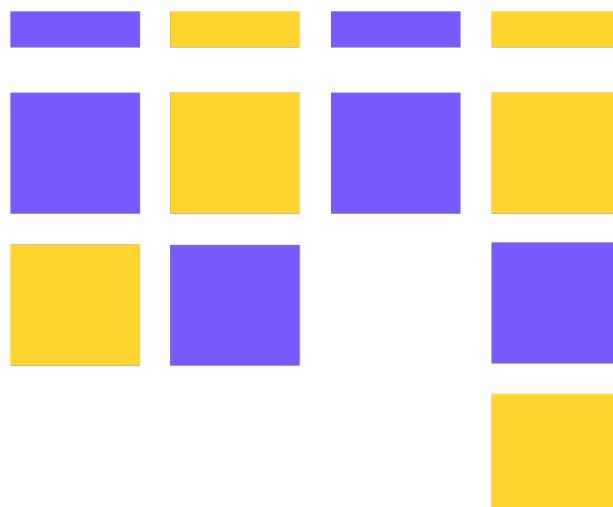


Асхат Уразбаев

# Управление DS проектами и продуктами

при помощи

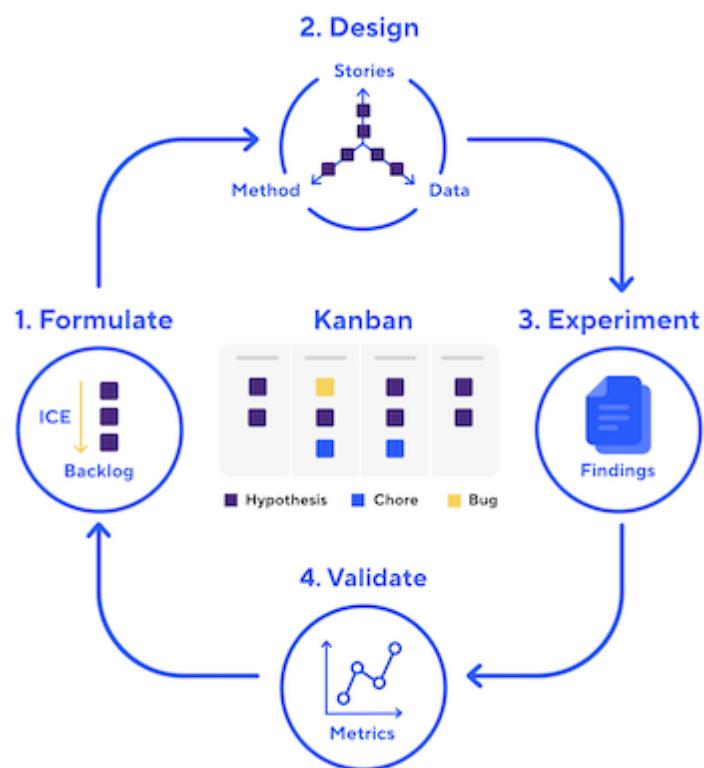
# Lean Data Science



# Lean Data Science

Управление DS проектами и продуктами при помощи Lean Data Science

Асхат Уразбаев



## Содержание

<b>Благодарности</b>	<b>5</b>
<b>Введение</b>	<b>6</b>
Почему данные проекты проваливаются? . . . . .	6
<b>What is Lean DS</b>	<b>7</b>
О структуре книги . . . . .	7
<b>AI Canvas</b>	<b>8</b>
Какую проблему решаем . . . . .	8
AI Canvas Structure . . . . .	8
Как использовать AI Canvas . . . . .	9
Как заполнять канвас . . . . .	9
Когда нужно использовать канвас . . . . .	11
Когда не нужно использовать канвас . . . . .	11
Преимущества использования AI Canvas . . . . .	11
Преимущества использования AI Canvas по сравнению с ТЗ . . . . .	12
<b>Story Mapping</b>	<b>13</b>
Какую проблему решаем . . . . .	13
Проектируем продукт при помощи User Story Map . . . . .	13
Как создать Story Map . . . . .	14
Пример Story Map . . . . .	15
Хинты создания Story Map . . . . .	16
Когда использовать Story Map . . . . .	17
Когда нет смысла использовать Story Map . . . . .	17
Преимущества Story Mapping . . . . .	17
Практическое использование Story Mapping . . . . .	18
About Story Mapping . . . . .	18
<b>Гипотезы</b>	<b>19</b>
Какую проблему решаем . . . . .	19
Свойства традиционной постановки . . . . .	19
К каким проблемам может привести традиционная постановка задач . . . . .	20
Три типа гипотез . . . . .	20
Продуктовая гипотеза . . . . .	21
Формулирование данных- и метод- гипотез . . . . .	21
Правила создания DS-гипотезы . . . . .	22
Приемочные критерии (Acceptance Criteria) . . . . .	22
Пример приемочных критериев . . . . .	22
Новая гипотеза . . . . .	23
Преимущества использования гипотез . . . . .	24
Когда не надо формулировать гипотезы . . . . .	24

<b>Приоритизация ICE/RICE</b>	<b>25</b>
Какую проблему решаем . . . . .	25
Приоритизация методом ICE . . . . .	25
Impact . . . . .	25
Effort . . . . .	26
Confidence . . . . .	26
Как провести сессию по приоритизации гипотез . . . . .	27
Приоритизация при помощи RICE . . . . .	28
Особенности проведения приоритизации . . . . .	28
Как развивать подход к приоритизации . . . . .	29
Зачем нужна приоритизация . . . . .	29
Когда не надо использовать ICE/RICE . . . . .	30
Когда использовать ICE/RICE . . . . .	30
<b>Декомпозиция методом «Мерседеса»</b>	<b>31</b>
Зачем нужно планирование . . . . .	31
Правильный учет рисков . . . . .	31
Планирование методом Мерседеса . . . . .	33
Подготовка к планированию . . . . .	33
Шаг 1. Брейншторминг гипотез . . . . .	33
Шаг 2. Выделение Proof of Concept/MVP . . . . .	35
Шаг 3. Анализ рисков по чек-листам . . . . .	36
Особенности использования подхода «Мерседес» . . . . .	37
Преимущества использования . . . . .	38
<b>Вопросы (Questions)</b>	<b>39</b>
Story questions . . . . .	39
Data questions . . . . .	39
Method questions . . . . .	39
Примеры использования вопросов . . . . .	39
<b>Кроссфункциональная команда</b>	<b>41</b>
Какую проблему решаем . . . . .	41
Кросс-функциональная команда . . . . .	42
T-shape professional . . . . .	43
Основные возражения против кросс-функциональности команд . . . . .	43
Команда DS as a Service . . . . .	44
Когда использовать кросс-функциональные команды не эффективно . . . . .	44
Преимущества кросс-функциональной продуктовой команды . . . . .	45
Лидерские роли в команде . . . . .	45
<b>Выбор подхода</b>	<b>46</b>
Какую проблему решаем . . . . .	46
В чем причина таких проблем? . . . . .	46
<b>О канбане</b>	<b>48</b>
Практики канбан . . . . .	48

<b>Визуализация работ</b>	<b>49</b>
Колонки канбан-доски . . . . .	49
Блокеры . . . . .	49
Визуализация ответственности . . . . .	50
Вытягивание . . . . .	51
Тикет . . . . .	52
Пример тикета в Trello . . . . .	52
<b>Типы работ (Work Item Types)</b>	<b>53</b>
<b>Ограничение количества одновременно выполняющейся работы</b>	<b>54</b>
Закон Литтла . . . . .	54
Как работает ограничение WIP в Канбан . . . . .	55
<b>Срочные задачи (Expedite) и Swimlanes</b>	<b>57</b>
<b>Явные политики (Explicit Policies)</b>	<b>58</b>
Пример Explicit Policies . . . . .	59
<b>Встречи в канбан</b>	<b>60</b>
Stand-up meeting aka Kanban-meeting . . . . .	60
Планирование aka replenishment meeting (встреча по пополнению) . . . . .	60
Ретроспектива aka Ревью сервиса поставки . . . . .	61
Формат проведения ретроспективы . . . . .	61
<b>Управление потоком работ</b>	<b>63</b>
Lead Time Distribution . . . . .	63
Throughput Diagram . . . . .	63
<b>Практическое использование LeanDS</b>	<b>65</b>
Нужно ли использовать все практики LeanDS? . . . . .	65
Как проводить внедрение? . . . . .	65
<b>Об авторе</b>	<b>66</b>

## Благодарности

От всего сердца хочу поблагодарить Алексея Могильникова и Юлию Рубцову. Вместе со мной они являются соавторами подхода LeanDS. Они активно развивают, придумывают новые идеи, убивают неработающие и обучаются подходу команды по всему миру. Круто быть частью такой команды!

Отдельно огромное спасибо всем, кто участвовал в работе сообщества LeanDS: презентовал свои методы на митапах, участвовал в обсуждениях и пробовал на практике LeanDS.

И наконец, передаю много kudos сообществу ODS.ai и его основателю Алексею Натекину за поддержку и участии в работе!

## Введение

Издание VentureBeat утверждает<sup>1</sup>, что 87% DS проектов не доходят до прода. Это не единственное такое исследование, есть и другие<sup>2</sup>. В целом цифры схожи.

В среднем в ИТ ситуация лучше. Согласно другому исследованию Gartner, ИТ проекты проваливаются в 50% случаях. Под провалом имеют в виду недостижение целей, перерасход бюджета или непопадание в сроки. То есть в итоге ИТ проекты хотя бы доходят до промышленной эксплуатации.



**Рис. 1:** Она кусается?

87 % Data Science проектов не попадают в прод

### Почему data проекты проваливаются?

В книге мы рассмотрим несколько причин провалов AI проектов и разберемся, как перестроить работу так, чтобы увеличить вероятность успеха. Вот эти причины:

- Разные цели у участников проекта
- Разрыв в коммуникациях между бизнесом и командой
- Низкая прозрачность процесса
- Недостаточное командное взаимодействие

Индустрия AI прошла недостаточно большой путь, чтобы обрасти необходимыми методами, инструментами и практиками. В то же время коллеги из разработки ПО худо-бедно научилисьправляться с этими сложностями. Можем ли мы просто скопировать их подходы?

И да, и нет. Конечно, нет смысла вслепую проходить тот путь, который прошла индустрия разработки ПО, однако Data Science проекты имеют свою специфику и слепое копирование приносит больше вреда, чем пользы.

В этой книге мы поговорим о практиках, большинство из которых были позаимствованы из Software Development и были адаптированы к Data Science проектам и продуктам.

<sup>1</sup><https://venturebeat.com/2019/07/19/why-do-87-of-data-science-projects-never-make-it-into-production/>

<sup>2</sup><https://designingforanalytics.com/resources/failure-rates-for-analytics-bi-iot-and-big-data-projects-85-yikes/>

## What is Lean DS

Lean Data Science – открытый подход к управлению DS проектами и продуктами –

- Ориентированный на бизнес
- Основанный на гипотезах
- Командный и итеративный
- Строгий и воспроизводимый

Это позволит увеличить вероятность успеха за счет прозрачности процессов, снижения Time-to-Market и фокуса на инженерном совершенстве

LeanDS ориентирован на бизнес. Мы обсуждаем финансовую эффективность ML проектов и фокусируемся на их ускорении. Тем не менее, отдельные практики LeanDS можно использовать в исследовательских и тренировочных проектах.

Ключевая идея LeanDS – явная и четкая формулировка гипотез.

LeanDS содержит практики командной работы. Работа ведется итеративно и инкрементально – от простого к сложному с фокусом на быстрой поставке ценного результата.

Строгий и воспроизводимый. Техническое совершенство кода, автоматизация тестирования и поставки очень важно для достижения хорошего бизнес-результата.

## О структуре книги

Мы пройдем по всему жизненному циклу создания ML продукта – от идеи до реализации и рассмотрим практики позволяющие упростить и ускорить его создание.

Мы рассмотрим старт нового ML-продукта и начнем с создания «Технического Задания» на продукт в виде AI Project Canvas. ML-продукт может оказаться частью большего продукта и мы расскажем, как его спроектировать в главе Story Map.

Мы обсудим как ставить задачи членам команды в виде продуктовых гипотез и поговорим об их приоритизации. Для планирования работ по гипотезе воспользуемся подходом «Мерседес».

Мы поговорим о структуре команды, а в последних главах рассмотрим более подробно подход Канбан в применении к DS проектам.

## AI Canvas

### Какую проблему решаем

Вы лидер DS-команды. Вас в коридоре поймал за пуговицу кто-то из бизнеса и начал рассказывать вам про модель, которую надо сделать.

Вот в чем идея: У компании есть чат. Сотрудники call-центра (агенты) отвечают на вопросы клиентов. Было бы здорово автоматизировать этот процесс: пусть на вопросы клиентов отвечает бот!

Все понятно! Приступаем к работе. С чего начнем? Пожалуй, надо посмотреть, какие данные есть по существующим диалогам с клиентами и исследовать SOTA по этой теме...

Стоп! Приступать к работе еще рано. А действительно, что может пойти не так?

Да что угодно! Вашу модель не готовы будут интегрировать команда CRM, потенциальная экономия от использования модели может не окупить трудозатраты, директор call-центра откажется вносить изменения в работу подразделения, бизнес может иметь сильно завышенные представления о результатах работы модели и откажется принимать результаты и многое что еще. Все эти не случившиеся пока проблемы (мы будем называть их рисками) могут привести к провалу проекта.

Приступать к работе рано. Нужно синхронизировать ожидания всех участников проекта. Если мы обнаружим проблему вроде перечисленных выше, мы сможем ее обсудить и решить. Если вдруг она нерешаема, мы можем убить проект до его начала и сэкономить несколько месяцев работы квалифицированных и дорогих специалистов.

Давайте обсудим все наши вводные предположения и ожидания! Для проведения такого совещания мы воспользуемся AI Canvas.

AI Canvas – односторонний шаблон концепции (vision) AI проекта, документирующий его ключевые предположения

### AI Canvas Structure

AI Canvas состоит из 10 блоков с ключевыми вопросами, на которые нужно ответить перед стартом проекта.

В течении совещания его нужно заполнить стикерами-идеями по проекту.

Цель – обеспечить непротиворечивость и целостность концепции проекта

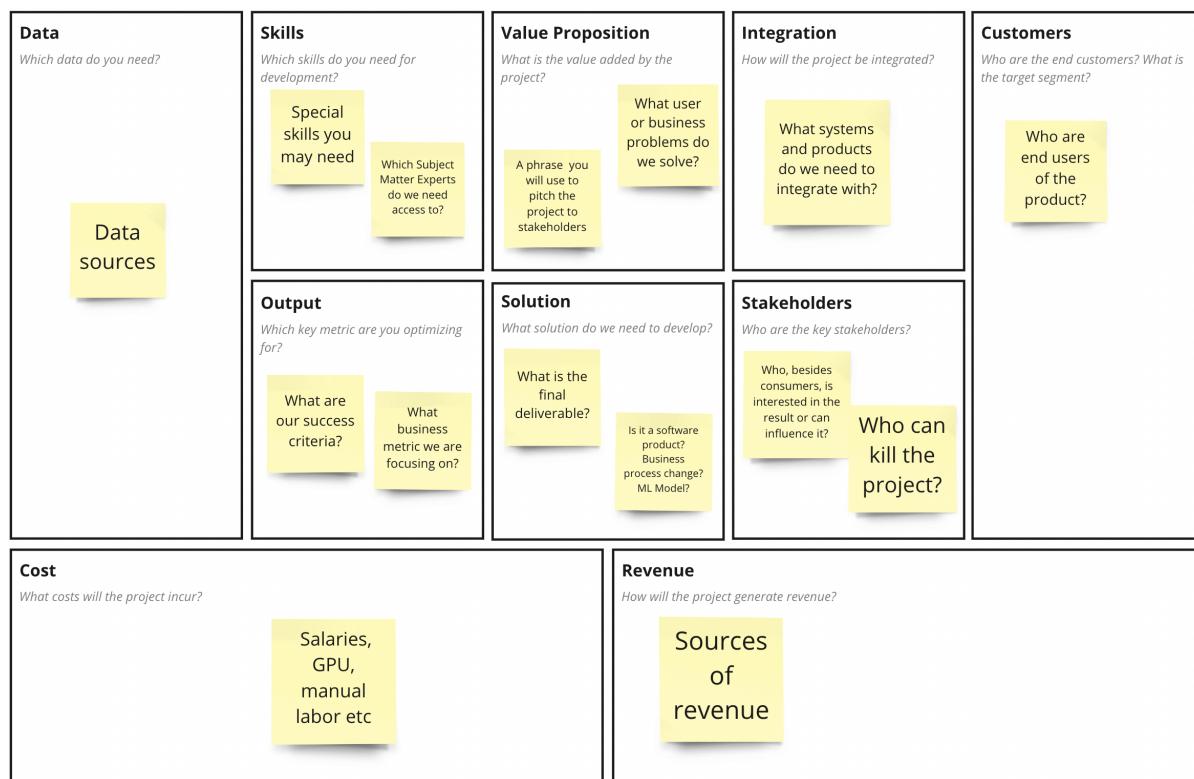


Рис. 2: AI Canvas Structure

## Как использовать AI Canvas

Вы собираете совещание со всеми заинтересованными лицами:

- Представители бизнеса (заказчики)
- Члены вашей команды
- Специалисты предметной области
- Представители смежных команд и служб

Общее правило такое: стараемся вовлечь тех, кто может дать обратную связь по предстоящему проекту.

## Как заполнять канвас

- Лучше начать заполнять канвас с блоков Customer и Value Proposition
- Помните, что это просто идеи, любой участник может повесить стикер!
- Не застревайте долго на обсуждении, если нет информации. Не знаете – повесьте стикер с вопросом. Я предпочитаю выделять такие вопросы другим цветом
- Проверяйте сами и просите участников проверить, насколько непротиворечивой получилась итоговая картинка.
- Канвас может не получиться с первого раза. Дайте себе домашнее задание ответить на вопросы и запланируйте отдельное совещание после их проработки
- Не удалось затащить на совещание важного стейкхолдера? Не беда, до него можно дойти позже и валидировать результат.

- Бывает так, что по итогам заполнения канваса выясняется что проект даже теоретически не имеет смысла делать. В этом случае мы можем принять решение о его остановке или заморозке. Поздравим друг друга, мы сэкономили кучу времени для других проектов и не стали делать демотивирующую работу «в стол»



Рис. 3: Change my mind!

#### №№ Пример заполнения канваса

Итак, вы провели совещание со всеми заинтересованными лицами. По итогам совещания выяснилось несколько интересных фактов:

- Представители бизнеса считали, что можно автоматизировать 80% ответов на вопросы. Ваша реалистичная оценка — 15% в ближайший год
- СОО компании категорически против, чтобы автоматизировать общение с сегментом Effluent — пользователями на дорогом тарифе. Наш фокус — массовый сегмент
- Массовый сегмент сейчас обслуживает 15 человек и реально их снизить на 2-3. Кажется, нет смысла затевать проект ради такой экономии?
- СОО очень интересовался возможностью ускорить ответы пользователям. Это могло бы помочь снизить отток пользователей
- Ходят слухи, что команда CRM сейчас переходит на новую систему и не готово к интеграции. И на совещание они не пришли, потому что им некогда.

Вы наметили три вопроса на проработку и договорились встретится еще раз через неделю.

Что произошло? Мы выявили ключевые риски проекта, без решения которых не имеет смысла серьезно вкладываться в техническую часть работы.

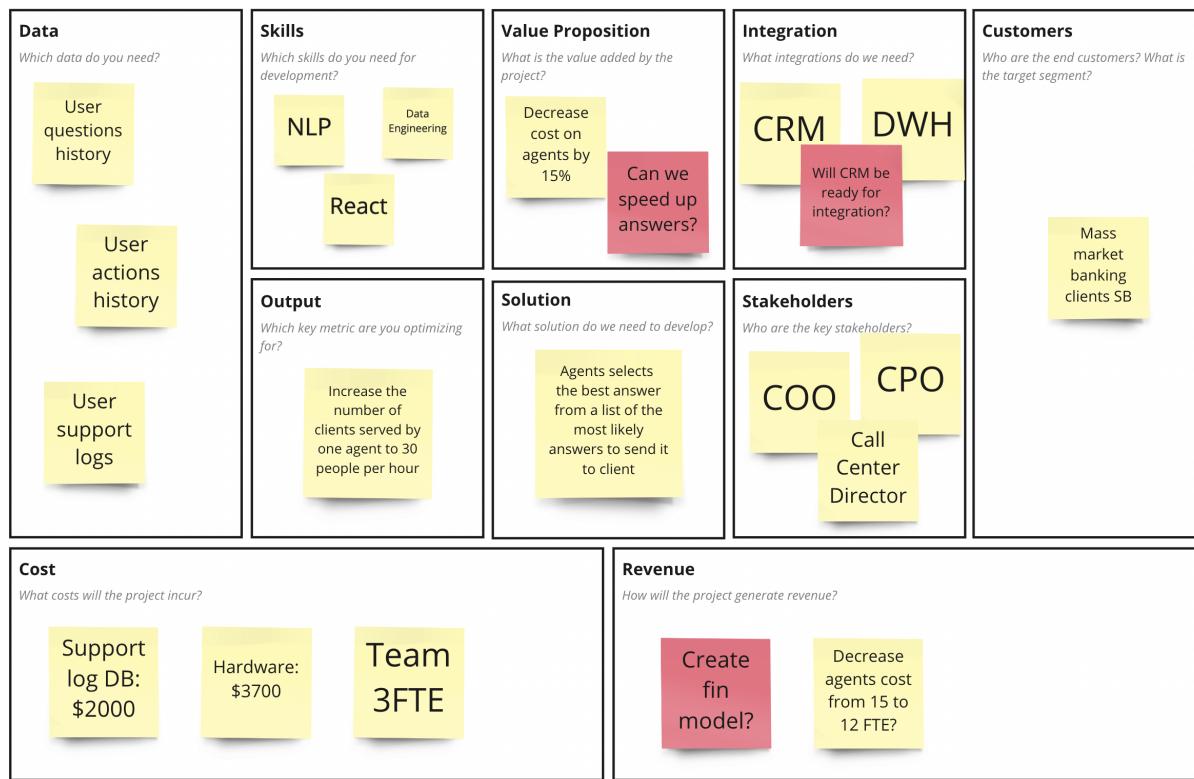


Рис. 4: AI Canvas Example

### Когда нужно использовать канвас

Внутри компании стартует новый проект или продукт и перед стартом нужно синхронизировать ожидания участников. Вы – консультант по AI и перед стартом работы с заказчиком нужно договорится о критериях успеха проекта

### Когда не нужно использовать канвас

Проект небольшой (скажем, доработка уже существующей модели) и нет смысла собирать отдельное совещание по каждой такой задаче. В этом случае вполне достаточно просто сформулировать продуктовую гипотезу

### Преимущества использования AI Canvas

- Канвас позволяет рано выявить завышенные ожидания со стороны бизнеса
- Не делаем проекты, которые даже теоретически не имеет смысла затевать
- Канвас позволяет совместно двигаться к согласованной бизнес цели
- Все члены команды понимают, какой результат может получиться с точки зрения бизнеса и сфокусированы на ней

## Преимущества использования AI Canvas по сравнению с ТЗ

Почему бы вместо канваса не написать традиционный документ — техническое задание для проекта? Пусть аналитик его напишет и со всеми согласует. Зачем все эти сложности и лишние встречи?

- Создание концепции проекта — брейншторминг и часто требует сфокусированной креативной совместной работы
- О концепции сложного проекта можно договорится за 1-2 встречи (скажем, одна-две недели), в то время как согласовывать сложный документ гораздо дольше
- У разных стейкхолдеров проекта могут оказаться противоречивые мнения, так что встреча в любом случае понадобится
- Формальное ТЗ при необходимости можно написать по итогам согласования AI Canvas

Идея использовать канвасы вместо документов изначально популяризована Алексом Остервальдером. Он использовал Business Model Canvas для создания бизнес-моделей организации. Изначально AI Project Canvas предложен и описан Jan Zawadsky в этой статье: <https://towardsdatascience.com/introducing-the-ai-project-canvas-e88e29eb7024>

## Story Mapping

### Какую проблему решаем

Вы сделали замечательную модель. Она подсказывает агенту наиболее вероятные ответы. Вы отдаете модель команде CRM и просите интегрировать ее в систему, которой пользуется call-центр. Коллеги забирают задачу и через несколько недель результат попадает в прод.

Однако результат вас расстраивает.

На подсказки никто не кликает. Они находятся за пределами видимости на экране, и их неудобно использовать. Но даже после перестановки выясняется, что ситуация не улучшается: агенты просто не понимают, зачем это нужно.

Кроме того, оказалось что CRM не поддерживает A/B тестирование и мы не можем честно измерить бизнес-эффект. Эта доработка возможна, но ее придется запланировать на следующий квартал.

Что пошло не так?

Конечная цель проекта — решенная проблема бизнеса или пользователя. Часто ML решение является составной частью какого-то большего продукта. Помимо собственно модели, такой продукт содержит и другие компоненты. Он может содержать интерфейс пользователя, должен взаимодействовать с другими системами и продуктами и быть вписанным в бизнес-процессы организации.

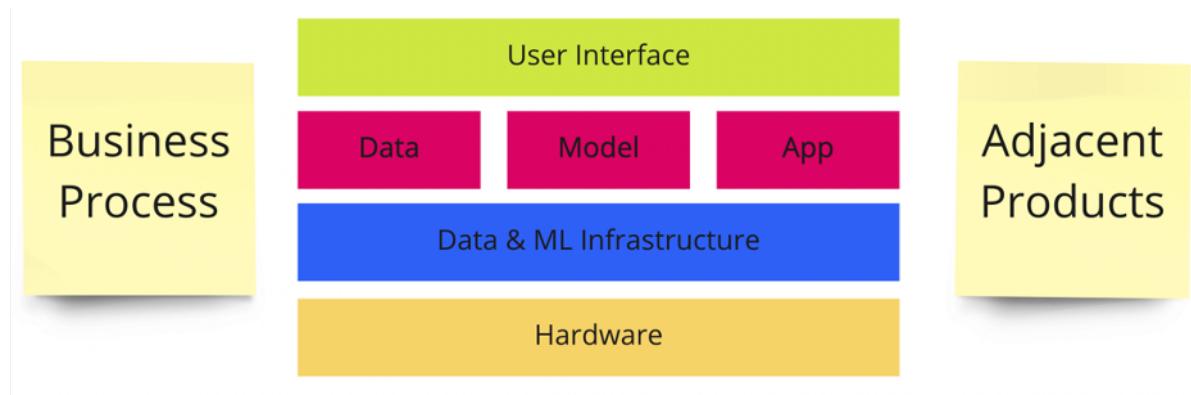


Рис. 5: ML в контексте

Бизнес-процесс, пользовательский интерфейс, взаимодействие с внешними продуктами нужно спроектировать. Конечным итогом такой работы должен стать бэклог продукта — план, содержащий задачи (такие как пользовательские истории) для всех команд, вовлеченных в создание продукта.

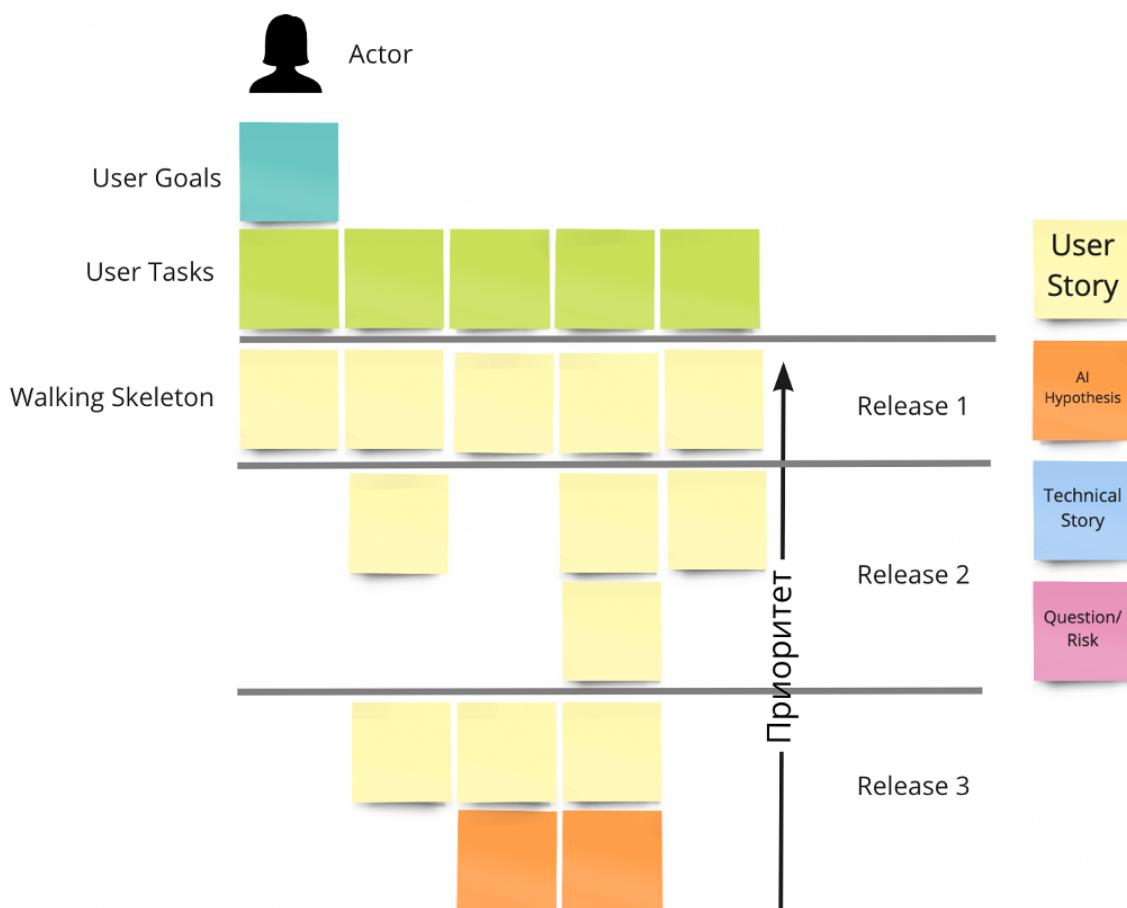
### Проектируем продукт при помощи User Story Map

Вы собираете всех вовлеченных людей на встречу по проектированию ML продукта.

Story Mapping –совещание по созданию карты проекта или продукта в виде набора пользовательских историй

Точно также, как и в случае AI Canvas, нам понадобится вовлечь всех, кто может дать полезную обратную связь по тому, как спроектировать итоговый продукт.

- Представители бизнеса (заказчики)
- Члены вашей команды
- Специалисты предметной области
- Представители смежных команд



**Рис. 6:** Структура Story Map

### Как создать Story Map

Создание Story Map проводится за одну или несколько встреч. Проходит оно в примерно в такой последовательности:

- **Создание ролей/персон (actors).** Напишите на стикерах одного или несколько пользователей (ролей/персон) продукта. Это наши действующие лица, именно они взаимодействуют с системой: конечные пользователи, модераторы систем, агенты и т.д.

- **Пользовательские цели.** Для ключевого актора напишите на стикерах несколько пользовательских целей.
- **Пользовательские задачи.** Для каждой цели напишите несколько пользовательских задач, описывающих как именно наш актор достигает своей цели. Можно относится к пользовательским задачам как шагам бизнес-процесса.
- **Walking Skeleton.** Под каждой пользовательской задачей добавьте стикер, каким самым простым образом можно изменить систему, чтобы достичь цели.
- **User Stories.** Под Walking Skeleton мы располагаем улучшения и дополнения функциональности нашего продукта.
- **Technical Tasks.** Добавьте на Story Map инфраструктурные и архитектурные работы
- **Questions.** Во время обсуждения могут возникнуть вопросы без ответа. Повесьте их в виде стикеров, они останутся как домашняя работа.
- **AI Hypothesis.** Проанализируйте, где именно применение ML может автоматизировать бизнес-процесс и повесьте по стикеру про каждую такую идею

## Пример Story Map

В итоге у вас должен получиться примерно такая Story Map:

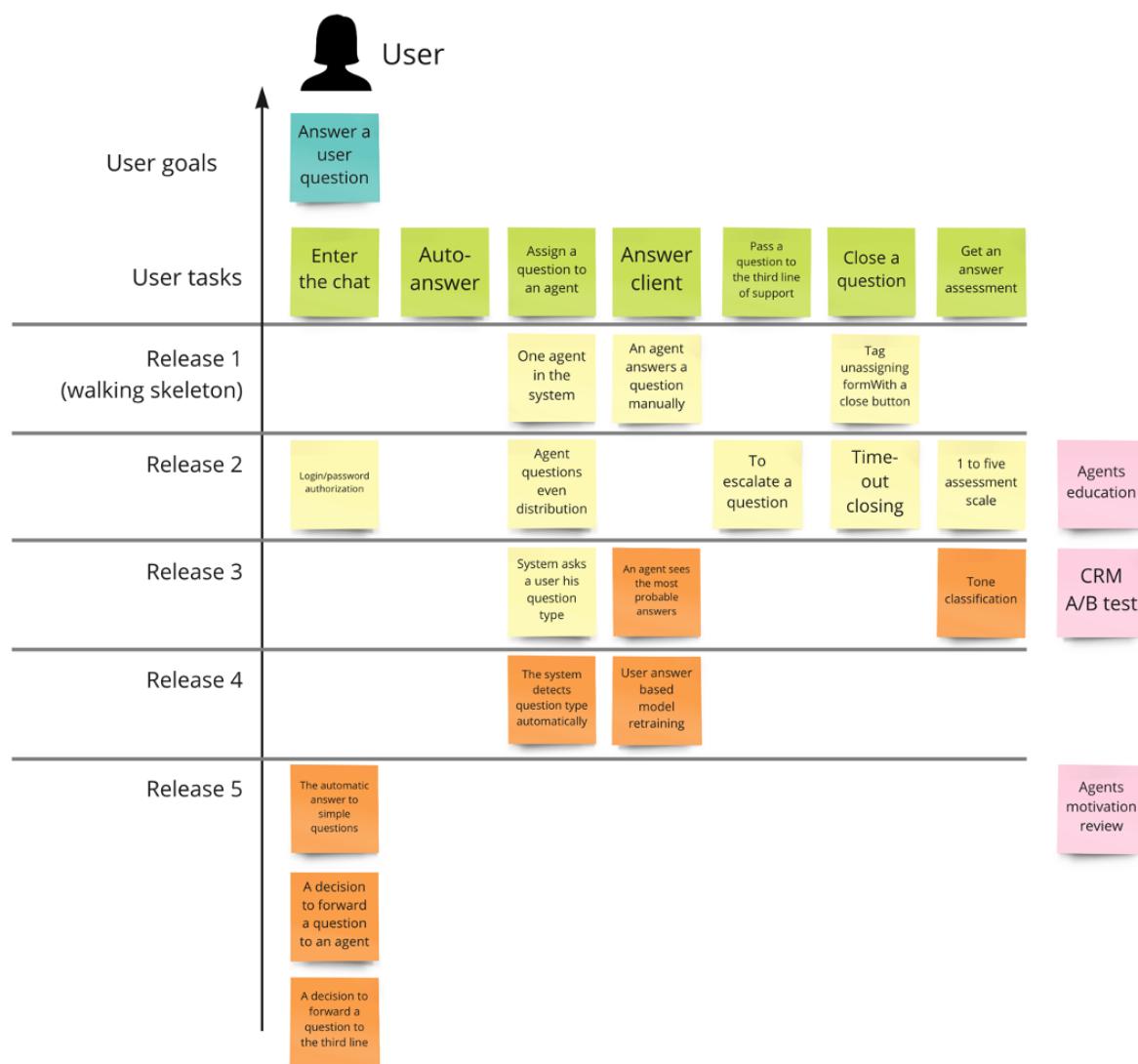


Рис. 7: Пример Story Map

Обратите внимание, что в этом примере мы разделили продукт на 5 релизов:

- Release 1. Walking Skeleton. Основа продукта, на которую мы будем нанизывать последующую функциональность
- Release 2. MVP без ML. Ручной ответ агентов в чате. Мы готовы выйти в прод на небольшой сегмент пользователей с этим набором функциональности
- Release 3. Подсказки агентам наиболее вероятных ответов при помощи ML
- Release 4. Автоматический роутинг (передача) агентам по их специализации
- Release 5. Автоматический ответ системы на простые вопросы

Кроме этого, мы нашли 3 риска, и должны будем их как-то адресовать в процессе проекта.

## Хинты создания Story Map

- Иногда бизнес-процесс оказывается нелинейным. Однако, нет смысла строить диаграммы из стикеров. Относитесь к Story Map как к истории, которую вы рассказываете

заинтересованным лицам.

- Иногда части бизнес-процесса повторяются (например, для разных акторов). Нет смысла их дублировать — для следующего актора вносите только изменения.
- Walking Skeleton это техническая основа для будущего продукта. Это не Minimum Viable Product, так как далеко не всегда его можно выпустить на реальных пользователей. Тем не менее большинство «граблей» мы соберем при его создании.
- Как только Walking Skeleton создан, мы имеем возможность инкрементально улучшать наш продукт. Приоритет располагается слева направо сверху вниз по слоям. Удобно прочертить предполагаемые релизы прямо на доске.
- Создание Story Map — важнейшая часть проектирования продукта и может занять значительно время — от нескольких часов до пары дней для крупного продукта.
- У некоторых важных стейкхолдеров нет столько времени. Сделайте ревью с ними вашего Story Map после его создания: совместно пробегитесь по нему, соберите обратную связь и внесите корректизы.
- Итоговый Story Map обязательно будет меняться по мере того, как вы строите ваш продукт и получаете обратную связь. При необходимости Story Mapping можно повторить (например, после каждого релиза).
- Наиболее важен ближайший релиз. Проектировать бэклог глубоко по слоям вниз — потеря времени. Ваше представление о том, как надо решать задачу поменяется.

## Когда использовать Story Map

- Проектирование всего продукта, где ML является важной, но не единственной его составляющей.
- Для анализа точек применения ML в организации. В этом случае Story Map представляет собой карту бизнес-процессов организации. Члены команды ищут точки, где ML может быть потенциально применен и принесет пользу.

## Когда нет смысла использовать Story Map

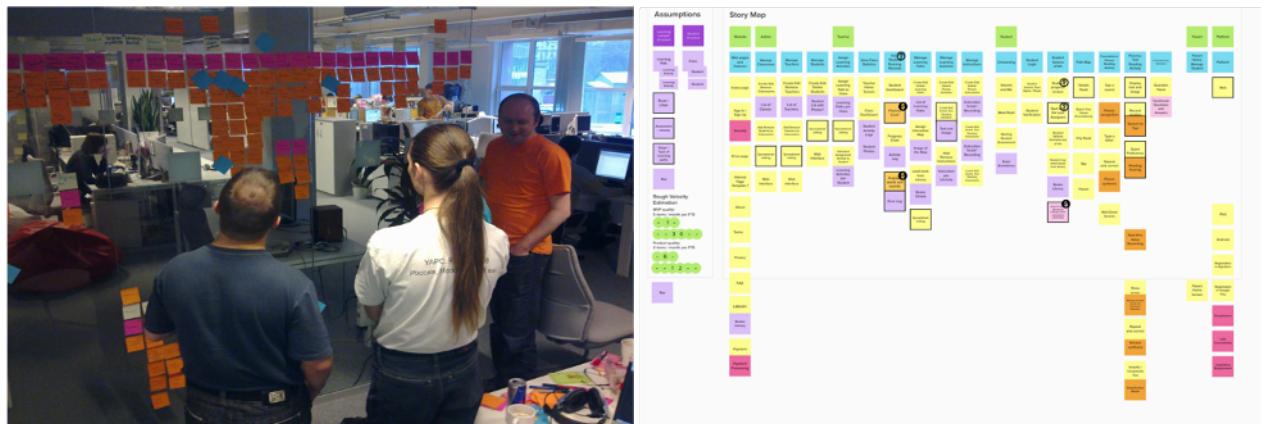
Когда задача команды — »прикрутить» ML к уже существующему и работающему бизнес-процессу и при условии, что сам бизнес-процесс сильно не изменится.

## Преимущества Story Mapping

- Проектирование продукта требует сфокусированной креативной совместной работы
- Планирование и проектирование важнее собственно плана. Вовлечение всех заинтересованных лиц в проектирование создают чувство ответственности за финальный результат
- Итогом Story Mapping является бэклог продукта. Если отслеживать прогресс по бэклогу (то есть отмечать, какие работы сделаны, а какие еще нет) можно получить прозрачную картинку по реальному прогрессу.

## Практическое использование Story Mapping

Story Map можно создать на очной встрече при помощи стикеров разных цветов. Можно сбраться на распределенную встречу и воспользоваться одним из инструментов совместной распределенной работы: miro.com, mural.co и др.



## About Story Mapping

Поход был предложен Джейфом Паттоном и популяризирован в его книге *User Story Mapping*<sup>a</sup>. Он оказался удивительно удобным для проектирования в том числе ML продуктов. Единственное изменение, которое мы внесли в подход — добавили шаг по генерации AI гипотез.

<sup>a</sup><https://www.amazon.com/User-Story-Mapping-Discover-Product/dp/B08TZGKKF2>

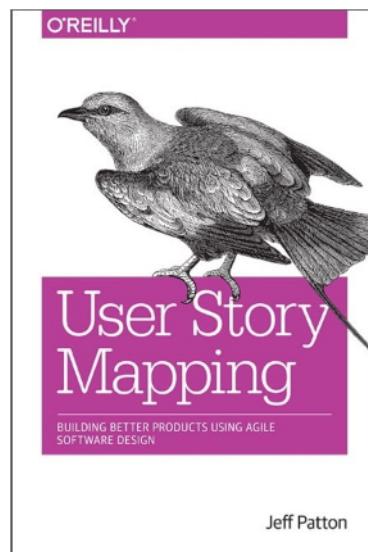
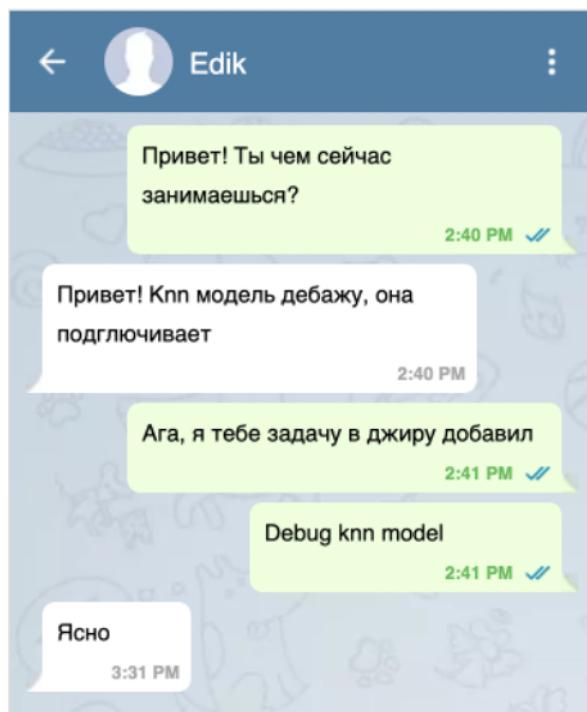


Рис. 8: User Story Mapping

## Гипотезы

### Какую проблему решаем

Сайентист Эдик уже неделю занимается новой моделью. Он засиживается допоздна и в целом выглядит очень занятым. «Надо бы ему задачу поставить» — думаете вы.



**Рис. 9:** Chat with DS

Давайте поговорим о том, как должны выглядеть задачи для сайентистов.

На примере снизу две типичных задачи. Примерно в такой формулировке их часто можно найти в тасктрекере.

Исследовать диалоговые системы для чатбота

Debug kNN model.

### Свойства традиционной постановки

- Предполагается, что исполнитель уже в курсе всех деталей
- Из формулировки задачи неясна ее бизнес ценность
- Из формулировки задачи неясны критерии ее окончания
- Задачу невозможно протестировать
- Задачу невозможно оценить
- Задачу невозможно приоритизировать
- Задача непонятна остальной команде
- Задача непонятна бизнесу и заинтересованным лицам

## К каким проблемам может привести традиционная постановка задач

- Встречи с командой непрозрачны для заинтересованных лиц и заказчиков. Они не понимают, чем занимаются члены команды.
- Члены команды закапываются в задачи. На пути решения они сталкиваются с новыми и новыми проблемами, увлекаются интересными, но неважными направлениями работ и в итоге задача может занять неограниченное время
- Иногда вместо решения проблемы «в лоб» заказчик или коллега могли бы предложить простой workaround, но из-за непрозрачности постановки это невозможно
- Ревью другим членом команды поверхностное, так как деталей постановки он не знает. В лучшем случае, другой член команды может обеспечить ревью качества кода, но не способа решения самой задачи

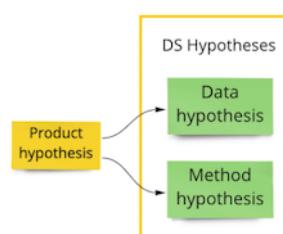


**Рис. 10:** Дата команда взаимодействует с заказчиками

### Три типа гипотез

Гораздо более эффективный способ ставить задачи — явным образом формулировать их в виде гипотез.

Мы рассмотрим три типа гипотез.



**Рис. 11:** Types of Hypotheses

- **Продуктовая гипотеза.** Предположение, что определенное изменение принесет ценность для клиента или бизнеса
- **Дата-гипотеза.** Предположение, что дополнительные данные улучшат метрику

- **Метод-гипотеза.** Предположение, что использование определенного метода улучшит метрику

## Продуктовая гипотеза

Продуктовая гипотеза разбивается (декомпозируется) на одну или несколько data- и метод-гипотез.

Для того, чтобы сформулировать продуктовую гипотезу, ответьте на 4 вопроса:

- Проблема. Какую проблему пользователя или бизнеса мы решаем?
- Решение. Какое изменение мы хотим сделать?
- На кого повлияет изменение?
- Метрика. Какой метрикой мы измерим результат?

Шаблон:

Мы полагаем, что [решили проблему] для [аудитории]. Для этого мы [сделаем изменения]. Мы окажемся правы, если [получим метрику]

Пример:

Мы полагаем, что [снизим затраты на агентов], [обслуживающих массовый сегмент]. Для этого мы [автоматически классифицируем клиентов]. Мы окажемся правы, если [увеличим количество клиентов, обслуживаемых одним агентом, которые остались довольны ответом агента на 10%]

Обратите внимание на словосочетание «мы полагаем». Оно явным образом говорит заказчику и всем заинтересованным лицам, что гипотеза не является обязательством, а процесс поставки ценности носит экспериментальный характер.

В формулировке продуктовой гипотезы мы не упоминаем способ достижения нашей цели.

Проекты это тоже продуктовые гипотезы

Ранее мы использовали AI Project Canvas для того, чтобы описать концепцию проекта. Ее можно коротко описать в виде продуктовой гипотезы.

Обратите внимание, что гипотеза из примера выше соответствует концепции проекта из главы об AI Project Canvas.

## Формулирование data- и метод- гипотез

Совместно с Эдиком вы сформулировали продуктовую гипотезу. Она имеет бизнес ценность, понятна заказчикам и команде, ее можно экспериментально валидировать. Есть одна маленькая проблема – проверка гипотезы займет слишком много времени: возможно, недели или даже месяцы работы.

Нужно уметь разбивать такую большую работу на относительно небольшие гипотезы не более человека-недели по трудозатратам. Тогда прогресс будет понятен и прозрачен.

Шаблон DS-гипотезы:

Мы полагаем, что [используя данные] и [метод обучения] мы получим [метрику] не менее [значения]

Пример DS-гипотезы:

Мы полагаем, что на кластеризованных вопросах пользователя и ответах агента мы можем построить линейную модель классификации для выбора подсказки. Это позволит получить Top 5 Accuracy не менее 50%

## Правила создания DS-гипотезы

- Каждая DS-гипотеза связана со своей родительской продуктовой гипотезой. Они описывают предполагаемый способ достижения цели, указанной в этой продуктовой гипотезе
- DS гипотезы бывают двух типов: дата- и метод- гипотезы
- Каждая дата/метод гипотеза содержит атомарное улучшение модели. Мы либо добавляем новые данные, либо улучшаем метод. \*\* Нужно стараться избегать ситуации, когда мы одновременно улучшаем данные и метод. Тогда если даже мы получим улучшение, то не сможем выяснить, с чем именно оно связано.
- При декомпозиции лучше двигаться итеративно и инкрементально, от простого к сложному. Мы начинаем с простых дата/метод гипотез, которые могут обеспечить baseline и постепенно усложняем, добавляя новые данные и методы.

## Приемочные критерии (Acceptance Criteria)

Для каждой получившейся гипотезы нужно сформулировать Приемочные критерии.

Приемочные критерии – это чек-лист условий, которым должна удовлетворять работа.

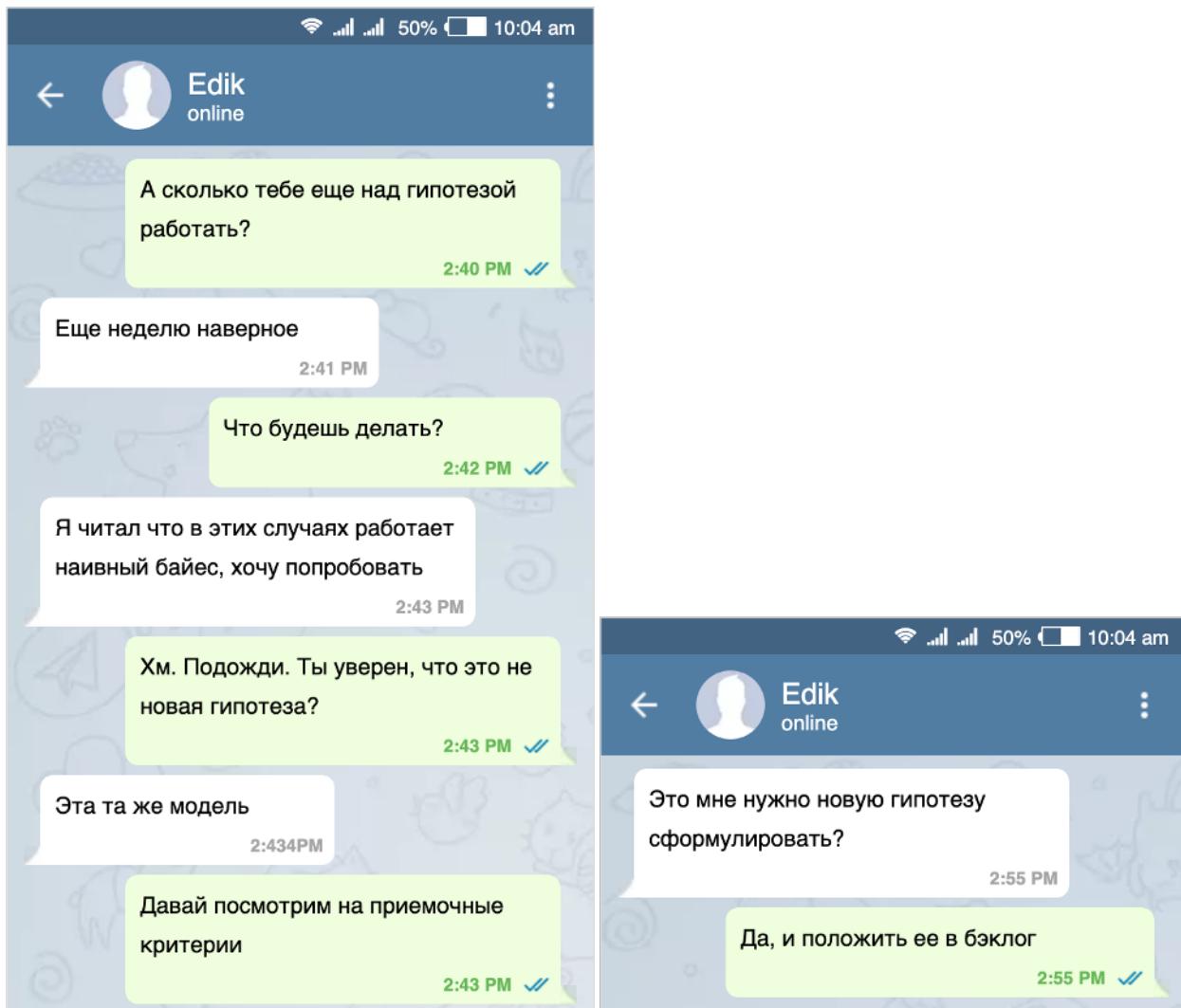
Приемочные критерии к гипотезе создает команда на этапе анализа (до реализации). Это позволяет обсудить в команде и согласовать способ достижения цели.

## Пример приемочных критериев

Мы полагаем, что на кластеризованных вопросах пользователя и ответах агента мы можем построить линейную модель классификации для выбора подсказки. Это позволит получить Top 5 Accuracy не менее 50%

- Пары вопросов-ответов кластеризованы
- Выбран критерий количества кластеров. Выборка кластеризована на это количество кластеров
- Метрики посчитаны на отложенной выборке и приложены к тикету
- Проведено ревью кода
- Код влит в репозиторий

## Новая гипотеза



Новая гипотеза:

Мы полагаем, что на кластеризованных вопросах пользователя и ответах агента мы можем использовать SVM с линейным ядром для выбора подсказки. Это позволит получить Top 5 Accuracy не менее 50%

Эдик работает над новой гипотезой. Работа должна быть уже закончена, но Эдик просит еще одну неделю.

В чем причина задержки? Не удается получить нужные метрики и Эдик хочет попробовать еще один способ.

Вы обсуждаете формулировку гипотезы и приемочные критерии и соглашаетесь, что это новый метод, а значит новая метод-гипотеза.

Не нужно бросаться ее делать. Ее имеет смысл сформулировать и отложить в бэклог.

Я и в мыслях не допускаю, что Эдик увлекся интересной статьей и бросил работу по линейной модели. Эдик не такой! Но что, если он ошибся, и она, все-таки, может дать хороший результат? Надо доделать все работы по ней: досчитать метрики, сформулировать файндинги, отправить код в репозиторий, провести код реview и т.д.

Новая гипотеза в бэклоге будет в свое время обсуждена командой, по ней будут сформированы приемочные критерии. Она пройдет по своему жизненному циклу.

## Преимущества использования гипотез

- Формулировка гипотезы фокусирует на достижении результата
- Работы по четко сформулированным гипотезам более предсказуемы, их можно оценить
- Доведение гипотез до конца позволяет четко формулировать итоги экспериментов, делать меньше ошибок и вести прозрачную документацию
- Процесс становится прозрачнее для всех заинтересованных лиц
- Процесс генерации идей и гипотез становится командной работой
- Результат работы члена команды можно протестировать
- Процесс работы над одним направлением (продуктовой гипотезой) можно распараллелить по членам команды, разбив на несколько дата- и метод- гипотез.

## Когда не надо формулировать гипотезы

- Если продуктовая гипотеза небольшая (менее человека-недели), ее можно не разбивать на дата- и метод- гипотезы
- Если для проверки дата- или метод- гипотезы требуется немного времени (скажем, меньше пары часов), команда может договорится не создавать отдельных гипотез

## Приоритизация ICE/RICE

### Какую проблему решаем

Бэклог заполнен интересными и ценными идеями (продуктовыми гипотезами). С какой из них имеет смысл начинать? Разумеется, с той, что принесет максимальное количество денег! Но как это определить?

Почему бы не спросить у бизнеса?

Вы приходите к самому важному руководителю и он методом пристального взгляда легко определяет ту, которая ему покажется максимально полезной. Предположим, он не ошибается и она действительно самая ценная. Действительно ли с нее нужно начинать?

Вообще говоря нет! Чтобы принять решение о приоритете, нужно знать еще и трудозатраты на реализацию гипотезы.

Предположим, наиболее ценная гипотеза способна принести \$1 млн, а следующие три только по \$0.5 млн. Пусть трудозатраты на реализацию первой – 9 месяцев, а для каждой из трех других по три месяца. Получается, за 9 месяцев работы мы могли бы сделать три простых гипотезы и заработать \$1.5 млн, а вместо этого потратим их на первую гипотезу и заработаем только \$1 млн.

Давайте теперь вспомним, что все идеи из нашего бэклога – гипотезы, и мы вообще можем не суметь их реализовать. Нам нужно каким-то образом учесть вероятность достижения успеха.

### Приоритизация методом ICE

Для гипотез из нашего бэклога нужно вычислить Score по следующей формуле:

$$\text{Score} = \frac{\text{Impact} * \text{Confidence}}{\text{Effort}}$$

Рис. 12: ICE Formula

Чем выше получившийся Score, тем выше приоритет.

Давайте теперь рассмотрим множители нашей формулы.

#### Impact

Impact – это мера влияния на итоговый результат с точки зрения бизнеса.

Очень часто мы не можем посчитать ее в конкретных деньгах – рублях или долларах:

- Недостаточно данных для более-менее точных расчетов
- Гипотеза сама по себе ценности не несет, но позволит открыть дорогу другим ценным гипотезам

- Ценность для бизнеса может быть не связана напрямую с доходом (например, цели связаны с захватом новых рынков или обеспечением требований со стороны регулятора)

Однако мы неплохо можем оценить ценность одной гипотезы относительно другой.

Категория	Impact (value points)
Massive	3
High	2
Medium	1
Low	0.5
Very low	0.25

Поэтому удобно оценивать ценность в относительных единицах.

Мы обычно используем набор фиксированных категорий с указанными оценками Impact как в таблице справа. Мы выбираем из бэклога среднюю по влиянию гипотезу в качестве эталонной. Она становится Medium и ее Impact равен 1. Если другая гипотеза в два раза более ценная, то ее Impact равен 2. Если в два раза менее ценная, чем эталонная, то 0.5. Если гипотеза кажется очень ценной, ее Impact может достигать трех. И так далее.

## Effort

Effort – оценка трудозатрат на реализацию гипотезы.

Посчитать трудозатраты в человеко-часах или днях просто невозможно:

- Данных недостаточно для точного расчета
- Мы не знаем пока, с какими трудностями мы столкнемся по ходу реализации
- Трудозатраты окажутся разными для разных исполнителей: оценка опытного сайентиста будет отличаться в разы от оценки джуна

Точно также, как и в случае Impact, воспользуемся оценкой в относительных единицах. В литературе их называют Story Points.

Часто используют прогрессивные шкалы, например последовательность Фибоначчи или степеней двойки, чтобы не спорить, например, какова сложность гипотезы – 5 или 6 пойнтов.

Обычно выбирают одну простую в реализации гипотезу и обозначают ее за 1 SP (story points). Если гипотеза кажется в два раза сложнее эталонной, то ее оценка будет равна 2 SP. И так далее.

## Confidence

Confidence – это оценка вероятности (уверенности) в том, что гипотеза может быть реализована и окажет нужный бизнес эффект

Удобнее всего confidence измерять в процентах.

Очевидно, что confidence не может превышать 100%. Опять-таки, лучше использовать набор фиксированных категорий с соответствующими оценками Confidence.

Category	Confidence
High	100%
Medium	80%
Low	50%
Very low	10%

## Как провести сессию по приоритизации гипотез

У каждого из заинтересованных лиц и членов команды может быть свое представление о степени влияния на бизнес, трудозатратах и уверенности в результате для гипотез из бэклога. Лучше всего провести совместную сессию и согласовать свои представления.

Один из интересных вариантов – использовать Planning Poker. Давайте рассмотрим это на примере оценки трудозатрат.



Рис. 13: Planning Poker Cards

- Сначала участники совместно выбирают эталонную гипотезу. Она должна быть простой и понятной. Мы обозначаем ее за 1 Story Point.
- Каждый участник получает колоду карт с последовательностью Фибоначчи
- Ведущий представляет следующую гипотезу
- Участники выбирают из своей колоды предполагаемую оценку (во сколько раз гипотеза больше эталонной) и выкладывают карту рубашкой вверх
- По команде ведущего карты вскрываются
- Те, кто выкинул максимальную и минимальную оценку, аргументируют их
- Процедура повторяется, пока мы не сойдемся достаточно близко.

- Итоговое значение записывается, мыдвигаемся к следующей гипотезе

Аналогичным образом можно оценивать и Impact и Confidence. После этого мы считаем скоры и получаем финальную оценку приоритета.

## Приоритизация при помощи RICE

В некоторых случаях удобно разделить Impact на два множителя — Reach и Impact. Тогда формула будет выглядеть так:

$$\text{Score} = \frac{\text{Reach} * \text{Impact} * \text{Confidence}}{\text{Effort}}$$

Рис. 14: RICE

Reach — оценка охвата гипотезы.

В зависимости от контекста конкретного бизнеса, она может быть выражена в количестве пользователей, заказах, транзакциях и так далее.

Impact — мера влияния на итоговый бизнес-результат для одной единицы в reach.

Например, вы продаете Data Products банкам. Тогда Impact — оценка влияния гипотезы на конкретный банк. Например, одна гипотеза может быть очень нужна условному усредненному банку и ее Impact = 3. Ценность другой гипотезы равна 1. Однако рынок (пропорциональный количеству банков, способных потенциально заинтересоваться соответствующим продуктом) для первой гипотезы большой и равен 3, а для второй гипотезы рынок в три раза меньше и равен 1.

## Особенности проведения приоритизации

- Приглашайте DS команду участвовать в сессии приоритизации. Обсуждение Impact и Confidence заинтересованных лиц дает им огромное количество контекста, тем самым вовлекает в работу и помогает принимать решения лучше
- Члены DS команды могут не хотеть участвовать. Объясните, что без оценки Effort приоритизация будет ошибочной, а оценку трудозатрат могут сделать только они
- Оценка Score не является абсолютно верной и финальной. Постфактум можно убедиться, что мы часто ошибаемся в оценках и, как следствие, приоритетах. Однако наличие правил приоритизации позволяет выигрывать в среднем, на большом количестве гипотез
- В оценке трудозатрат участвуют только непосредственные исполнители то есть те, кто будет делать работу руками. Обычно у представителей бизнеса слишком заниженные оценки трудозатрат

## Как развивать подход к приоритизации

В вашей организации подход к приоритизации будет развиваться со временем. Вот несколько идей, как и куда его можно улучшать:

- Confidence можно разделить отдельно на два множителя — вероятность успеха с точки зрения бизнеса и техническую реализуемость
- Шкалу Confidence можно стандартизировать с помощью набора критериев. Например, наличие аналогичного кейса в вашей стране, проведенный А/В-тест, посчитанная бизнес модель и т.д. повышает уверенность
- Точно также можно ввести шкалу критериев для Impact

В процесс приоритизации может вмешаться высокое руководство и внезапно повысить приоритет каких-то гипотез и тем самым обесценить вклад участников процесса.

Очень хорошо работает приглашение руководства на следующую сессию приоритизации.

Волонтеристски изменить приоритет будет невозможно. Вместо этого они будут стараться убедить в более высоком импакте или уверенности.

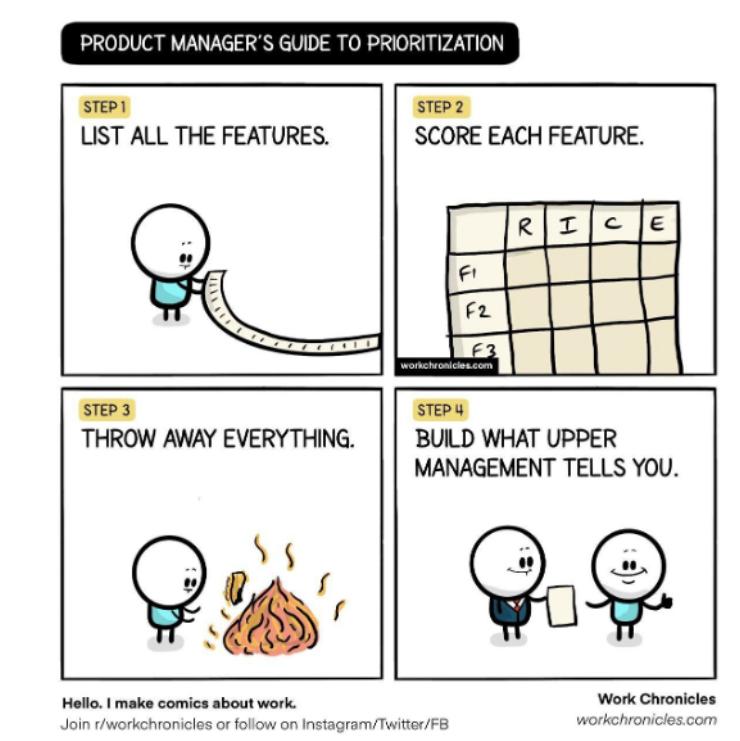


Рис. 15: Prioritization

## Зачем нужна приоритизация

Давайте перечислим несколько преимуществ приоритизации

- Позволяет максимизировать доход для бизнеса
- Отсекает ненужные и малоперспективные работы
- Позволяет договариваться различным заинтересованным лицам о приоритетах

## Когда не надо использовать ICE/RICE

- ICE/RICE предназначен для приоритизации исключительно продуктовых гипотез. Дата- и метод- гипотезы не имеют отдельной от продуктовой гипотезы ценности. Подход к приоритизации таких гипотез описан в главе о декомпозиции продуктовых гипотез
- Если у вашей команды небольшое количество продуктовых гипотез (скажем, 1-3), использовать сложный метод приоритизации, очевидно, не имеет смысла.

## Когда использовать ICE/RICE

- На уровне команды — если у вашей команды много продуктовых гипотез в бэклоге
- На уровне организации — можно иметь один общий бэклог из продуктовых гипотез на нескольких команд и приоритизировать его перед передачей конкретной команде на реализацию

Подход ICE был предложен Шоном Эллисом (Sean Ellis), автором термина growth hacking

## Декомпозиция методом «Мерседеса»

В главе про Story Map мы говорили о планировании всего софтверного продукта, где ML-продукт является его составляющей. В этой главе мы поговорим о планировании самого ML-продукта. Можно сказать, речь пойдет о создании бэклога для AI-проекта или большой продуктовой гипотезы.

### Зачем нужно планирование

Часто говорят, что создавать планы исследовательских проектов не имеет смысла. Появление новой информации приводит к тому, что с таким трудом созданный план теряет ценность и инженеры разочаровываются в самом процессе планирования.

Давайте разберемся, почему планирование важно и заодно поговорим, каким должно быть правильное планирование.

### Правильный учет рисков

Создание модели может занять долгое время — несколько месяцев. По-настоящему проверить эффективность модели мы сможем ближе к концу ее жизненного цикла, с началом интеграции с другими продуктами, и даже позже — на реальных пользователях и в боевом режиме.

Представьте, что вы долго создавали модель и она даже показывает приемлемые результаты. И вдруг с началом внедрения оказывается, что данные, на которые вы рассчитывали, в проде недоступны и модель не может принести пользу.

Возможно, вы даже интегрировали модель и она показывает разумные результаты на A/B-тесте, однако ревью со стороны комплайенса запрещает вам использовать эти данные из-за юридических проблем.

Конечно, мы могли бы подумать об этих рисках заранее, если бы догадались спросить себя об этом. В этом случае мы бы сэкономили очень много времени.

План должен подсвечивать риски и управлять ими.



Рис. 16: Суть планирования

Все команда должна участвовать в планировании

Бывает, что за планирование отвечает DS Lead. Это самый опытный член команды. Он садится, крепко задумывается и пишет план работ. Части плана попадают членам команды уже в виде заданий.



Рис. 17: Роль DS лиде

На самом деле в планировании должны участвовать все члены команды. Вырастет качество решений. У участников появляется общий контекст: понимание целей, методов, ограничений и рисков.

В планировании должна участвовать вся команда

План должен обеспечивать итеративность и инкрементальность

План должен быть инкрементальным. Мы начинаем с простых и быстрых способов достижения цели и двигаемся к все более сложным.

Иногда команды пренебрегают этим правилом по разным причинам. Например, хочется сделать что-то интересное, а не возиться с простыми методами. Возможно, на более сложном подходе настаивает заказчик.

И, все-таки, начать надо с создания простой модели.

Это позволяет получить начальный baseline – точку отчета, с которой мы будем сравнивать дальнейшие улучшения. Как правило, бейзлайн удается получить быстро, показать заинтересованным лицам и получить какую-то обратную связь. Даже такую начальную модель имеет смысл начать автоматизировать. Это позволит на практике отловить множество технических рисков интеграции продукта.

План должен обеспечивать возможность его изменения

План проекта должен допускать гибкость. При появлении новой информации план можно легко перестроить.

План должен фокусироваться на валидации гипотез

Начинать реализацию имеет смысл с наиболее рискованной гипотезы, способной убить проект

## Планирование методом Мерседеса

Давайте рассмотрим процедуру планирования ML-продукта подходом, который мы назвали методом «Мерседеса». Она обеспечивает совместное создание итеративного инкрементального плана, учитывая ключевые риски.

Планирование методом Мерседеса состоит из 4 шагов:

1. Брейншторминг гипотез
2. Выделение Proof of Concept/MVP
3. Анализ рисков по чек-листам

В сессии планирования участвует вся команда:

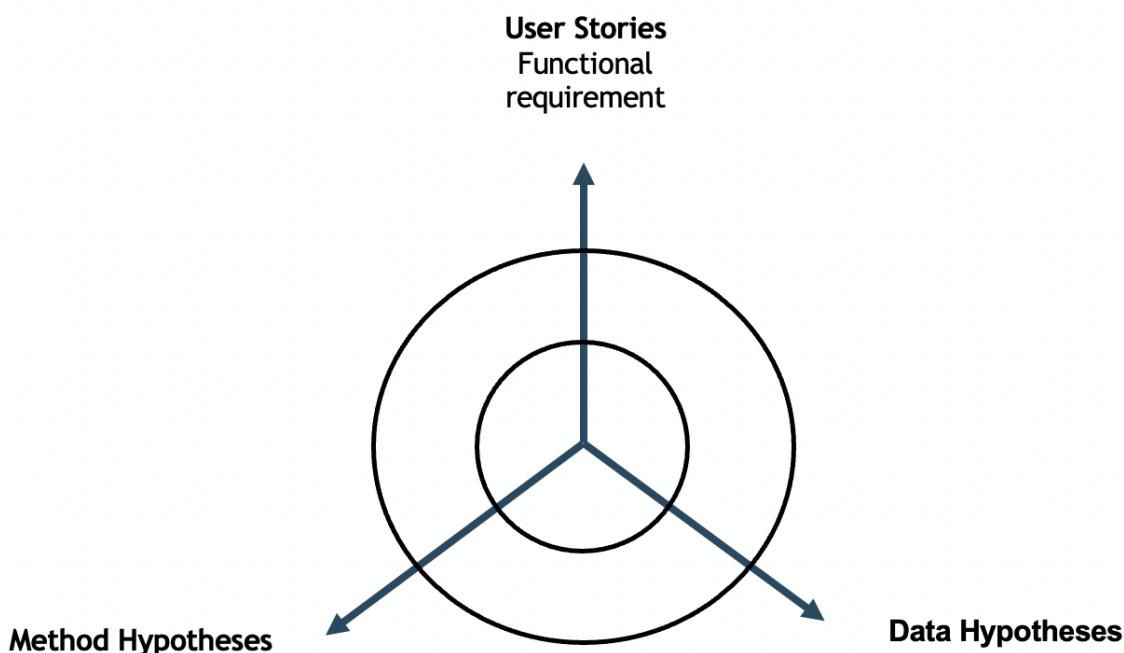
- Члены вашей команды
- Специалисты предметной области

При необходимости подключаются и другие заинтересованные лица:

- Представители смежных команд и служб
- Представители бизнеса (заказчики)

## Подготовка к планированию

Подготовьте для сессии три оси, как показано на рисунке.



**Рис. 18:** Decomposition Axes

### Шаг 1. Брейншторминг гипотез

Совместно с командой наклейте идеи гипотез вдоль трех осей:

- **Method Hypothesis.** Вдоль оси Метод мы располагаем Метод-гипотезы, начиная с самой простой. Например, можно начать с эвристического подхода, затем расположить линейные методы, и лишь потом уже нейронные сети.
- **Data Hypothesis.** Вдоль оси Дата мы располагаем Дата-гипотезы. Начинаем с данных, которые доступны уже сейчас и добавляем новые источники в порядке сложности получения данных.
- **User Stories.** Вдоль оси User Stories мы располагаем пользовательские истории, то есть софтверные работы, которые нужно сделать для достижения финального результата. Их также нужно расположить от простого к сложному.
- **Questions.** В процесс брейншторминга у вас обязательно будут сложности. Например, вы можете не понимать, какие данные доступны или какие интеграции потребуются. Подклейте стикер с соответствующим вопросом. Обычно мы выделяем такие вопросы другим цветом.
- **Technical Tasks.** Инфраструктурные и архитектурные работы, например интеграция модели с бэкендом, развернуть HDFS, создать инфраструктуру запуска модели в проде

В результате должна получиться примерно такая картинка:

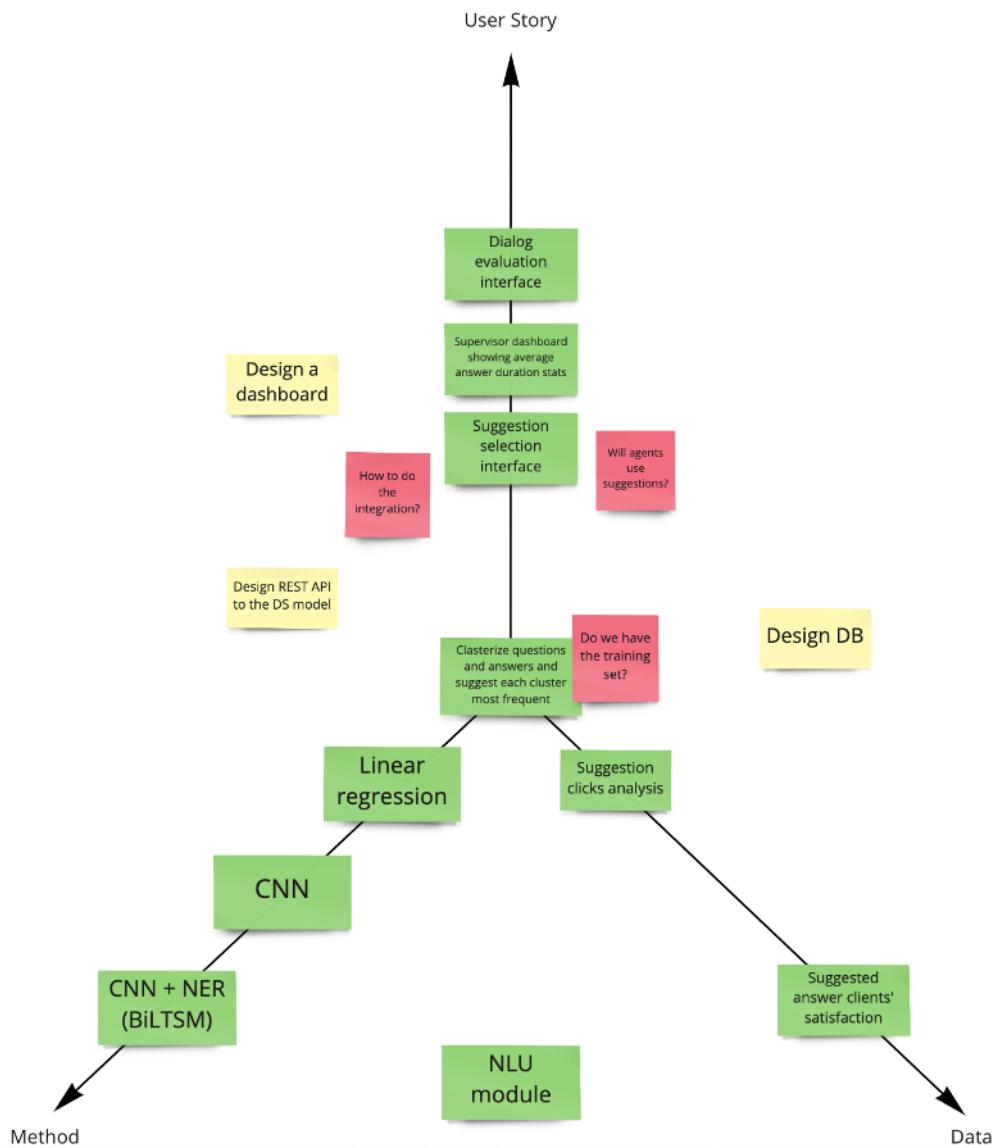


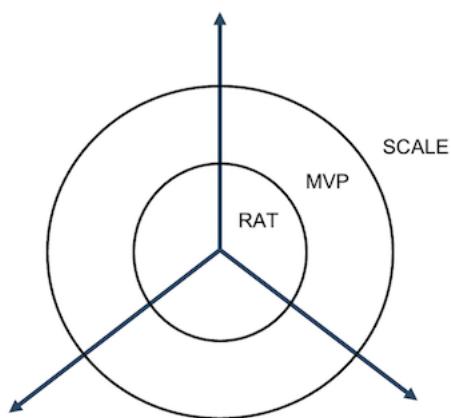
Рис. 19: Пример декомпозиции

## Шаг 2. Выделение Proof of Concept/MVP

К окончанию предыдущего этапа у вас может получиться довольно много гипотез.

Давайте вспомним, что главной задачей является максимально быстрая валидация продуктовой гипотезы. Нам нужно понять, что является более приоритетным, а что можно сделать чуть позже.

Визуально разделяем двумя концентрическими кругами на три области, как показано на рисунке



**Рис. 20:** Области

- **RAT (Riskiest Assumption Test).** Самые рискованные гипотезы, которые можно проверить еще до выкатки на конечных пользователей и убить продуктовую гипотезу.
- **MVP (Minimum Viable Product).** Минимальный продукт, который можно валидировать на конечном пользователе и проверить гипотезу
- **SCALE.** Дополнительные улучшения и доработки, которые можно отложить на более поздние этапы создания продукта.

Обсудите и распределите ваши гипотезы и пользовательские истории по этим трем категориям. Страйтесь минимизировать RAT и MVP.

Чем ближе к центру, тем выше приоритет гипотезы.

### Шаг 3. Анализ рисков по чек-листам

Управление рисками означает, что на раннем этапе проекта можно провести анализ потенциальных проблем, которые могут ожидать нас в будущем.

В сообществе LeanDS мы собрали набор типичных проблем в виде набора вопросов, которые команда может себе задать.

Пройдите по списку вопросов вместе с командой и попробуйте ответить на каждый из них. Возможны такие варианты:

1. У вас есть ответ на этот вопрос. Двигаемся дальше
2. Вопрос не релевантен к вашей ситуации. Двигаемся дальше
3. Вопрос релевантен, у вас нет хорошего ответа. Повесьте стикер с соответствующей проблемой.

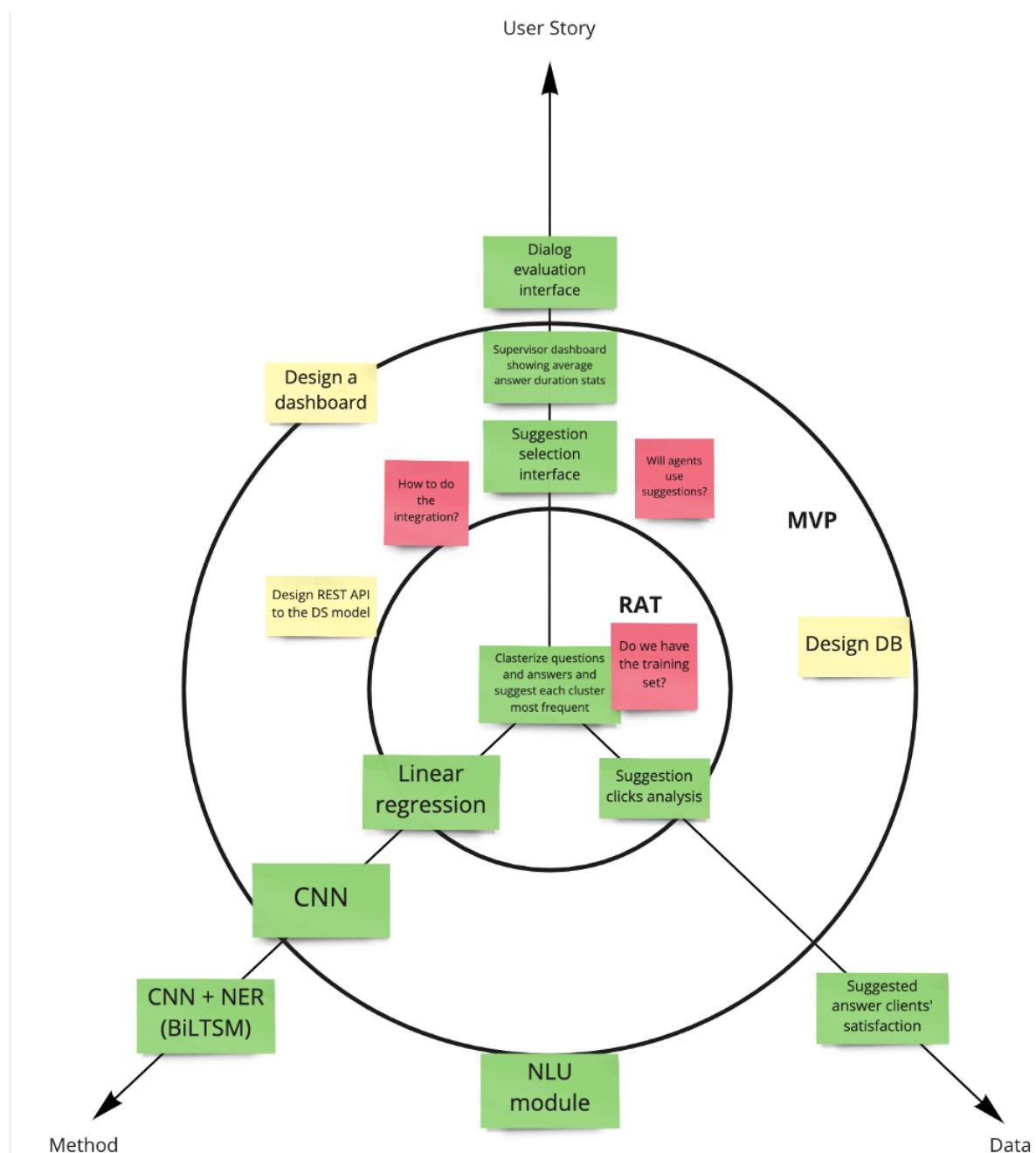


Рис. 21: Использование вопросов

### Особенности использования подхода «Мерседес»

- Итогом сессии является набор заголовков гипотез. После сессии нужно оформить в них в виде гипотез («мы полагаем...») и положить их в бэклог продукта
- По мере продвижения к плану работ нужно возвращаться и проводить ревью и апдейт плана
- По длительности «Мерседес» обычно занимает около часа
- Риски могут быть также частью бэклога, если требуют вовлечения со стороны команды.
- Состав вопросов можно пополнить типичными проблемами, характерными для вашего рынка и области применения ML

## Преимущества использования

- Команда и эксперты предметной области совместно придумывает план и разделяет веру в возможность достижения результата
- При появлении проблем команда понимает, как это влияет на итоговый план и может предлагать качественные идеи по изменению плана
- У команды есть список рисков. Если заняться ими достаточно рано, их можно избежать или ослабить их влияние.
- План представляет собой прозрачную картину достижения целей, понятную заинтересованным лицам
- Благодаря использованию RAT/MVP план сфокусирован на быстром тестировании гипотез
- Наличие множества гипотез в плане позволяет распараллелить работу по ним и ускорить создание продукта

Эта книга – первое письменное изложение идеи декомпозиции ML проектов методом Мерседеса в истории человечества

## Вопросы (Questions)

### Story questions

1. Как будет работать сервис, когда модель выйдет в прод?
2. Какие технические ограничения существуют для сервиса?
3. Какие системы и команды зависят от сервиса?
4. Как проводить валидацию модели в проде (А/Б-тест, ...)? Как валидация должна быть устроена?
5. Что будет, если сервис откажет?
6. Насколько быстро меняются данные, Как часто нужно повторно обучать модель?
7. Как будет производится мониторинг работы модели и сервиса в проде?

### Data questions

1. Достаточно ли данных для обучения модели?
2. Есть ли в данных для обучения предсказательная сила для решаемой задачи?
3. Качество данных достаточно для использования в модели?
4. Есть ли разметка?
5. Можно ли купить дополнительные данные?
6. Можно ли использовать публично доступные данные?
7. Заражены ли данные предрассудками (biased)?
8. Как модель будет получать данные в проде?
9. Совпадают ли данные в трейне и проде?
10. Как контролировать качество данных в проде?
11. Как сохранить качество данных после выхода в прод?
12. Какие юридические риски связаны с использованием данных?
13. Есть ли какие-то требования к данным, связанные с законодательным регулированием?

### Method questions

1. Есть ли SOTA (State of the Art) для задачи? Какие у нее значения мер качества?
2. Как эту работу делают эксперты (Subject Matter Expert) сейчас?
3. Требуется ли интерпретируемость модели?
4. Требуется ли внешняя независимая валидация модели?

### Примеры использования вопросов

Давайте рассмотрим несколько примеров применения этих вопросов при декомпозиции продуктовой гипотезы.

**Как будет работать сервис, когда модель выйдет в прод?**

Модель должна как-то взаимодействовать с окружением. Понимаем ли мы как модель будет интегрирована с существующими продуктами? Например, у нас есть пользовательская история “Интерфейс для выбора подсказки”. Такой интерфейс не может работать без того, чтобы текущая система обращалась к модели, передавала туда запрос и получала сгенерированные подсказки. Есть ли задача на такую интеграцию на доске? Если такой задачи нет, ее нужно добавить в виде Technical Task.

### Требуется ли интерпретируемость модели?

Если требуется интерпретируемость модели, то мы скорее всего не сможем использовать CNN или LSTM для реализации модели и нам нужно убрать эти гипотезы из декомпозиции и добавить вместо них другие, более интерпретируемые модели.

### Достаточно ли данных для обучения модели?

Если данных не достаточно, то вся наша декомпозиция теряет смысл! Нам прежде всего нужно ответить на этот вопрос. Это супер критический риск и мы поместим его в самый центр в RAT.

### Как эту работу делают эксперты (Subject Matter Expert) сейчас?

При декомпозиции мы набросали много довольно сложных моделей, подготовка которых займет заметное время. Знаем ли мы как решается задача сейчас? Может быть есть простой набор правил, которые хорошо работают и которые можно использовать как бейзлайн для нашей задачи.

### Можно ли использовать публично доступные данные?

В задачах работы с текстами могут помочь внешние публично доступные наборы данных. Их добавление в процессе обучения может заметно повысить качество работы. Знаем ли мы о таких наборах данных? Где их достать? Как использовать? Эти вопросы тоже можно добавить на доску.

## Кроссфункциональная команда

### Какую проблему решаем

Концепция продукта описана при помощи AI Project Canvas. Теперь пора стартовать разработку нового продукта. Хорошо, что в канвасе перечислены навыки членов команды. Вот кто нам понадобится:

- Data Scientist (DS) будет строить модель и заниматься данными
- Backend Engineer (BE) будет заниматься инфраструктурой и автоматизировать модель, как только она будет готова

Вы ставите задачу каждому из них.

Прошло пару месяцев. BE закончил с инфраструктурой, но модель пока не готова. Не простоявать же ему? Тут в соседнем проекте как раз есть срочная работа, пусть ей пока займется.

Внезапно DS получает хорошие результаты! Модель готова. Но наш BE уже на другом проекте и у него свои обязательства. Не простоявать же DS?

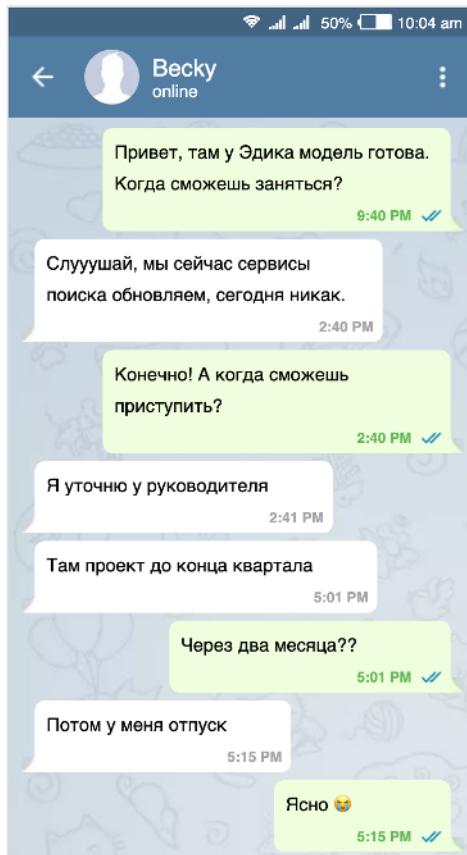


Рис. 22: Chat

BE освободился и приступил к автоматизации модели. У него есть несколько вопросов к DS – данные успели немного устареть и модель нужно переделать, но DS пока на другом проекте и заняться этим не может. Что ж! Пока можно BE переключить на новый проект.

Проходит какое-то время. Наконец проект завершен. Пора передавать его заказчику. Он был очень заинтересован в начале проекта, но ближе к концу охладел и у него нет времени принять результат.

И все-таки удается довести модель до продажи. К сожалению, она не показывает нужных бизнес-результатов. Кто будет отвечать за исправление и доработку?

Итак, такое разделение работ неэффективно и приводит к следующим проблемам:

- Постоянные переключения между проектами приводят к задержке реализации
- Удлинение сроков проекта приводит к тому, что участники успевают основательно забыть контекст проекта и тратят время на то, чтобы вспомнить его
- За время реализации актуальность проекта может упасть до нуля
- За время проекта успевают накопиться изменения, которые приводят к лишним переделкам
- Участники проекта могут уйти из компании или подразделения и вместе с ними уйдут знания по проекту
- Никто из команды не отвечает за итоговый бизнес-результат

## Кросс-функциональная команда

Вместо разделения фронта работ по участникам или по подкомандам, давайте создадим единую команду, отвечающую за конечный бизнес-результат. В нее должны войти все компетенции, необходимые для достижения этого результата. Такая команда называется кросс-функциональной.

Кросс-функциональная команда — группа людей с различной функциональной экспертизой, работающая на одну общую цель, с единым планом, имеющая взаимные обязательства и чувствующая ответственность по отношению друг другу

Вот свойства такой команды:

- Общая цель. Команда несет ответственность за достижение бизнес-результата, а не создание модели и вывод в продажу
- В команду (по возможности) включаются специалисты со всей экспертизой, необходимой для достижения цели
- Единый план (бэклог). Команда совместно определяет способ достижения этой цели: проектирует итоговый продукт и планирует работу
- Все члены команды имеют взаимные обязательства. Это означает, что они помогают друг другу, обмениваются работой и проводят реview результатов работы друг друга
- Общие встречи (планирования, стендапы, ретроспективы). Команда регулярно встречается и синхронизирует работу и придумывает способы ее улучшить
- Стабильный состав. Команда формируется на длительный срок, все участники команды выделены в нее на 100%.

## T-shape professional

Легко сказать – единая команда, которая отвечает за результат, но как это будет работать? Если в одной команде окажутся и DS и BE, хватит ли каждому работы? Чем будет заниматься BE, пока DS будет делать модель?

Члены команды смогут эффективно друг другу помогать, если мы уберем искусственные функциональные барьеры между ними.

Каждый участник команды должен быть T-shaped professional: глубоко разбираться в своей области и достаточно хорошо в смежных областях, чтобы суметь помочь другим членам команды.



Рис. 23: T-shape specialization

## Основные возражения против кросс-функциональности команд

Если вы до этого не видели, как работает такая кросс-функциональная команда, вы, наверное, серьезно обеспокоены. Весь ваш менеджерский опыт говорит, что такая конструкция в целом нежизнеспособна.

Давайте рассмотрим несколько типичных возражений:

### Backend Engineer не сможет сделать модель

Речь не идет о том, чтобы полностью на сто процентов поменяться работой. Мы говорим о возможности взаимопомощи. Сможет ли BE помочь DS сгенерировать какой-то замороченный признак (фичу)? Сделать автотест? Помочь с настройкой инфраструктуры? Если DS является узким звеном процесса, то любая помощь со стороны других инженеров сильно ускоряет общий процесс разработки и уменьшает сроки проекта.

### DS не захочет делать работу Backend Engineer

Просто спросите ваших сотрудников, готовы ли они помогать друг другу, чтобы быстрее добиться целей проекта. Никаких проблем вы не увидите. Люди, как минимум, не против командной работы и если даже у них не было похожего опыта, вполне готовы попробовать.

### Они математики / пришли писать код, и не мотивированы добиваться каких-то там бизнес-результатов

Наличие общей цели дает им чувство сопричастности к чему-то большему. Командная работа обеспечивает дружескую поддержку и помогает против синдрома самозванца. Ответственность за бизнес-результат увеличивает автономность в принятии решений, ведь теперь инженер свободен в выборе способа достижения цели. Все это приводит к увеличению мотивированности и вовлеченности членов команды.

### У нас много ML-продуктов. Нам просто не набрать столько команд!

Разница в том, что мы не формируем команды вокруг продуктов, а передаем продукты командам. Одна стабильная кросс-функциональная ML команда может отвечать за несколько ML-продуктов. Благодаря кросс-функциональности такая команда отвечает не просто за функционирование модели, но и за то, что она приносит нужный бизнес-результат, то есть за создание, поддержку и развитие модели. Стабильный состав команды обеспечивает долгий срок жизни продуктам.

### Как именно члены команды смогут обмениваться работами?

Действительно, синхронизация работы кросс-функциональной команды – дело не тривиальное. Нужен процесс, который позволит команде делать это эффективно. В более поздних главах мы рассмотрим Канбан, как один из способов организовывать взаимодействие в такой команде.

## Команда DS as a Service

Одиночная работа сайентиста над бизнес-задачей – путь к выгоранию. Даже если есть предпосылки к отказу от кросс-функциональной команды, я очень рекомендую создавать выделенные DS-команды, даже если они будут состоять исключительно из DS. Мы называем такие команды DS as a Service. Сайентисты в команде смогут помогать друг другу, делать реview кода, поддерживать в сложных ситуациях.

### Когда использовать кросс-функциональные команды не эффективно

Давайте посмотрим ситуации когда использование кросс-функциональных команд невозможно или неэффективно и имеет смысл создавать команды DS as a Service.

- Вы только набираете сайентистов в компанию. Возможно, сразу передать сайентиста в продуктовую команду не лучшая идея. Создание команды DS as a Service позволит команде набраться экспертизы, понять, где есть потребность в их работе, а главное, выработать общие правила и общую DS-культуру. В этом случае команда DS as a Service является переходной к появлению кросс-функциональных продуктовых команд.
- Работа DS является совсем эпизодической и в продуктовой команде не нужно вовлечение со стороны DS на большое количество времени. В этом случае можно привлекать DS в продуктовые команды на то время, пока разрабатывается ML-сервис. При этом DS точно также посещает мероприятия команды, к которой он присоединяется. Это позволяет другим членам команды понимать, чем он занимается и оперативно помогать для того, чтобы быстрее получить бизнес-результат.

## Преимущества кросс-функциональной продуктовой команды

Итак, мы считаем, что кросс-функциональная команда стабильного состава, сформированная вокруг одного или нескольких ML-продуктов – эффективный способ увеличить эффективность:

- Команда отвечает за бизнес-результат
- Члены команды мотивированы и вовлечены
- Команды несут ответственность за ML-продукты на протяжении всего их жизненного цикла
- Командное взаимодействие обеспечивает лучший T2V (time to value) за счет плотного взаимодействия в течении проекта

## Лидерские роли в команде

- Product Manager/Product Owner. Несет ответственность (accountable) за бизнес-результат перед заинтересованными лицами.
- Process Lead. Отвечает за эффективную работу команды. Фасilitирует (проводит) встречи, решает проблемы с зависимостями, является коучем и ментором для других членов команды.

## Выбор подхода

### Какую проблему решаем

Вы сформировали команду проекта. Она состоит из двух DS и одного BE. Как будет работать команда? Вы поговорили с другими руководителями в компании. Большинство используют Scrum. Вы изучили вопрос и Scrum выглядит простым и понятным.

Вот свойства фреймворка Scrum:

- Работа разбивается на короткие отрезки времени, которые называются спринтами. Типичная длина спрингта — две недели
- Для каждого спрингта определяется цель спрингта. Спрингт считается успешным, если команда добилась цели спрингта.
- Бэклог спрингта состоит из пользовательских историй (user stories). Каждая пользовательская история имеет ценность для пользователя
- Пользовательские истории начинаются в спрингте и доделываются внутри спрингта до конца. Если точнее — они должны соответствовать критериям готовности (Definition of Done). Это означает, что пользовательская история разработана, протестирована, баги найдены, исправлены и закрыты, продукт задеплоен на stage или prod окружение.

Что дает команде такая работа?

- Цель спрингта обеспечивает фокус на результате. Грубо говоря, есть чем похвастаться на Sprint Review (встрече с заинтересованными лицами), которая проходит в конце каждого спрингта.
- Короткий Time to Market. От момента старта разработки до поставки проходит один спрингт, в большинстве команд это 2 недели
- Это очень упрощает планирование. Если мы на самом деле фокусируемся на доделывании до конца наших пользовательских историй, то мы не тянем в следующий спрингт доделки и баги с прошлых спрингтов. Тогда все прозрачно и понятно!

Выглядит разумно. Вы объясняете команде правила и запускаете свой первый спрингт.

Через несколько спрингтов вы замечаете несколько проблем:

- Никакой разумной цели спрингта, как правило, поставить не получается. Даже если вдруг мы формулируем цель, достичь ее к концу спрингта практически невозможно.
- В конце спрингта есть куча недоделанной работы, которая в конце просто переносится на следующий спрингт
- Внутри спрингта из-за discovery-характера DS-проектов может случиться нечто, что полностью уничтожает смысл доделывать спрингт до конца.

Scrum работает совсем не так, как в других софтверных командах.

### В чем причина таких проблем?

У всех этих проблем есть одна корневая причина. Для большинства DS-проектов Жизненный Цикл гипотезы намного превышает две недели. По моим ощущениям, реализация

гипотезы для большинства DS проектов от момента начала до состояния «готово» занимает 3-8 недель.

В итоге пропадает сам смысл спринта, как периода времени, в течении которого мы должны доделать ценную для пользователя работу от начала до конца.

На картинке слева в течении спринта команда успевает сделать несколько ценных для бизнеса результатов. На картинке справа создание ценного результата занимает больше времени.

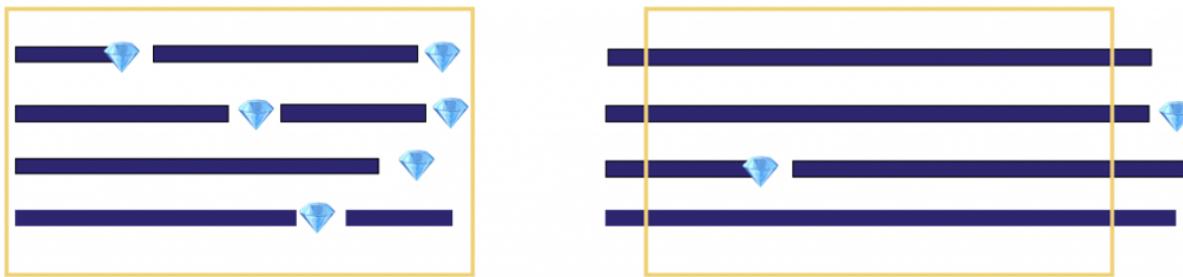


Рис. 24: Scrum vs Kanban

Почему же многие DS-команды выбирают Scrum? Мне кажется, тут играет роль некий психологический фактор. Планирование спринта для таких команд выглядит так: каждый DS набирает себе работы на две недели. Это дает ощущение контроля за происходящим и приятное удовлетворение тем, что все заняты и никто не бездельничает.

Как вариант, можно удлинить спринт до двух месяцев или просто относится к концу спринта как к дате встречи по планированию и не требовать завершения ценной работы. Оба этих подхода снижают прозрачность статуса работы. В течении очень длительного периода времени ничего не меняется: задачи не доделываются, нет возможности получить обратную связь.

Альтернативным способом ведения работ является использование метода Kanban.

## О канбане

В рамках этой короткой книги мы не будем детально рассматривать канбан-метод. Он предназначен для управления всеми типами профессиональных сервисов. Мы рассмотрим его исключительно в применении к командам, занимающимися DS-проектами и продуктами.

В канбан включены три повестки (то есть призыва к действию):

- Устойчивость. Часто в работе мы видим проблему и отправляем все силы на ее решение. Потом появляется новая проблема и мы опять бросаемся на ее решение. Канбан предлагает так отрегулировать работу, чтобы она стала равномерной и устойчивой.
- Ориентация на оказание услуг. В центре внимания — фокус на максимальной удовлетворенности потребителей результатом работы
- Живучесть. Какие бы изменения не ожидали нас, мы должны быть способны быстро перестроить работу

## Практики канбан

Канбан состоит из 6 практик.

- Визуализация
- Ограничение незавершенной работы
- Управление потоком
- Применение явных политик
- Применение обратной связи
- Совместное улучшение, экспериментальное развитие

Далее мы посмотрим, как их можно применить к созданию ML-продуктов.

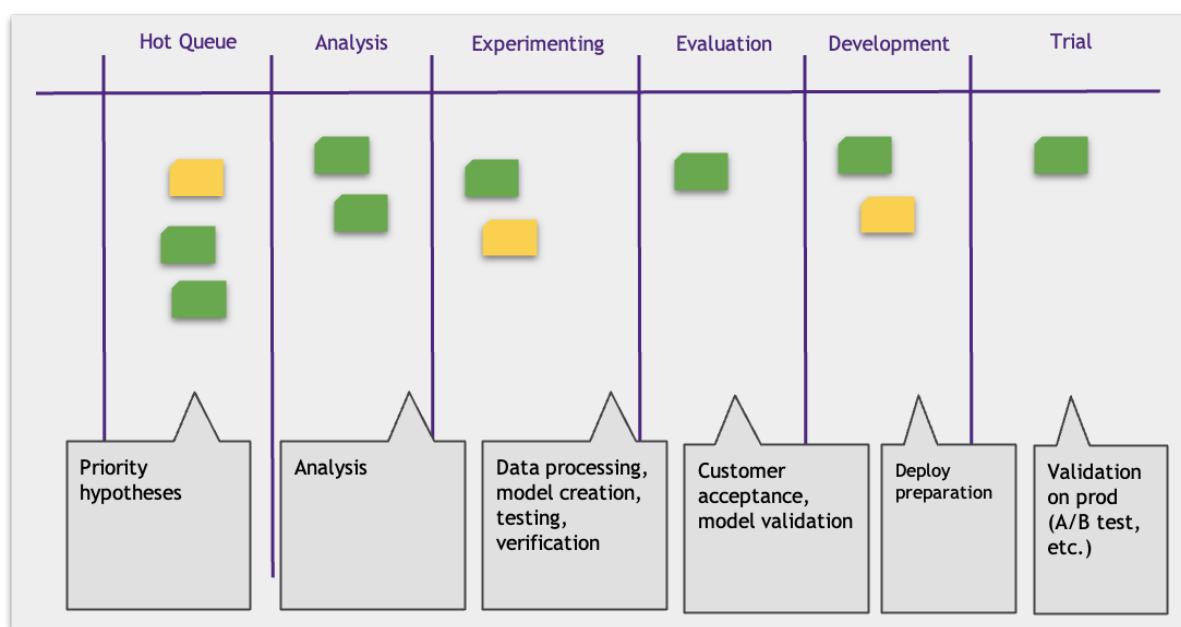
## Визуализация работ

### Колонки канбан-доски

Как можно повысить прозрачность работ? Конечно, каждая гипотеза может быть либо сделанной, либо находится в работе. Многие гипотезы выполняются долго и наличие некоторой промежуточной информации о ее статусе повысило бы прозрачность.

Канбан предлагает визуализировать промежуточные этапы жизни гипотезы на канбан-доске при помощи отдельных колонок.

Каждая работа превращается в тикет (ticket) и помещается на доску. Тикеты двигаются по доске слева направо, перемещаясь из колонки в колонку по мере изменения статуса. Проваленные гипотезы убираются с доски.



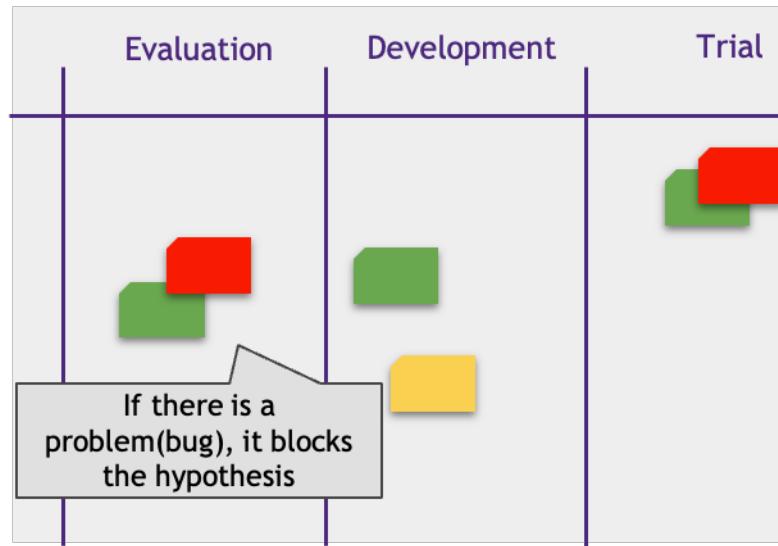
**Рис. 25:** Пример канбан-доски команды

Доска вашей команды может выглядеть по-другому. Например, вы можете захотеть разбить колонку Experimenting на Data Preparation (подготовка данных) и Modelling (создание модели).

### Блокеры

Иногда член команды не может продолжить работу над гипотезой из-за проблемы. Например, мы ждем выборку от клиента. Почему бы не переключиться пока на другую гипотезу?

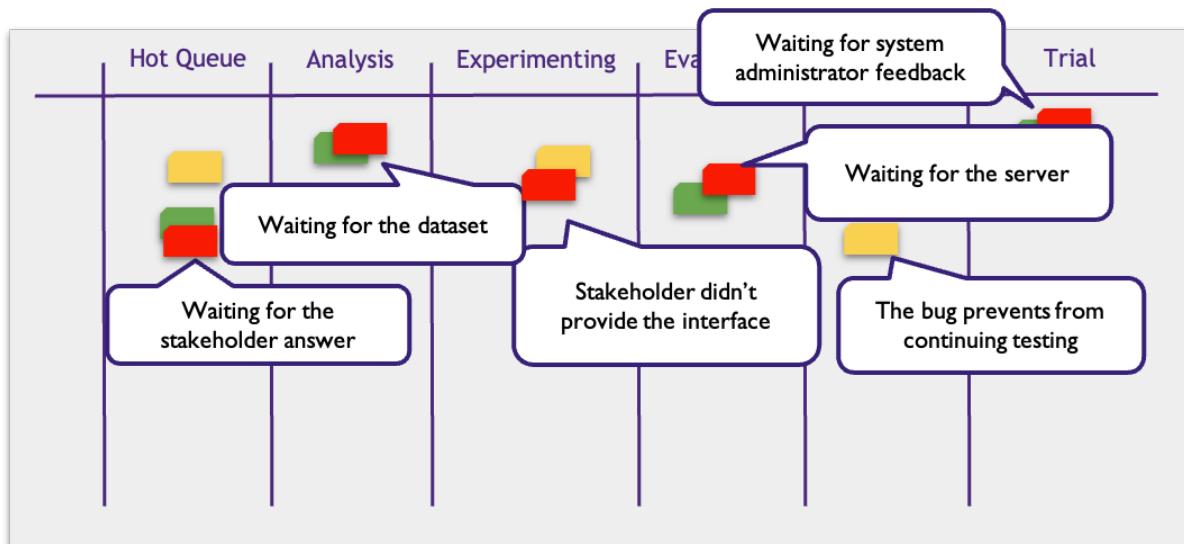
Таких проблем может быть множество. Все они замедляют работу над гипотезами и ухудшают Time2Value.



**Рис. 26:** Блокеры на канбан-доске

Давайте визуализируем блокирующие проблемы. К тикету гипотезы подклеим стикер с описанием блокирующей проблемы. Тогда один взгляд на доску скажет нам, сколько и каких гипотез заблокировано.

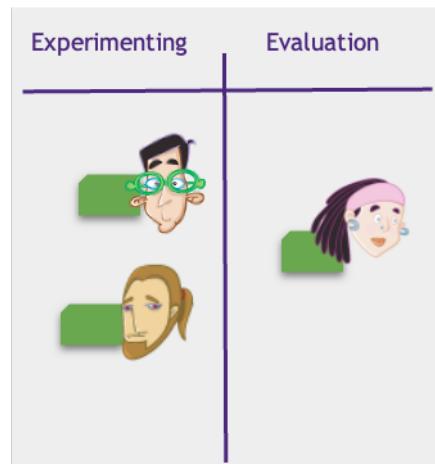
На блокере можно указать предполагаемые сроки решения и ответственного члена команды.



**Рис. 27:** Примеры блокеров

### Визуализация ответственности

Эдик сегодня занимается созданием модели. Для того, чтобы это визуализировать, он прикрепляет свой аватар на соответствующий тикет. В качестве аватара на живой доске мы часто используем фотографию. В электронных досках это делается по-разному, в зависимости от инструмента, например можно указать ответственного (responsible). Как только Эдик закончил свою часть работы, он убирает свой аватар с тикета.

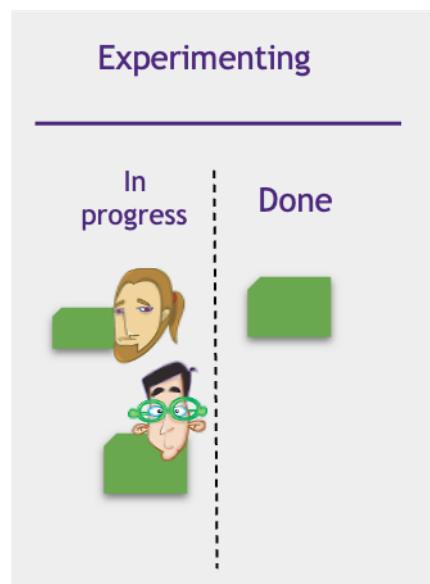


**Рис. 28:** Аватары

- Одной гипотезой может заниматься сразу несколько человек и тогда на ней указывается сразу несколько аватаров.
- Гипотеза может перейти к другому человеку и тогда аватар меняется на новый.
- Если гипотезой никто не занимается, очевидно, на ней нет аватаров.
- По аватару мы видим, кто отвечает за задачу и ожидаем от этого человека отчет по задаче на стендах.

## Вытягивание

Бекэнд разработчица Беки занимается оценкой и автоматизацией моделей, которые строят сапиентисты Эдик и Тим. Процесс передачи работ можно визуализировать следующим образом. Колонка разделяется на две части: In progress и Done. Пока идет работа над гипотезами, их тикеты находятся в подколонке In Progress. Как только работа закончена - они перемещаются в подколонку Done. Это сигнализирует Беки, что она может ими заняться.



**Рис. 29:** Вытягивание работ

## Тикет

В большинстве команд используются электронные доски управления задачами: Jira, Kanban, Trello, Redmine и т.д.

По такой доске двигаются электронные тикеты. Вот составляющие элементы этого тикета:

- Название. Короткое и запоминающееся название гипотезы
- Описание. Сформулированное в виде гипотезы описание
- Member. Ответственный за тикет член команды.
- Чек-лист задач
- Комментарии

Чек-лист задач — список работ, которые надо сделать по тикету. Не обязательно планировать все работы по тикету вплоть до его валидации. Это вполне может быть набор самых ближайших и понятных задач. Относитесь к этому списку как TO DO листу, а не коммитменту. В любой момент можно удалить/добавить или изменить элементы этого списка.

Чек-лист удобно обсуждать совместно с другими членами команды. Это повышает прозрачность и вовлеченность участников.

## Пример тикета в Trello

The screenshot shows a Trello ticket card with the following details:

- Title:** Patching pre-processing
- Column:** Analysis
- Participants:** A placeholder for a user profile picture and a '+' button to add more participants.
- Description:** We believe that patching pre-processing (which used by flair now) will increase the Spacy metric (which model learns from scratch now) to the current flair level.
- Metrics:** F1 0.752, Precision 0.882, Recall 0.655
- TODO List:**
  - Status:** checked
  - Name:** Setup pre-processing, which is in Flare now
  - Status:** unchecked
  - Name:** Compute metrics
  - Status:** unchecked
  - Name:** Compare to baseline and create Findings
- Add element:** A button at the bottom of the TODO list.

Рис. 30: Ticket Example

## Типы работ (Work Item Types)

Давайте поговорим о том, какие тикеты ходят по доске

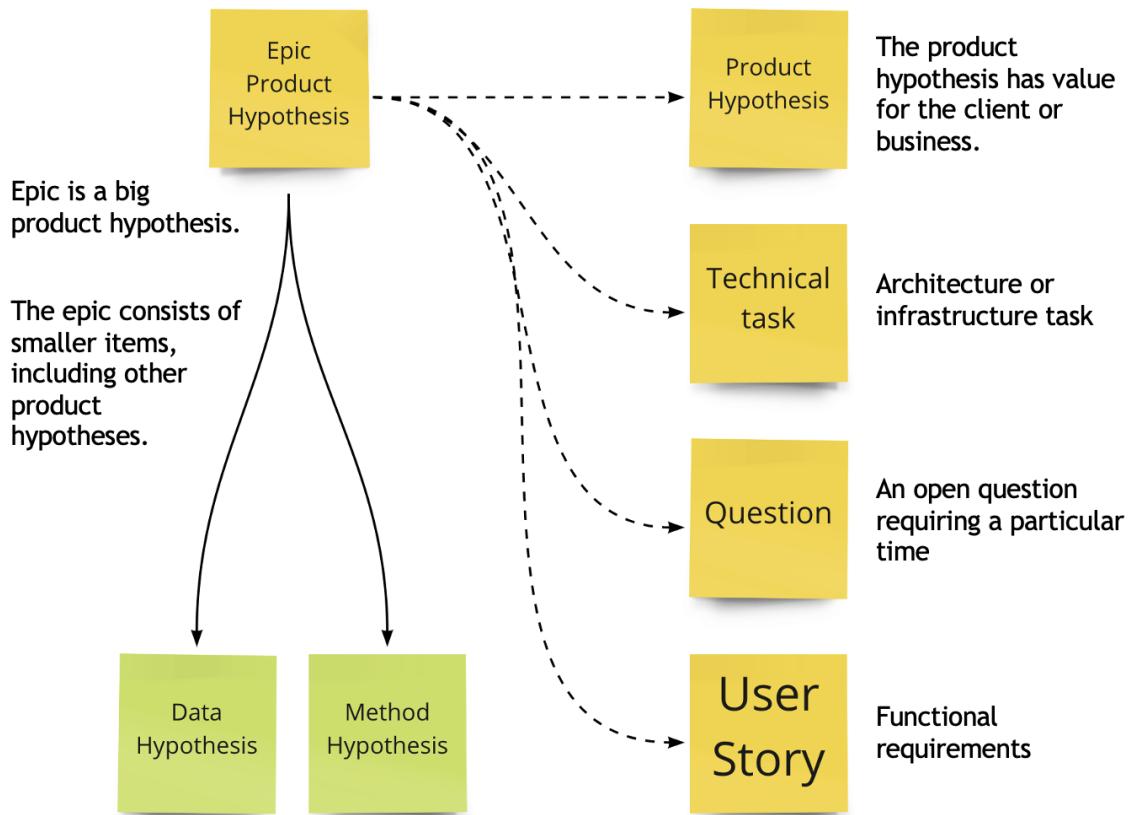


Рис. 31: Work Item Types

Слово «Эпик» означает, что работа большая. Обычно у команды 1-3 эпика в работе и помещать их на доску команды не имеет смысла. Они там просто не будут двигаться.

## Ограничение количества одновременно выполняющейся работы

### Закон Литтла

Представьте себе, что вы оказались в очереди в Starbucks за стаканчиком любимого напитка (я лично предпочитаю капучино на кокосовом молоке). Перед вами в очереди 10 человек. Кафе покидает примерно 1 человек в минуту. Через какое время вы получите свой напиток? Очевидно, вам придется подождать примерно 10 минут = 10 человек / 1 человек в минуту.



Рис. 32: Starbucks Queue

В теории массового обслуживания существует закон Литтла (Little's Law), который гласит, что среднее время цикла (времени обслуживания) элемента равняется среднему количеству элементов в системе поделенное на среднюю скорость обслуживания.

$$\text{Lead Time} = \frac{\text{Work In Progress}}{\text{Throughput}}$$

Рис. 33: Закон Литтла

В моем примере вы сами являетесь элементом системы. Тогда Lead Time – среднее время ожидания в очереди, Work In Progress – среднее количество людей в очереди, а Throughput – средняя скорость обслуживания, то есть среднее количество покупателей, выходящих с напитком из магазина за минуту.

Закон Литтла работает для любой системы массового обслуживания. Вы можете оказаться в большом торговом центре, где есть кофе-пойнты, магазины и рестораны. И даже в этом случае среднее время присутствия человека в торговом центре равно среднему количеству людей внутри торгового центра, поделенному на среднюю скорость выхода людей из него.

Давайте применим закон Литтла к тикетам на канбан-доске. Получается, что для снижения среднего времени прохождения тикета нам надо либо снизить количество тикетов на

доске либо увеличить пропускную способность (скорость) нашей системы.

Теперь понятно почему важно ограничить максимальное количество тикетов на канбан-доске. Такое ограничение в канбане называется WIP-limit (Work in Progress limit).

## Как работает ограничение WIP в Канбан

Над каждой колонкой мы пишем максимальное количество тикетов (WIP), которые могут быть в этой колонке.

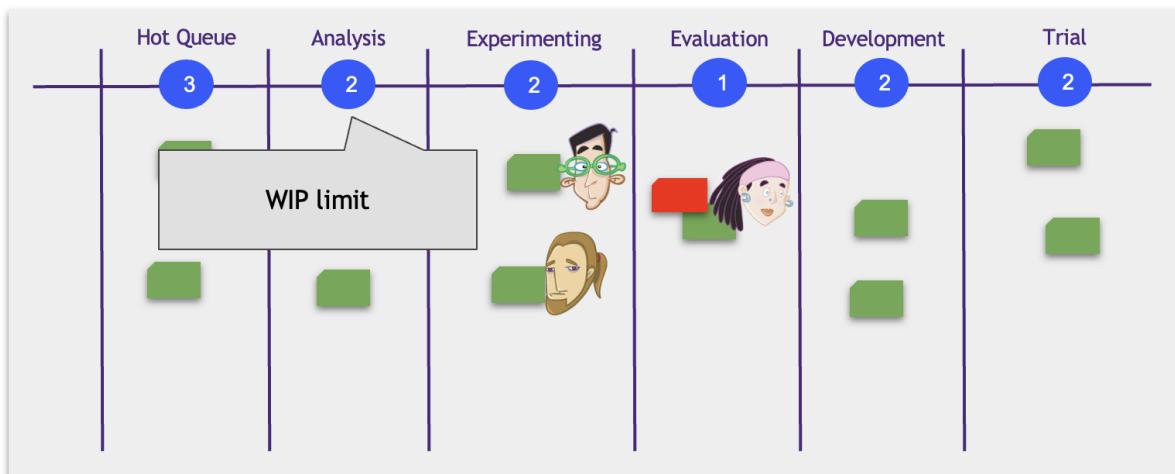


Рис. 34: WIP for Kanban Board

Эдик закончил работу, модель готова и перед отправкой на автоматизацию ее нужно проверить. Она попадает на этап Evaluation к Беки. Она находит проблему.

Эдик освободился и по идеи может взять в колонку Experimenting новую интересную работу. Но ему не дает ограничение  $WIP=2$ . Ему придется решить проблему с багом и лишь потом он сможет переключиться на новую гипотезу.

Ограничение WIP заставляет команду работать совместно и фокусироваться на доделывании работ.

- Теперь мы не можем игнорировать блокеры. Если их станет слишком много, нам придется ими заняться.
- Если кто-то из членов команды перегружен работой, то на доске появятся очереди и опять таки, ограничение не даст нам их игнорировать.
- Освободившийся человек займется наиболее приоритетной работой



**Рис. 35:** Иллюстрация с обложки книги Дэвида Андерсона “Kanban: Successful Evolutionary Change for Your Technology Business”

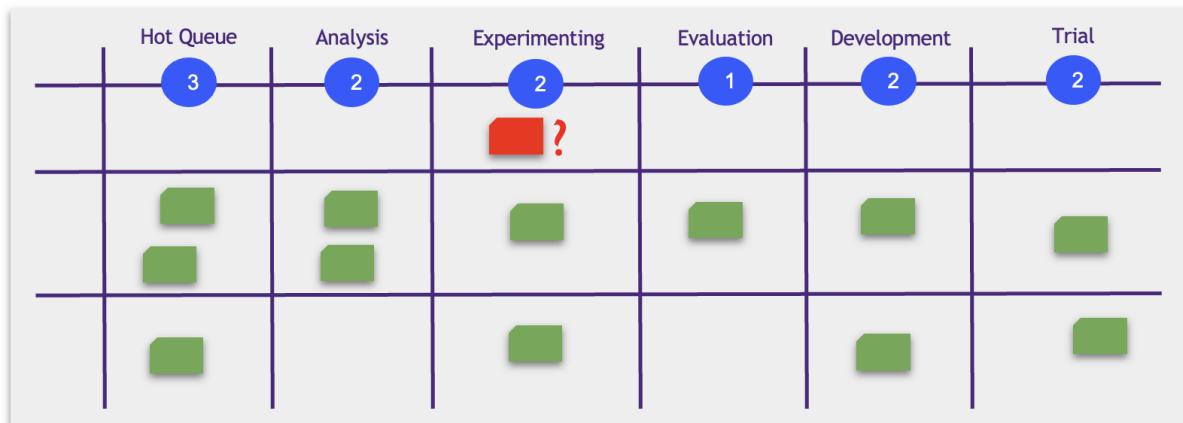
Иллюстрация с обложки книги Дэвида Андерсона “Kanban: Successful Evolutionary Change for Your Technology Business”

## Срочные задачи (Expedite) и Swimlanes

У одного из клиентов обнаруживается проблема: по какой-то причине модель в проде перестала показывать хорошие результаты. Вы завели новый баг и он сразу попадает на этап Experimenting.

Это срочный баг, мы теряем деньги из-за того, что модель не работает. Но как быть с ограничением WIP? В колонке Experimenting не может быть больше двух тикетов!

Действительно, мы не хотим нарушать правила. Откатывать назад множество тикетов, чтобы искусственно соблюсти правило тоже не разумно.



**Рис. 36:** Ускоренный swimlane

Вместо этого мы выделяем полосу движения (swimline) для срочных тикетов и договариваемся о новом правиле: в этой полосе тикет может нарушать ограничения WIP.

Тикеты такого типа называются Expedite (ускоренными).

Таких дорожек (swimlines) можно сделать несколько и по каждому из них пустить отдельные категории работ:

- Отдельные ML-продукты или продуктовые гипотезы
- Типы работ
- Различные заказчики
- ...

## Явные политики (Explicit Policies)

Вы находите проблему в процессе и придумываете правило, которое позволяет эту проблему решить или хотя бы минимизировать.

Например, вы решаете указывать зависимости от внешних команд на этапе анализа гипотезы. Это позволит видеть риски получить блокер заранее и решить, что с ним делать.

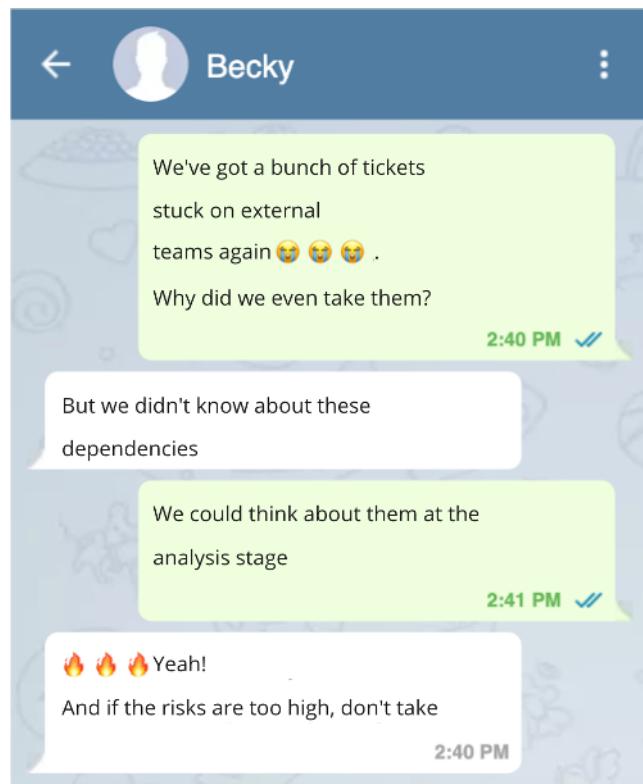


Рис. 37: Chat

Такие правила удобнее всего записывать в виде Explicit Policy (Явных политик).

Explicit Policy — чек-лист со списком правил, отвечающих на вопрос «можно ли перенести тикет в следующую колонку».

## Пример Explicit Policies

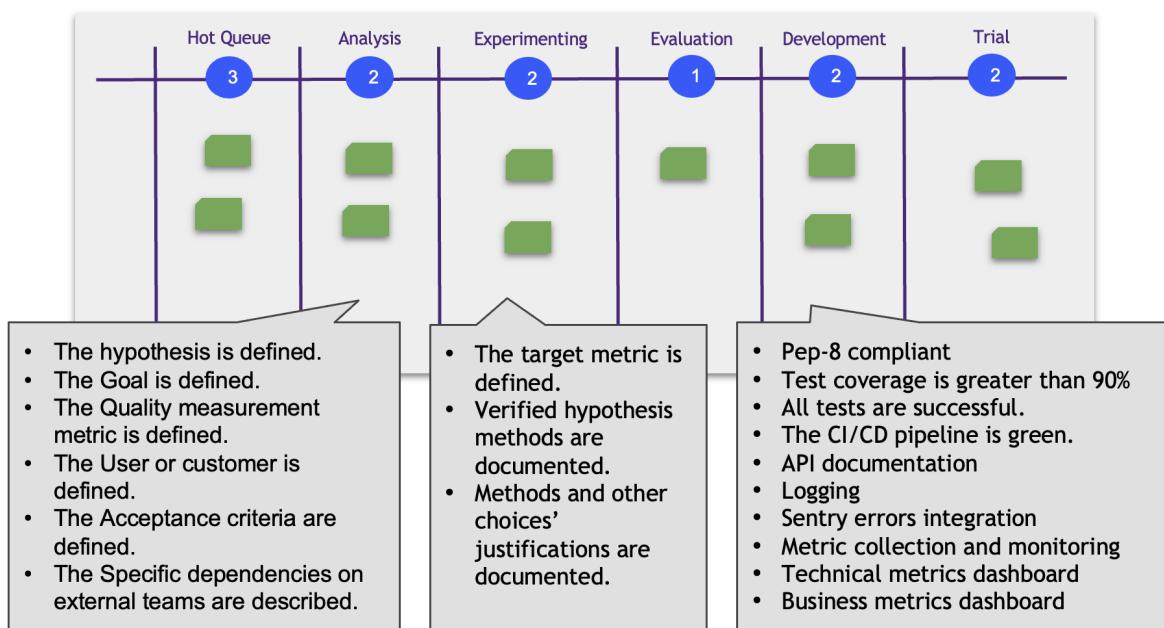


Рис. 38: Пример Explicit Policies

## Встречи в канбан

### Stand-up meeting aka Kanban-meeting

Стэндап – ежедневная встреча команды около доски команды. Вот правила, по которым проводится встреча:

- Участники – команда + внешние эксперты при необходимости
- Стендап проходит ежедневно в одно и то же время
- Команда проходит по тикетам доски справа налево
- Ведущий встречи по каждому тикету задает вопрос «Что мешает продвинуть эту работу дальше?»
- Все вскрытые блокеры получают ответственных и даты решения
- Команда фокусируется на ускорении продвижения задач по доске
- Команда протягивает в первую очередь самые старые задачи

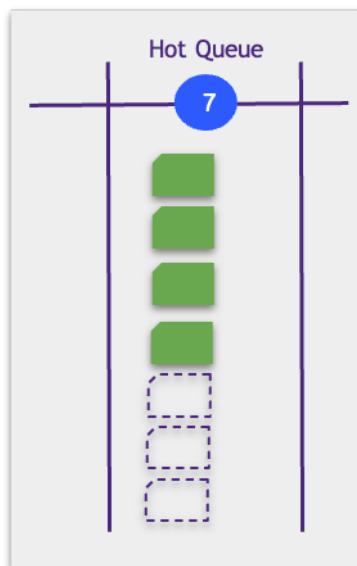
Мы стараемся уложить длительность стендапа в 15 минут. Это может показаться сложной задачей, но со временем вы научитесь укладываться даже в более короткий срок:

- На старте запуска стендапа мы обсуждаем каждый тикет, по мере роста опыта команды обсуждаем только блокеры Команда приучается работать с доской вне стендапа, обозначая блокеры, помечая в комментариях статусы, разбивая работы на задачи и т.д.
- Команда взаимодействует друг с другом ad-hoc, постоянно и непрерывно, используя стендап просто как дополнительную точку синхронизации
- На самом стендапе команда учится говорить только важную для других информацию
- Все необходимые совместные обсуждения переносятся за пределы стендапа

### Планирование aka replenishment meeting (встреча по пополнению)

Вот правила, по которым проходит планирование:

- Участники – команда + внешние эксперты + заинтересованные лица
- Участники выбирают наиболее важные работы из бэклога и помещают их в Hot Queue
- Мы можем поместить в Hot Queue столько работ, сколько у нас есть свободных слотов. На картинке у нас WIP = 7, в Hot Queue 4 гипотезы, значит всего есть три свободных слота для новых работ



**Рис. 39:** Replenishment

### Ретроспектива aka Ревью сервиса поставки

Регулярно (обычно раз в две недели либо раз в месяц) команда собирается для анализа эффективности собственной работы. Типичная длительность встречи – от 1 часа до 1.5 часов.

У ретроспектины есть главное правило (Prime Directive), которое ввел Норман Керт:

Вне зависимости от того, что мы обнаружим, мы понимаем и твердо верим, что все сделали максимум возможного в существовавших на тот момент ограничениях времени, навыков, возможностей и ресурсов.

дуктивность встречи будет нулевой, если она превратится в поток взаимных обвинений. Мы обсуждаем, как изменить систему, чтобы исправить найденные проблемы.

Вот несколько правил проведения ретроспектины:

- Ретроспектива – обсуждение эффективности процесса, а не итогового продукта
- Результаты ретроспективы – изменения подходов к работе. Они могут найти отражение в появлении или удалении колонок на доске, ограничений WIP, правил Explicit Policies, правил использования доски
- Решения ретроспективы - это эксперименты. Мы договариваемся попробовать запустить предложенные там изменения и вернуться к обсуждению их эффективности на последующих ретроспективах

### Формат проведения ретроспектины

Форматов проведения ретроспектины существует огромное количество. Один из самых простых и удобных выглядит следующим образом:

Вы разбиваете доску на 4 части: \* Плюсы. Что было позитивным с момента проведения прошлой ретроспектины, что мы хотим сохранить в процессе. \* Проблемы. Что замедляло, понижало эффективность работы, привносило риски \* Идеи. Что можно улучшить для решения проблем \* План. Что мы попробуем изменить в процессе

Ведущий зачитывает результаты прошлой ретроспектины, затем последовательно проводит участников через все части доски. Команда накидывает стикеры с идеями. Конечный итог ретроспектины — план с набором изменений процесса.

## Управление потоком работ

Одна из ключевых идей канбан – управление работами, а не людьми. Мы выстраиваем поток работ с помощью канбан-доски, помогая участникам процесса самоорганизоваться вокруг такого потока. Мы добиваемся устойчивости такого потока: настраиваем колонки, ограничения WIP и правила Explicit Policies, чтобы задачи текли быстро (с низким Lead Time) и с хорошей пропускной способностью (Throughput). Изменения неизбежны, и мы постоянно подкручиваем кранники и болтики нашей системы. Сама команда следит за процессом и предлагает улучшения – ведь весь процесс прозрачен и визуален.

Эффективность предлагаемых улучшений можно отследить при помощи нескольких метрик и диаграмм. Численное управление потоком работ – отдельная интересная и очень большая тема. Мы рассмотрим лишь два примера таких метрик.

### Lead Time Distribution

Эта диаграмма позволяет оценить Lead Time – за какое время мы делаем работы. По оси X мы откладываем время, за которое была выполнена работа, по оси Y – количество соответствующих работ. Обычно считают медианное время выполнения работ и 85% персентиль для каждого типа работ (Work Item Type). Цель команды – снизить Lead Time до нужного уровня.

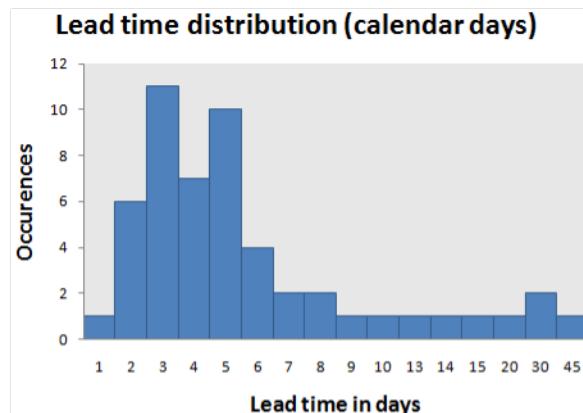
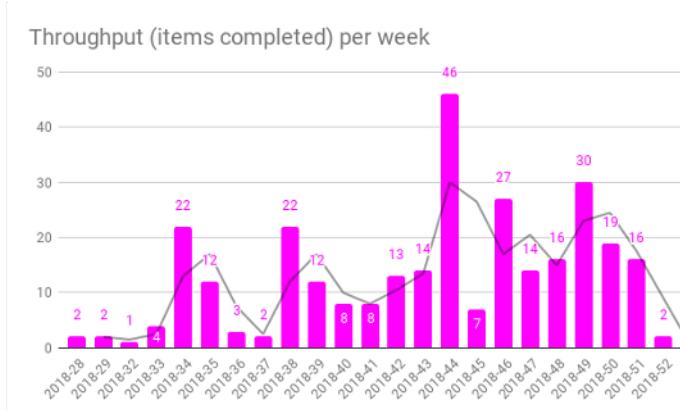


Рис. 40: Lead Time Distribution Diagram

### Throughput Diagram

Вторая важнейшая метрика для команды – пропускная способность, то есть количество работ, которые мы даем в единицу времени (например, в месяц). Эту метрику интересно сравнивать с потребностью (Demand) – количество новых работ в бэклоге команды. Если Throughput заметно меньше Demand – пора думать о расширении команды.



Канбан-метод был предложен David Anderson и впервые опубликован в книге *Kanban. Successful Evolutionary Change for Your Technology Business*

Канбан-метод был предложен David Anderson и впервые опубликован в книге *Kanban. Successful Evolutionary Change for Your Technology Business<sup>a</sup>*.

<sup>a</sup><https://www.amazon.com/Kanban-David-J-Anderson-ebook/dp/B0057H2M70/>

## Практическое использование LeanDS

Отлично! Вы практически дочитали эту книгу! Мы с вами рассмотрели ключевые практики управления проектами и продуктами в Data Science.

Я надеюсь, вы зажглись попробовать LeanDS на практике!

В заключение хочу сделать несколько замечаний по поводу их практического использования и внедрения.

### Нужно ли использовать все практики LeanDS?

Конечно нет!

Некоторые практики вполне могут оказаться нерелевантными к вашему контексту. Не ко всем практикам вы можете оказаться готовыми на данном этапе. Например, бизнес может отказаться участвовать в приоритизации. Это тоже нормально.

### Как проводить внедрение?

- Запускать сразу все практики в работу довольно опасно.
- С какой практики начать? К сожалению, общего ответа на эту тему не существует. Начните с самых простых и полезных по вашему мнению.
- Большая часть практик намеренно описана в некотором начальном и простом варианте. Вы обязательно будете их дорабатывать под себя и развивать дальше.
- Страйтесь принимать решения по изменению процесса открыто, совместно обсуждая их в команде
- Присоединяйтесь к сообществу LeanDS и задавайте вопросы в нашем чате, мы обязательно поможем!

## Об авторе



**Рис. 41:** Асхат Уразбаев

Привет!

Спасибо, что читаете эту книгу!

Меня зовут Асхат Уразбаев. Я - основатель сообщества LeanDS и один из авторов подхода LeanDS.

Я окончил МФТИ в 2000 году и прошел достаточно классическую карьеру от разработчика до руководителя проектов, руководил тестированием. В компании Luxoft я поучаствовал в Agile проектах в роли владельца продукта и скрам-мастера. С 2006 года я стал аджалайл-коучем.

В 2008 году вместе с партнером я основал компанию ScrumTrek и с тех пор тренирую и консультирую компании по гибким подходам. Моя работа – помогать компаниям выстраивать более эффективные подходы к работе.

Так случилось, что несколько лет назад я оказался руководителем проектного офиса в компании, занимающейся консалтингом в Data Science. Оказалось, что «классические» гибкие подходы к таким проектам не работают. Нам пришлось методом проб и ошибок придумывать работающие практики. Они легли в основу этого подхода LeanDS, описанного в этой книге.

<https://www.linkedin.com/in/urazbaev/>