

# **Процессы работы команд**

Владислав Горбунов - Head of DS

# О чем поговорим

- Уровни процессов
- Какие бывают компании с ML / DS
- Модели оплаты команд и получение бюджета на проекты
- Приоритезация проектов
- Общий процесс реализации проектов с ML / DS
- Проработка новой идеи и старт проекта
- Модели и методологии
- Операционная деятельность
- Ретроспективы и развитие процессов работы
- Как выстроить свой процесс работы

# Уровни процессов

- Управление компанией
- Управление подразделением
- Управление продуктом
- Управление проектом
- Управление командой / операционной деятельностью

# Какие бывают компании с ML - командами

- Продуктовые (зарабатывают на IT-сервисах, в т.ч. с ML)
- Проектные (инхаус разработка, зарабатывают не на IT-сервисах)
- Проектные (подрядные работы / аутсорс, зарабатывают на “продаже” людей)
- Не IT (инхаус промпт-инжиниринг, предобученные модели для популярных задач)
- Исследовательские организации (финансирование из фондов)

# Модели оплаты команд

- Fixed Price - классический проектный подход
- Time & Materials - оплата по факту выполненных работ
- Мешок денег - инвестиция в функцию на период времени с ожидаемым возвратом инвестиций

# Получение бюджета на проекты

- Инвестиционный (проектный) комитет - вы не принимаете решения, только готовите материалы для защиты проекта и получения бюджета
- Инвестиция на подразделение - вы получили бюджет на работу подразделения на период времени, сами приоритезируете какие проекты и в каком порядке будете делать
- Приоритезация на стороне руководителя - вы не принимаете решения, можете попробовать убедить делать определенный проект

# Как может происходить приоритезация проектов?

RICE

- R – Reach (Охват): Сколько пользователей затронет проблема или гипотеза в определенный период времени.
- I – Impact (Влияние): Оценка степени влияния на каждого пользователя (сколько денег / времени даст) / насколько может повлиять на OKR / доход компании или на таргет модели
- C – Confidence (Уверенность): Уровень уверенности в данных / веры в гипотезу / в реальное существование проблемы / возможность достижения целевой метрики на аудитории.
- E – Effort (Усилия): Оценка сложности / объема работы, необходимой для выполнения проекта.

Можно закодировать каждое из значений от 1 до 5, перемножить и отранжировать все идеи

RICE может также использоваться некоторыми командами для приоритезации гипотез

# Пример RICE

# Проверка гипотез и их приоритезация

Упрощенный подход (фин. моделирование):

- Доход в месяц (охват \* влияние или потенциальный доход \* уверенность)
- Расход в месяц \* количество месяцев разработки + стоимость поддержки
- Точка возврата инвестиций / накопленный доход на точку во времени

Приоритезация по планируемому накопленному доходу на точку во времени (или другому аналогичному KPI)

# Как может происходить приоритезация проектов?

1. Собрать все гипотезы
2. Выбрать метрики соответствующие стратегическим целям компании / текущим вызовам / OKR / обратной связи пользователей, поддержки, продаж
3. Отранжировать по RICE или упрощенному подходу
4. Определить какая доступна специализация команды для выполнения задач / интерес для роста в новую специализацию
5. Отрезать нужное количество задач, которые можно сделать за разумное время

# Как выглядит работа над проектом?

- Предварительная проработка (бизнес анализ) / Пред-проект / Старт проекта
- Обзор и анализ существующих решений
- Построение базового решения - эвристики / no-shot learning
- Построение ML базового решения - линейки / few-shot learning
- Построение качественного ML решения - проведение исследований и разработки
- Подготовка документации об исследованиях и реализованных моделях
- Внедрение результатов
- Поддержка и развитие решения

# Предварительная проработка проекта

- Понять бизнес-цели (помочь заказчику их сформулировать)
- Определить возможные подходы и решения по запросу от заказчика
- Обозначить технологические пределы возможностей
- Продумать риски
- Повторно продумать и скорректировать бизнес-цели
- Подготовить план работ по проекту
- Получить деньги на работу по проекту

**Как может выглядеть процесс проработки проекта**

# Вопросы для проработки проекта

# Старт проекта

Проект успешно защищен, вы получили согласование от руководства на начало работ.

Как могут выглядеть процессы работы?

# Ценности

Agile manifesto v1

- **Люди и взаимодействие** важнее инструментов и процессов
- **Работающий продукт** важнее исчерпывающей документации
- **Сотрудничество с заказчиком** важнее согласованных заранее условий контракта
- **Готовность к изменениям** важнее следования первоначальному плану

Agile manifesto v2

- **Командная работа и ответственность** важнее отдельных людей и их взаимодействия, что важнее процессов и инструментов
- **Бизнес-ценность** важнее работающего продукта, что важнее исчерпывающей документации
- **Построение партнёрства** важнее сотрудничества с заказчиком, что важнее согласования условий контракта
- **Подготовка к изменениям** важнее готовности к (реакции на) изменениям, что важнее следования первоначальному плану

# Ценности

А если взять научный метод, можно сформулировать подобные ценности в компании:

- **Открытость** важнее чем догматизм - Знания постоянно развиваются и открытость к новым идеям и исправлению ошибок имеют большее значение, чем строгое придерживание устоявшихся теорий или методик.
- **Воспроизводимость** важнее чем уникальность - Подтверждение результатов через воспроизводимость экспериментов ценится выше, чем однократные и уникальные результаты, которые нельзя повторить или проверить.
- **Критическое мышление** важнее чем авторитет - Вопросы и сомнения в отношении данных, методов и выводов считаются более важными, чем слепое следование авторитетам или традициям. Это подчеркивает важность независимой оценки и проверки полученных результатов.
- **Прозрачность** важнее секретности - Делиться методами, данными и результатами с научным сообществом (или хотя бы внутри компании) считается более важным, чем скрывать их. Это способствует коллективному прогрессу.



# Методологии работы команд DS / MLE

Подходы, которые не имеют фокуса на команды специалистов по данным:

- Classic Project Management (a.k.a. waterfall. But Iterative / Incremental / Spiral. Based on PMI - PMBoK)
- Scrum / Scrum-But (iterative & incremental)
- IT Kanban (flow)

# Методологии работы команд DS / MLE - PMI

**Classic Project Management** - Фреймворк, предоставляющий огромное количество инструментов, которые нужны, чтобы в условиях большой неопределенности уложиться в заданные ограничения по времени, стоимости и объему работ так, чтобы все заинтересованные люди были удовлетворены результатом, а продукт поставлен конечным пользователям:

- Управление содержанием (как определить что нужно сделать для успеха проекта)
- Управление расписанием (как определить очередность работ и составить расписание)
- Управление стоимостью (как определить стоимость, заложить резервы)
- Управление качеством (как поймем что то что делается и как это делается достаточно хорошо для всех стейкхолдеров)
- Управление рисками (как будем работать с вероятными событиями и извлекать из них пользу / реагировать на плохие события)
- Управление коммуникацией (как, с кем и как часто нужно общаться для успеха проекта)
- Управление человеческими ресурсами (как создавать, развивать команду и управлять людьми в процессе создания продукта)
- Управление заинтересованными лицами (как найти и управлять вовлеченностью людей, способных повлиять на проект)
- Управление закупками (как закупить услуги / оборудование)
- Управление интеграцией (как объединить все вышеперечисленное в единый план работ)

Чаще всего работа происходит по Fixed Price, когда определяется стоимость за объем работ, которые будут выполнены к фиксированной дате.

Классический подход может оказаться неэффективным для большинства Data Science проектов из-за их исследовательской природы и необходимости адаптации к новым данным и инсайтам по мере продвижения проекта.

Некоторые компании вынуждены использовать данный подход из-за специфики своей работы. Чаще всего проекты бьются на небольшие этапы, например в 3 месяца для проверки ограниченного количества гипотез.

**Основной минус** - нельзя менять гипотезы, запланированные к проверке

**Основной плюс** - множество инструментов, которые можно применять в своей работе не привязываясь к рамкам Fixed Price

# Методологии работы команд DS / MLE - Scrum

**Scrum** — это фреймворк для управления проектами, используемый в основном в разработке ПО. Включает в себя итерации, называемые спринтами, длительностью от одной до четырех недель. Каждый спринт начинается с планирования и заканчивается обзором инкремента продукта, что позволяет команде адаптироваться к изменениям и постоянно улучшать процесс работы.

- Команда - кроссфункциональная, состоит из разных специалистов. Есть выделенные роли Владельца продукта и Scrum-мастера
- Sprint - период времени в 1-4 недели, за которые происходит инкремент функциональности продукта.
- User Stories - единица задачи, способ описания требований к разрабатываемой системе, сформулированных на повседневном или деловом языке пользователя, как несколько предложений: “Я как (роль) хочу делать (действие) и получать (результат)”.
- Backlog - набор User Stories, которые на языке бизнеса отображают, что необходимо сделать. Фактически - приоритизированные хотелки от владельца продукта.
- Sprint Planning - Процедура выбора на спринт из бэклога конкретных задач, часто сопровождается грумингом - относительной оценкой задач по сложности.
- Sprint Review- Демонстрация результата работы команды заказчику
- Daily Meeting - Ежедневные встречи на 5-30 минут, на которых каждый проговаривает (Что сделал вчера, Что планирую делать сегодня, Какие есть проблемы
- Ретроспектива - групповой отзыв о процессах проделанной работы, планирование улучшения качества процесса

“Scrum-but” — это термин, используемый для описания ситуаций, когда организации или команды утверждают, что они используют Scrum, но вносят изменения в его ключевые элементы или принципы, оправдывая это особыми обстоятельствами. Эти изменения часто подрывают основные принципы гибкости и адаптивности, которые делают Scrum эффективным.

## **Основные минусы:**

- Не всегда можно точно оценить объем работ, который потребуется для полной проверки гипотезы, если заранее не подготовлен дизайн эксперимента.
- После завершения проверки гипотезы и появления новых гипотез нужно ждать конца спринта, чтобы взять новую (а иногда конца следующего спринта)

# Методологии работы команд DS / MLE - Scrum

# Методологии работы команд DS / MLE - IT-Kanban

IT-Kanban - Конвейер задач. Имеет всего 3 правила:

- Визуализация процесса разработки с помощью канбан-доски,
- Ограничение на количество задач на каждом этапе,
- Постоянное измерение производительности команды и улучшение процессов.

В канбане нет спринтов, из бэклога берется определенное число задач (чаще всего эти задачи даже не имеют оценку по сложности и длительности), на каждую задачу назначается исполнитель.

Задача мигрирует по доске по категориям выполнения до тех пор, пока не будет сделана.

**Основной плюс** - после завершения проверки гипотезы можно сразу взять нужную следующей

**Основной минус** - не понятно какие задачи нужно брать, как приоритезировать, нужна надстройка над конвейером для максимизации бизнес-ценности\*\*

# Методологии работы команд DS / MLE - CRISP-ML(Q)

- CRISP-DM - Авторы не включали деплой результатов исследований. Фокус был на майнинге чего-то из данных. Модели были не основным артефактом.
- CRISP-ML - Добавили деплой, обозначили что работа идет еще и над созданием моделей.
- CRISP-ML(Q) - Более подробно раскрыли каждый пункт и добавили проверку качества на каждом этапе, включая мониторинг работы в продуктиве, практики DevOps и MLOps.

# Методологии работы команд DS / MLE - CRISP-ML(Q)

## Понимание бизнеса и данных;

- Определить бизнес-цели;
- Преобразовать бизнес-цели в цели ML;
- Собрать и проверить данные;
- Оценить осуществимость проекта;
- Провести proof of concept;

## Подготовка данных;

- Выбор признаков;
- Преобразование данных;
- Баланс классов;
- Очистка данных;
- Конструирование признаков;
- Аугментация данных;
- Стандартизация данных;

## Моделирование;

- Определить метрику оценки качества модели;
- Выбор алгоритма ML;
- Добавление знаний предметной области для специализации модели;
- Тренировка модели;
- Документирование модели и экспериментов ML;
- Необязательно:
  - Применение трансферного обучения (с использованием предварительно обученных моделей);
  - Оптимизация модели;
  - Обучение ансамблей;

## Оценка;

- Проверка работоспособности модели;
- Определить робастность модели;
- Повысить объяснимость модели;
- Принять решение о развертывании модели;
- Документирование этапа оценки;

## Развертывание;

- Оценить модель в условиях работы в production;
- Обеспечить принятие пользователем и удобство использования;
- Использование практик управления моделями (Model Governance);
- Развертывание по выбранной стратегии;

## Мониторинг и поддержка;

- Мониторинг работы модели;
- Сравнение с ранее указанными критериями успеха (пороговыми значениями);
- Переобучение модели при необходимости;
- Сбор новых данных;
- Разметка новых точек данных;
- Повторение задач из этапов Моделирование и Оценка;
- Непрерывная интеграция, обучение и развертывание модели;

# Методологии работы команд DS / MLE - CRISP-ML(Q)

## Научный метод - Характеристики предмета исследования

- **CRISP-ML(Q) - Понимание бизнеса и данных**

Это единственный шаг, который полностью занят тем, чтобы правильно обозначить предмет исследования, однако в нем отсутствует фиксация наблюдения. Четко не определено, фиксируются ли измерения и определения.

- **CRISP-ML(Q) - Подготовка данных**

В этом пункте происходит ограничение области исследования, так как выбираются признаки, выполняется первичное преобразование данных для изучения объекта исследования.

Однако, этот шаг как будто включает в себя сразу и создание гипотез и их проверку, иначе не понятно, на основании чего делается отбор.

## Научный метод - Гипотезы и теории

- Работа с формированием гипотез в целом в данном методе отсутствует, возможно, она предполагается, однако явного указания нет.
- Складывается ощущение что это просто метод проб и ошибок, на основании чего и выбирается подходящая модель, а не методичный поиск наиболее вероятного ответа.

# Методологии работы команд DS / MLE - CRISP-ML(Q)

## Научный метод - Прогнозы

- Фиксируются метрики оценки, но это больше относится к характеристикам предмета исследований, чем к этому элементу научного метода.  
Так как отсутствует работа с гипотезами, то и не указывается, каким образом фиксируются прогнозы сделанные на их основе.

## Научный метод - Эксперименты

- **CRISP-ML(Q) - Моделирование;**
  - В этом шаге как будто совмещается то, что должно было бы быть в характеристиках предмета исследования, в формировании гипотез, прогнозе, а затем и в экспериментах.
  - Но вместо этого мы выбираем просто алгоритм, метрику оценки качества и гоняем модель, пока не получим результат.
- **CRISP-ML(Q) - Оценка;**
  - Этот шаг также включает в себя несколько элементов.
  - Есть важные пункты про проверку работоспособности и определение робастности, так как это полезные показатели при проведении экспериментов с моделями и последующей оценке результатов.

# Методологии работы команд DS / MLE - CRISP-ML(Q)

Практическое применение полученных результатов (внедрение в продуктив и MLOps).

- **CRISP-ML(Q) - Развертывание;**
  - Шаги отражают современные подходы к развертыванию сервисов с моделями машинного обучения.
- **CRISP-ML(Q) - Мониторинг и поддержка;**
  - Инженерные практики подчеркивают важность последующего мониторинга, сбора данных и улучшения модели.

## Вывод:

- CRISP-ML(Q) хорошая инженерная методология, которая, к сожалению, забывает про часть научного метода, связанную с работой с гипотезами, и таким образом выводит на первый план экспериментальный метод проб и ошибок.

# Методологии работы команд DS / MLE

Специализированные методологии (Имеют фокус на команды специалистов по данным)

- CRISP-DM / CRISP-ML / CRISP-ML(Q)
- Microsoft TDSP
- Domino DS Lifecycle
- RAMSYS
- Agile Data Science Lifecycle
- MIDST
- Development Workflows for Data Scientists
- Big Data Ideation, Assessment and Implementation
- Big Data Management Canvas
- Agile Delivery Framework
- Systematic Research on Big Data
- Big Data Managing Framework
- Data Science Edge
- Foundational Methodology for Data Science
- Analytics Canvas
- AI Ops
- Data Science Workflow
- EMC Data Analytics Lifecycle
- Toward data mining engineering

[Data Science Methodologies: Current Challenges and Future Approaches](#)

# Методологии работы команд DS / MLE

Фокус в различных подходах:

## Управление проектами:

- Определение рабочих процессов в рамках жизненного цикла Data Science
- Стандартизация структуры каталогов
- Систематическая документация проекта (обновление документации в течение всего жизненного цикла проекта)
- Визуализация текущего статуса проекта
- Синхронизация целей бизнеса и задач Data Science
- Согласование метрик успеха (бизнес-метрики и технические метрики)
- Разграничение прототипа и финального продукта (выделение этапов разработки продуктов)

## Управление командой:

- Распространение и обсуждение научных инсайтов (внутри / вне компании)
- Определений ролей и их обязанностей для улучшения координации внутри команды и с заинтересованными сторонами
- Работа в команде: рабочие процессы Git и договорённости по написанию кода

## Управление данными и информацией:

- Обеспечение воспроизводимости: создание хранилища знаний (базы знаний)
- Надёжное развёртывание: управление версиями кода, данных и моделей
- Разработка моделей для генерации знаний и ценности бизнесу
- Фокус на процессе, а не только на результате

# Методологии работы команд DS / MLE

# **Операционная деятельность - как может выглядеть процесс**

# Ретроспективы

Ретроспектива - практика итеративного улучшения процессов работы

- **Рефлексия на за прошедший период времени:** Команда обсуждает достигнутые результаты, анализирует успешные моменты и проблемы в процессе работы (Что понравилось, что не понравилось в работе, что хотелось бы изменить).
- **Приоритезация задач по улучшению:** Команда ранжирует задачи по улучшению процессов (например распределяет задачи по шкалам Важность / Сложность, оценивая задачи относительно друг-друга и приоритезирует задачи)
- **Планирование действий по улучшению:** Команда договаривается о конкретных действиях, направленных на улучшение процесса - кто в команде что будет делать, чтобы улучшить процессы.

В один момент времени лучше брать в работу не более 2-3 задач, например:

- **Важная и относительно простая задача** (к следующей ретроспективе скорее всего внедрите изменение и начнете собирать данные)
  - Систематизировать процесс документирования кода для улучшения понимания и поддержки кодовой базы в долгосрочной перспективе. Это ускорит процесс введения новых сотрудников в проект и упростит поддержку существующих решений.
- **Самая важная задача, с высокой сложностью** (к следующей ретроспективе возможно удастся снизить сложность задачи)
  - Создание инфраструктуры и процессов для автоматизации обновления и переобучения моделей машинного обучения на основе новых данных. Это требует сложной работы с инфраструктурой, алгоритмами и процессами проверки качества моделей, чтобы гарантировать их актуальность и эффективность.
- **Самая простая задача** (к следующей ретроспективе скорее всего внедрите изменение и начнете собирать данные)
  - Разработать и внедрить стандартные шаблоны для еженедельных отчетов о ходе проекта DS, чтобы упростить подготовку отчетов и обеспечить единообразие представления результатов работы

# Куда можно посмотреть для улучшения работы?

Совместная работа:

- Процессы работы
- Стандарты и шаблоны
- Обмен знаниями
- Обмен кодом и результатами исследований
- Обмен данными

Воспроизводимость исследований:

- Пайплайны и их оркестрация
- Документация по исследованиям
- Рабочее место и виртуальные окружения

Продуктовизация исследований

- Автоматизация в процессе создания моделей
- Хранилище моделей
- Сборка и поставка приложений
- Model API / Построение сервисов
- Мониторинг работы моделей

# Как выстроить свой процесс работы?

## 1. Определиться с ограничениями на уровне организации

1. Определить набор ценностей в подходе к созданию ML-решений (XХиВП, Agile manifesto v1/2, Monkey-ml, Научный метод, ...)
2. Какая модель финансирования вашего подразделения
3. Как выглядит процесс повышения ФОТ подразделения?

## 2. Проектный / продуктовый подход

1. Проектный или продуктовый подход в компании?
2. Какие инструменты и ограничения используются?
3. Определить правила проработки и старта новых проектов, согласовать процесс
4. Как будем приоритезировать и защищать проекты / гипотезы - на уровне проектных / продуктовых комитетов
5. С какой периодичностью нужна отчетность и какая отчетность / какая этапность проектов?

## 3. Начать строить процесс

1. Если уже как-то работаете - продолжать работать также + внедрить ретроспективы.
2. Если новый отдел, взять одну из моделей и одну из методологий на базе модели за основу + внедрить ретроспективы. (например, Spiral + IT-Kanban)

## 4. Итеративно улучшать процессы через ретроспективы, внедрять новые практики и инструменты

1. . Процессные практики - Как выглядит ваша операционная деятельность, как выполняются задачи, какие артефакты (подсматриваем в куче других методологий)
2. Проектные / продуктовые практики - Какие инструменты проектного (PMI) и продуктового управления надо использовать
3. Практики работы с данными и повышения качества исследований - Научный метод и воспроизводимость
4. Инженерные практики (MLOps) - Тестируете инструменты и пробуете в своей работы, замеряете эффективность

# Полный процесс исследовательской части можно описать так:

- Совместно генерируются гипотезы по проекту и прогнозы по гипотезам.
- Гипотезы отдаются на дизайн экспериментов отдельным DSам, чтобы составить план экспериментов.
- План экспериментов проходит ревью в команде, что всё ок и такие шаги действительно поддержат гипотезу или покажут противоречие ей.
- Проработанные гипотезы распределяются на исполнителей.
  - В идеальной ситуации, чтобы максимально избежать предвзятости, можно сделать так чтобы один человек предложил гипотезу, второй задизайнил эксперимент, третий человек провел эксперимент.
  - Если так не можете сделать, лучше чтобы дизайн эксперимент и проходил по его шагам человек, который не является автором гипотезы. В таком случае важно не поддаваться соблазну и не отойти от шагов эксперимента.
  - Если нет выхода то эксперимент будет дизайнить и проверять автор гипотезы, но ему стоит помнить о ловушке предвзятости и обязательно пройти ревью дизайна и не отходить от шагов эксперимента.
- По завершению исследования готовится набор артефактов - вывод, описание проделанных шагов, код, данные которые также проходит ревью - проверяется воспроизводимость методов/результатов.

Чем больше знаний о данных (вашем предмете исследования) вы накопите таким образом, тем более качественные модели сможете построить.

Задачи по внедрению инструментов так же можно проводить как эксперимент, для которого вы зафиксируете гипотезу и прогноз - чего хотите достичь внедрением такого инструмента в свои процессы (на что повлияете, как измерите) - чтобы случайно не попасться на эффект новизны и не использовать инструмент ради инструмента

Выбор задач по исследованиям на данных или моделях / апробации ml-ops инструментов целиком лежит на команде. Скорее всего в реальности апробацию конкретных инструментов вы будете делать по результатам обсуждения проблем на ретроспективах, отталкиваясь от проблем в ваших процессах. Кто конкретно возьмется за апробацию инструментов - решать команде. И никто не ограничивает возможность параллельной проверки инструмента или воспроизведения результатов проверки другим человеком в команде.

В ближайших лекциях как раз будет разбираться основной инструментарий для работы над исследованиями в команде и для обеспечения воспроизводимости