчет по практикуму по решению задач	ПК-2.1 ПК-2.2		
Тема 3. Организация рабочего процесса и контроль качества.	Отчет по практикуму по решению задач	ПК-2.1 ПК-2.2		
Тема 4. Непрерывная интеграция и доставка (CI/CD). Мониторинг и эксплуатация ML-моделей (MLOps)	Отчет по практикуму по решению задач	ПК-2.1		
			Зачет	ПК-2.1 ПК-2.2

4. Оценочные материалы для проведения текущего контроля успеваемости

Показатели и критерии оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Практикум по решению задач
25-21 – работа и отчет выполнены в срок, самостоятельно, правильно определены соответствующие спецификации, использована требуемая информация, правильно выполнены требуемые расчеты, сделаны необходимые выводы, хорошо аргументированы, даны исчерпывающие ответы на все поставленные вопросы; 20-15– работа и отчет выполнены в срок, самостоятельно, правильно определены соответствующие спецификации, использована требуемая информация, правильно выполнены требуемые расчеты, необходимые выводы сделаны частично, хорошо аргументированы, даны ответы на все поставленные вопросы; 14-8 – работа и отчет выполнены в срок, в основном самостоятельно, выводы сделаны частично, слабо аргументированы, даны ответы не на все вопросы; 7-1 – обучающийся подготовил работу и отчет несамостоятельно и/или не завершил в срок, описание содержит ошибки, выводы и ответы на вопросы отсутствуют.

Порядок начисления баллов

Если студент набрал достаточное количество баллов ТКУ по дисциплине (более 70 баллов) в течение семестра, он может отказаться от сдачи зачета, и результат будет выставлен автоматически по его согласию.

Типовые контрольные задания и иные материалы, необходимые для процедуры оценивания знаний, умений, навыков и(или) опыта деятельности

Тема 1. Введение в управление разработкой ПО

Задача 1. Вы — команда, которой поручено разработать систему предиктивной аналитики для крупного ритейлера. Бизнес-требования (BRD) получены, но они содержат много неопределенностей: требования к точности модели могут меняться, источники данных могут быть нестабильны, а дедлайн — через 4 месяца.

Сравните применимость Waterfall, Scrum и Kanban к данному проекту.

Выберите одну методологию (или гибридный подход), которая, на ваш взгляд, является оптимальной.

Напишите краткий отчет, аргументируя, почему выбранная методология лучше других справится с рисками проекта (изменение требований, R&D-компонент, необходимость быстрой обратной связи).

Возьмите одно бизнес-требование из BRD: «Система должна прогнозировать спрос на товары на 30 дней вперед с точностью не менее 90%».

Декомпозируйте это требование на 3-4 пользовательские истории (User Stories), соответствующие критериям INVEST.

Для одной из User Stories (например, «Как аналитик, я хочу видеть прогноз в виде графика по категориям товаров») декомпозируйте ее на 3-4 технические задачи для команды (ML-инженер, Backend-разработчик, DevOps).

Тема 2. Планирование и оценка проектов. Управление требованиями и бэклогом

Задача 1. Планирование спринта и оценка задач

Используя технические задачи, полученные в Практикуме 1, сформируйте бэклог спринта.

Сессия Planning Poker:Распределите роли в команде (Scrum Master, ML-инженер, Backend-разработчик).

Проведите симуляцию сессии Planning Poker для оценки набора из 5-7 технических задач.

Каждый участник тайно назначает оценку в Story Points (используя последовательность Фибоначчи).

Обсудите расхождения в оценках (особенно если разница между максимальной и минимальной оценкой> 3 пунктов) и придите к консенсусу.

Определите вместимость команды на спринт (например, 20 Story Points).

Выберите задачи из бэклога продукта, которые команда обязуется выполнить в этом спринте, не превышая лимит вместимости.

Зафиксируйте итоговый бэклог спринта.

Тема 3. Организация рабочего процесса и контроль качества

Задача 1. Имитация процесса код-ревью

Вам предоставляется фрагмент кода Python для модели машинного обучения (например, скрипт обучения и сохранения модели), содержащий преднамеренные ошибки и недочеты.

Проведите ревью предоставленного кода. Используйте чек-лист для проверки:

Есть ли ошибки в логике предобработки данных или обучения модели?

Соответствует ли код стандартам PEP8? (Используйте линтер flake8 или pylint для проверки).

Есть ли неоптимальные конструкции (например, вложенные циклы там, где можно применить векторизацию)?

Есть ли docstring у функций и классов?

Создайте Pull Request и напишите к нему как минимум 5 комментариев с указанием найденных проблем и предложениями по их исправлению.

Тема 4. Непрерывная интеграция и доставка (CI/CD). Мониторинг и эксплуатация ML-моделей (MLOps)

Задача 1. Создание CI/CD-пайплайна

У вас есть локальный репозиторий Python-проекта с кодом из Практикума 3.

Создайте файл конфигурации CI/CD-пайплайна (.gitlab-ci.yml или Jenkinsfile).

Настройте пайплайн так, чтобы при каждом коммите в ветку main или develop автоматически выполнялись следующие этапы:Install: Установка зависимостей из requirements.txt.

Lint: Проверка стиля кода с помощью flake8.

Test: Запуск модульных тестов (pytest).

Build: Сборка проекта (если применимо).

Опишите, какие ключевые метрики вы бы настроили для мониторинга REST API вашей модели в продакшене.

Укажите инструменты (например, Prometheus + Grafana) и конкретные метрики (время ответа, количество ошибок, время работы предсказания), которые помогут отслеживать работоспособность и деградацию модели.

5. Оценочные материалы для проведения промежуточной аттестации

Промежуточная аттестация по дисциплине проводится в форме зачета.

Зачет представляет собой выполнение обучающимся заданий билета, включающего в себя:

Задание №1 – теоретический вопрос на знание базовых понятий предметной области дисциплины, а также позволяющий оценить степень владения обучающегося принципами предметной области дисциплины, понимание их особенностей и взаимосвязи между ними.

Задание №2 – задание на анализ ситуации из предметной области дисциплины и выявление способности обучающегося выбирать и применять соответствующие принципы и методы решения практических проблем, близких к профессиональной деятельности.

Задания №3 – задания на проверку умений и навыков, полученных в результате освоения дисциплины.

Выполнение обучающимся заданий билета оценивается по следующей балльной шкале:

Задание 1: 0-30 баллов
Задание 2: 0-30 баллов
Задание 3: 0-40 баллов

«Зачтено»

– 90-100 – ответ правильный, логически выстроен, использована профессиональная терминология. Задания решены правильно. Обучающийся правильно интерпретирует полученный результат.

– 70 -89 – ответ в целом правильный, логически выстроен, использована профессиональная терминология. Ход решения заданий правильный, ответ неверный. Обучающийся в целом правильно интерпретирует полученный результат.

– 50 - 69 – ответ в основном правильный, логически выстроен, использована профессиональная терминология. Задание решено частично.

«Не зачтено»

– менее 50 – ответы на теоретическую часть неправильные или неполные. Задания не решены.

Типовые задания для проведения промежуточной аттестации обучающихся

Задания 1 типа

1. Дайте определение жизненного цикла проекта. Опишите основные этапы классического жизненного цикла и их последовательность.

2. Сравните каскадную модель (Waterfall) и гибкие методологии (Agile). В каких случаях применение Waterfall является оправданным, а в каких — рискованным?

3. Объясните ключевые принципы Agile-манифеста. Как эти принципы помогают адаптироваться к изменениям в ходе проекта?

4. В чем заключается разница между Scrum и Kanban? Опишите основные артефакты, роли и события в Scrum, а также принципы работы с потоком задач в Kanban.

5. Перечислите основные роли в кросс-функциональной команде разработки ПО (Product Owner, Scrum Master, ML-инженер, аналитик, DevOps) и кратко опишите зону ответственности каждой из них.

6. Каковы основные особенности и риски R&D-проектов в области искусственного интеллекта по сравнению с классическими проектами по разработке ПО?

7. Какие сложности возникают при применении стандартных Agile-практик к проектам с высоким уровнем исследовательской неопределенности и как их можно преодолеть?

8. Что такое пользовательская история (User Story)? Из каких компонентов она состоит и как декомпозируется на технические задачи?

9. Каким критериям должна соответствовать качественная User Story (INVEST)? Приведите пример хорошей и плохой пользовательской истории для системы предиктивной аналитики.

10. Что такое декомпозиция задач и почему она критически важна для успешного планирования? Каков основной принцип разделения задач (например, правило INVEST)?

11. Объясните разницу между оценкой в «идеальных часах» и использованием «story points». В чем преимущества относительной оценки (story points) перед абсолютной?

12. Какие методы оценки сложности задач вы знаете (например, T-shirt sizes, последовательность Фибоначчи, триангуляция)? В чем их суть?

13. Как инструменты вроде Jira или YouTrack помогают управлять бэклогом продукта? Назовите ключевые функции этих систем, используемые для планирования.

14. Что такое управление ожиданиями заказчика? Какие артефакты и практики используются для синхронизации видения продукта между командой и бизнесом?

15. В рамках практикума 2: как распределяются роли в команде во время планирования спринта и какие артефакты создаются по его итогам?

16. Каковы цели проведения код-ревью? Перечислите основные правила эффективного код-ревью, которые помогают поддерживать качество кода без демотивации разработчиков.

17. Опишите уровни автоматизированного тестирования (модульное, интеграционное, системное). Какую роль играют линтеры, форматтеры кода и контроль технического долга в обеспечении качества продукта?

18. Раскройте концепции Continuous Integration (CI), Continuous Delivery (CD) и Continuous Deployment (CDep). Как они взаимосвязаны в рамках DevOps-подхода?

19. Что такое Infrastructure as Code (IaC) и каковы преимущества управления инфраструктурой через код? Приведите примеры инструментов.

20. Что такое дрейф данных (data drift) и деградация модели? Какие метрики мониторинга необходимо настраивать для ML-модели в продакшене для своевременного обнаружения этих явлений?

Задания 2 типа

1. Заказчик требует от команды ИИ жесткого соблюдения сроков и детального плана на 6 месяцев вперед, но требования к модели постоянно меняются. Какую методологию вы выберете (Waterfall, Scrum, Kanban) и как убедите заказчика в ее эффективности? Какие артефакты вы предложите использовать вместо классического ТЗ?

2. В команде Data Science нет выделенного DevOps-инженера. Как вы организуете процесс развертывания модели в продакшен, используя принципы IaC (Infrastructure as Code), чтобы минимизировать зависимость от ручных операций?

3. Команда состоит из исследователей (Data Scientists), которые привыкли работать изолированно. Как вы примените принципы Agile для формирования кросс-функциональной команды и вовлечения их в общий процесс поставки ценности?

4. Бизнес-заказчик просит добавить в спринт критически важную функцию, которая не была запланирована. Как вы примените принципы Scrum для управления этим запросом, не нарушив целостность текущего спринта?

5. При анализе кейса «Система предиктивной аналитики» выяснилось, что 40% задач — это R&D (исследование новых архитектур). Как вы адаптируете стандартный двухнедельный спринт для такой неопределенности? Как измените критерии готовности (Definition of Done)?

6. Команда впервые оценивает задачу по подготовке датасета. Оценки разнятся от 2 до 20 story points. Опишите пошагово, как вы проведете сессию Planning Poker, чтобы прийти к консенсусу и выявить скрытые риски.

7. Бэклог продукта перегружен. Заказчик настаивает на реализации всех функций сразу. Применив технику MoSCoW, как вы приоритизируете задачи, чтобы показать заказчику, что часть требований может быть отложена без потери ценности?

8. ML-инженер оценивает задачу в «идеальных часах», но постоянно срывает сроки из-за переключений и встреч. Как вы переведете команду на оценку в story points и какие аргументы приведете инженеру для смены подхода?

9. В бэклоге есть техническая задача «Рефакторинг пайплайна обучения» и бизнес-задача «Добавление нового фичи». Как вы объясните Product Owner'у необходимость включения технической задачи в спринт, используя понятие технического долга?

10. Оценка задачи по написанию unit-тестов для сложной нейросети вызывает споры: одни говорят, что это легко, другие — что невозможно. Какой метод оценки сложности (например, T-shirt sizes или последовательность Фибоначчи) поможет снизить градус дискуссии и почему?

11. Во время код-ревью ML-инженер отклоняет pull request из-за нарушения PEP8, но его собственный код также не соответствует стандартам. Как вы, как Scrum Master, проведете ретроспективу и внедрите правило, которое решит эту проблему на системном уровне?

12. Модель на проде начала выдавать странные результаты. У вас есть подозрение на деградацию качества. Опишите пошаговый алгоритм действий команды, используя практики контроля качества и мониторинга.

13. Новый аналитик в команде пишет код на Python в стиле R (много вложенных циклов, нет векторизации). Это работает, но медленно. Как организовать процесс передачи знаний и контроля качества кода (code review), чтобы исправить ситуацию без конфликта?

14. Команда не пишет автотесты, аргументируя это тем, что «модели ML непредсказуемы». Какие виды автоматизированного тестирования (например, тесты на консистентность данных или smoke-тесты API) вы предложите внедрить в первую очередь?

15. При каждом коммите в ветку main падают тесты из-за конфликта версий библиотек (например, numpy). Опишите, как вы настроите CI-пайплайн (на примере GitLab CI/Jenkins) для изоляции зависимостей и предотвращения этой проблемы.

16. Обученная модель показывает отличную точность на тестовых данных, но после деплоя ее метрики резко падают через неделю. Опишите процесс настройки мониторинга и алертинга для раннего обнаружения деградации модели.

17. Data Scientist обучил новую версию модели и хочет заменить старую на проде простым копированием файла model_v2.pkl. Почему это плохой подход? Как применить принципы CI/CD и версионирования артефактов для безопасного обновления модели?

18. В CI-пайплайне сборка проекта занимает 40 минут из-за скачивания зависимостей при каждом запуске. Предложите конкретное решение для оптимизации пайплайна с использованием кэширования.

19. Вы обнаружили дрейф данных: распределение признаков на проде изменилось по сравнению с данными, на которых обучалась модель. Какие действия предпримет

команда согласно MLOps-практикам? Опишите цикл обратной связи.

20. Вам нужно запустить новый проект по компьютерному зрению с нуля.

а) Выберите методологию управления проектом и обоснуйте выбор.

б) Опишите структуру первого спринта и как вы декомпозируете задачу «Собрать датасет» на user stories.

в) Нарисуйте схему базового CI/CD-пайплайна для этого проекта (от коммита до деплоя).

Задания 3 типа

1. Декомпозируйте бизнес-требование «Пользователь хочет видеть прогноз продаж на следующий месяц» на 3–4 пользовательские истории (User Stories) по критериям INVEST. Оцените каждую историю в story points, используя последовательность Фибоначчи, и кратко обоснуйте свою оценку.

2. Дано 5 задач для ML-проекта. Примените метод MoSCoW, чтобы распределить их по категориям (Must have, Should have, Could have, Won't have). Обоснуйте, почему вы отнесли каждую задачу к конкретной категории.

1) Исправить критическую ошибку в продакшене
2) Написать unit-тесты для нового модуля
3) Исследовать новую архитектуру нейросети
4) Обновить документацию по API
5) Провести рефакторинг старого кода

3. У вас есть бэклог спринта с задачами, оцененными в 8, 5, 3 и 13 story points. Вместимость команды на спринт — 20 story points. Сформируйте оптимальный состав задач для спринта, чтобы максимизировать ценность, и объясните, почему вы не взяли задачу на 13 очков.

4. Вам предоставлен фрагмент Python-кода. Проведите его ревью: найдите как минимум одну логическую ошибку, одно нарушение PEP8 (стиля кода) и предложите одно улучшение производительности.

5. Напишите простой модульный тест (с использованием `pytest` или `unittest`) для функции, которая вычисляет среднее значение списка чисел. Тест должен проверять корректную работу функции с валидным списком и ее реакцию на пустой список.

6. Опишите последовательность шагов для настройки автоматического форматирования кода (например, с помощью `black`) и проверки стиля (с помощью `flake8` или `pylint`) в проекте на Python.

7. Напишите конфигурационный файл `.gitlab-ci.yml` для базового пайплайна, который выполняет два шага:

1. Устанавливает зависимости из `requirements.txt`.

2. Запускает тесты с помощью команды `pytest`.

8. Опишите процесс сохранения и регистрации новой версии ML-модели в репозитории артефактов (например, MLflow). Укажите, какие метаданные (кроме самой модели) необходимо логировать для обеспечения воспроизводимости эксперимента.

9. Опишите, какие ключевые метрики и данные необходимо собирать для мониторинга REST-API модели в продакшене для раннего обнаружения деградации ее качества (drift).

10. Представьте ситуацию: команда постоянно берет в спринт только бизнес-задачи, игнорируя технический долг. Как Scrum Master, предложите конкретную стратегию (в рамках одного или двух спринтов), чтобы убедить Product Owner'a выделить время на рефакторинг.