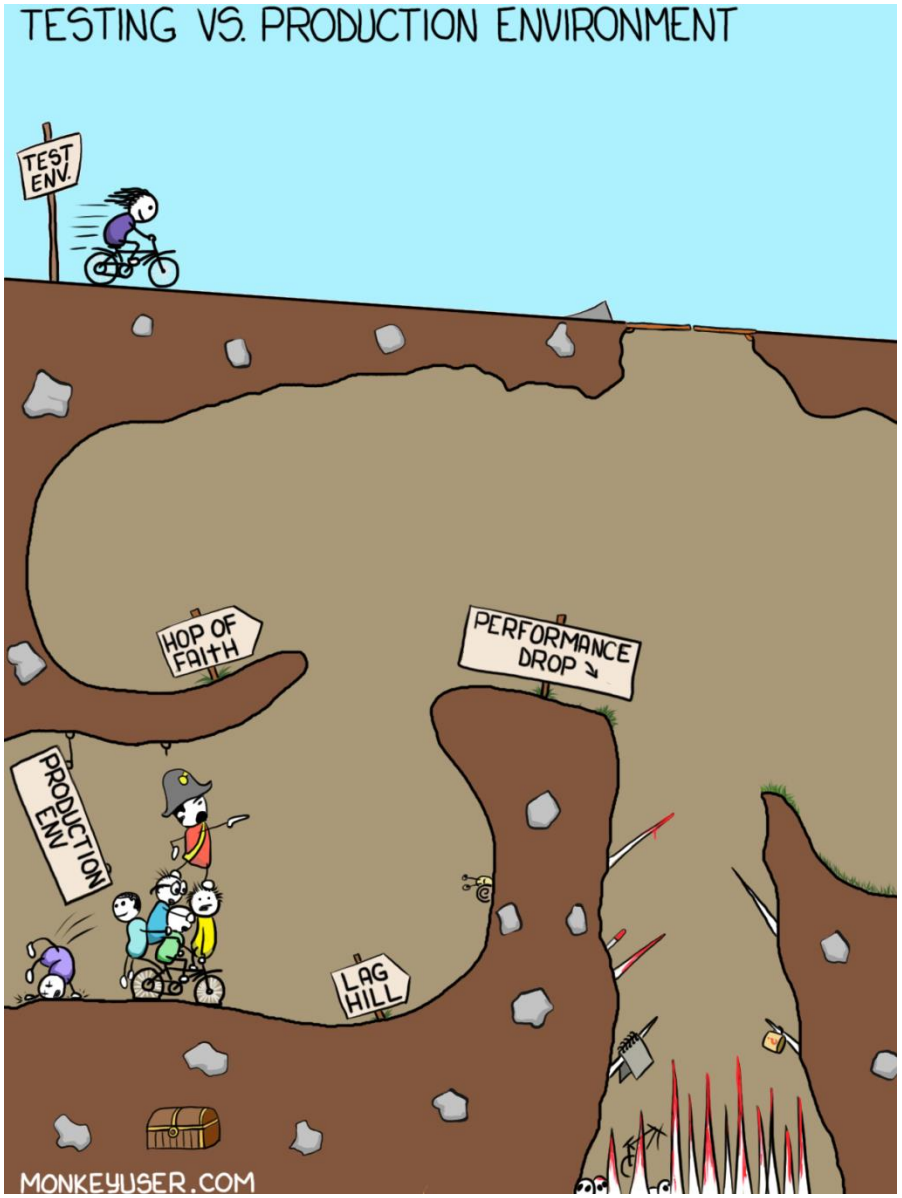
Software engineering

403 lab

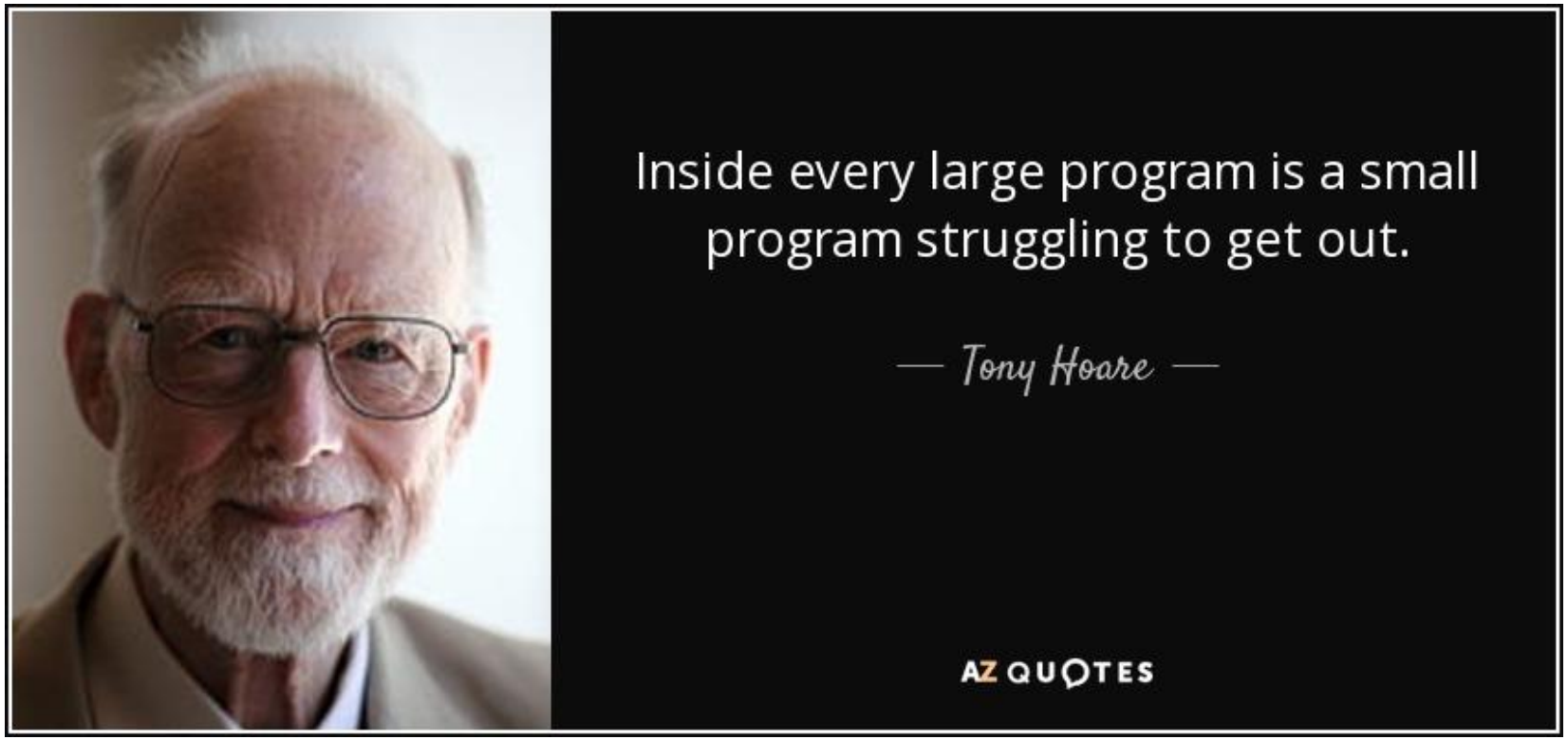
«Скучное» программирование

Влад Чесноков

Запихивание модели на прод – «отдельная тема»



Бизнес-логика – малая часть продукта



403 lab

А каковы другие части продукта?

- сбор данных

А каковы другие части продукта?

- сбор данных
- очистка/парсинг данных

А каковы другие части продукта?

- сбор данных
- очистка/парсинг данных
- валидация/фильтрация данных

А каковы другие части продукта?

- сбор данных
- очистка/парсинг данных
- валидация/фильтрация данных
- обработка ошибок

А каковы другие части продукта?

- сбор данных
- очистка/парсинг данных
- валидация/фильтрация данных
- обработка ошибок
- тесты/QA

А каковы другие части продукта?

- сбор данных
- очистка/парсинг данных
- валидация/фильтрация данных
- обработка ошибок
- тесты/QA
- организация кода (чтобы легче было поддерживать)
- настройка сборки

А каковы другие части продукта?

- сбор данных
- очистка/парсинг данных
- валидация/фильтрация данных
- обработка ошибок
- тесты/QA
- организация кода (чтобы легче было поддерживать)
- настройка сборки
- работа с бд/хранилищем, индексация данных

А каковы другие части продукта?

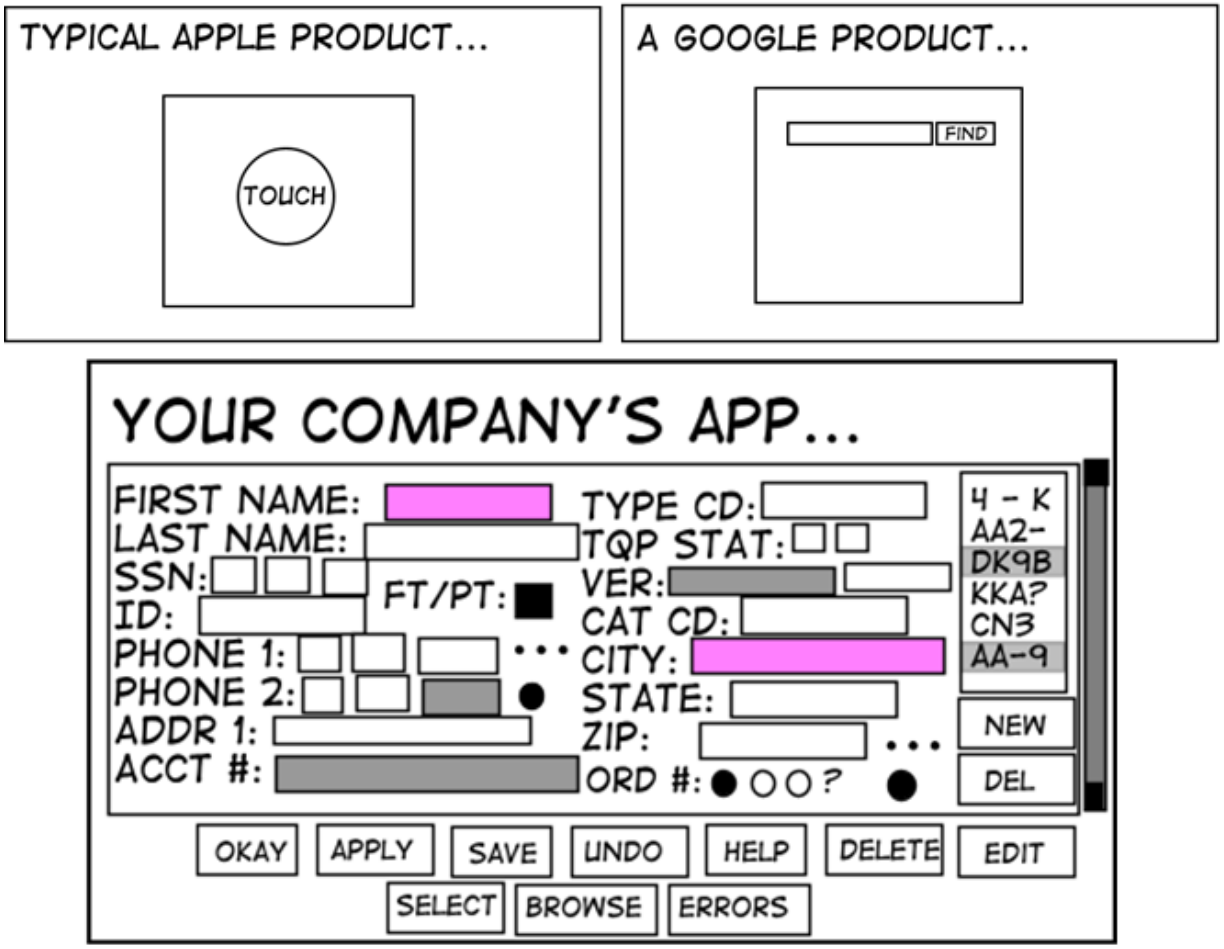
- сбор данных
- очистка/парсинг данных
- валидация/фильтрация данных
- обработка ошибок
- тесты/QA
- организация кода (чтобы легче было поддерживать)
- настройка сборки
- работа с бд/хранилищем, индексация данных
- UI/API

А каковы другие части продукта?

- сбор данных
- очистка/парсинг данных
- валидация/фильтрация данных
- обработка ошибок
- тесты/QA
- организация кода (чтобы легче было поддерживать)
- настройка сборки
- работа с бд/хранилищем, индексация данных
- UI/API

Но это еще не все

- UX



Но это еще не все

- UX
- обогащение данных

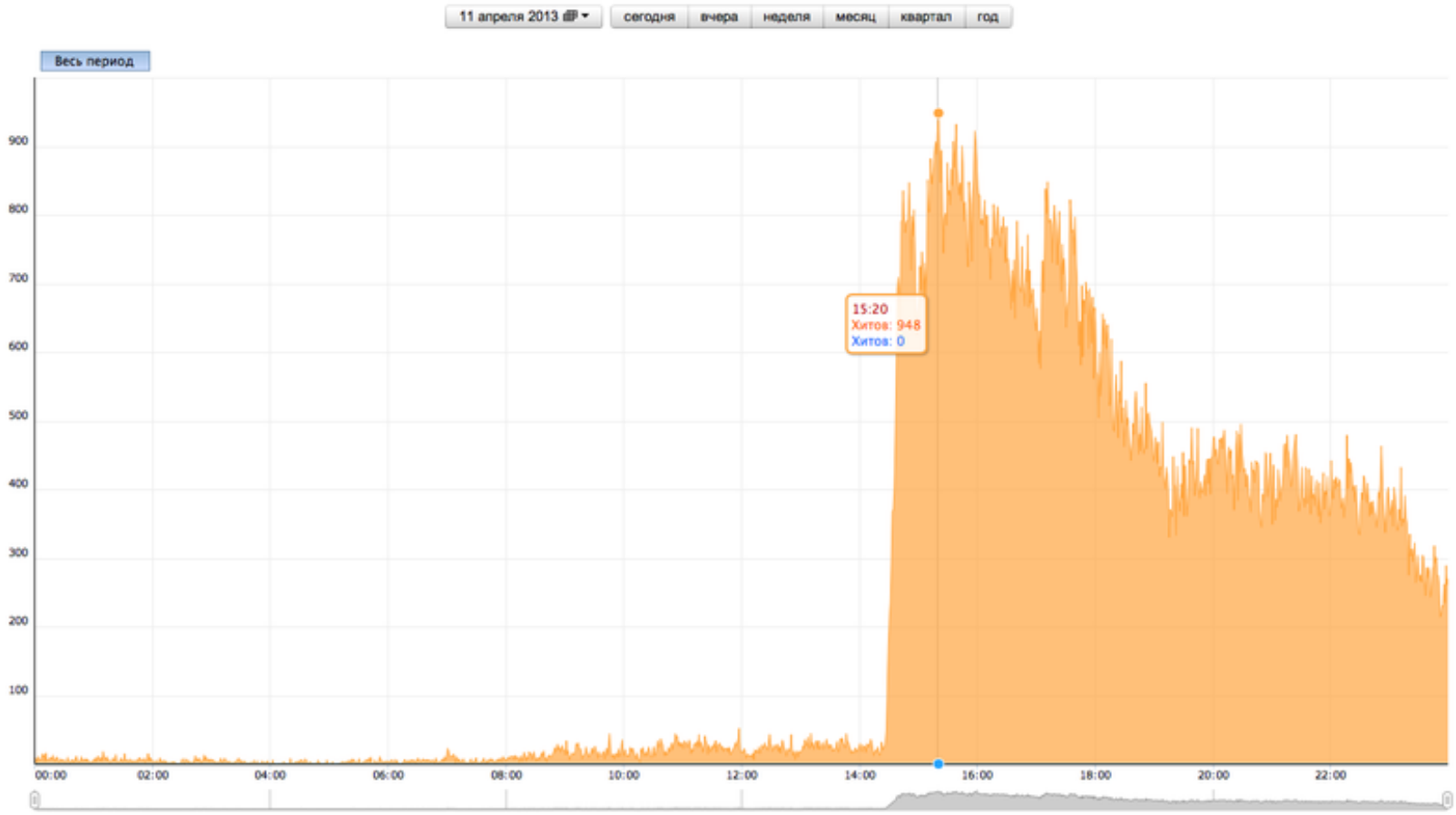
Но это еще не все

- UX
- обогащение данных
- сопряжение с внешними системами
- конфигурация

Но это еще не все

- UX
- обогащение данных
- сопряжение с внешними системами
- конфигурация
- настройка сети
- безопасность

Но это еще не все



Но это еще не все

- UX
- обогащение данных
- сопряжение с внешними системами
- конфигурация
- настройка сети
- безопасность
- масштабирование, высокая нагрузка
- отказоустойчивость

Но это еще не все

- UX
- обогащение данных
- сопряжение с внешними системами
- конфигурация
- настройка сети
- безопасность
- масштабирование, высокая нагрузка
- отказоустойчивость
- развертывание/оркестрация
- деплой CI/CD

Но это еще не все

- UX
- обогащение данных
- сопряжение с внешними системами
- конфигурация
- настройка сети
- безопасность
- масштабирование, высокая нагрузка
- отказоустойчивость
- развертывание/оркестрация
- деплой CI/CD

Но это еще не все

- схема/маппинг/индекс хранилища/БД
- оптимизация хранилищ

Но это еще не все

- схема/маппинг/индекс хранилища/БД
- оптимизация хранилищ
- перекладывание из одного места в другое

Но это еще не все

- схема/маппинг/индекс хранилища/БД
- оптимизация хранилищ
- перекладывание из одного места в другое
- удаление неактуальных данных/дедупликация
- бэкапы

Но это еще не все

- схема/маппинг/индекс хранилища/БД
- оптимизация хранилищ
- перекладывание из одного места в другое
- удаление неактуальных данных/дедупликация
- бэкапы
- логирование и анализ логов
- мониторинг, алертинг

Но это еще не все

- схема/маппинг/индекс хранилища/БД
- оптимизация хранилищ
- перекладывание из одного места в другое
- удаление неактуальных данных/дедупликация
- бэкапы
- логирование и анализ логов
- мониторинг, алертинг
- метрики, дашборды
- бенчмарки
- оптимизация кода

Но это еще не все

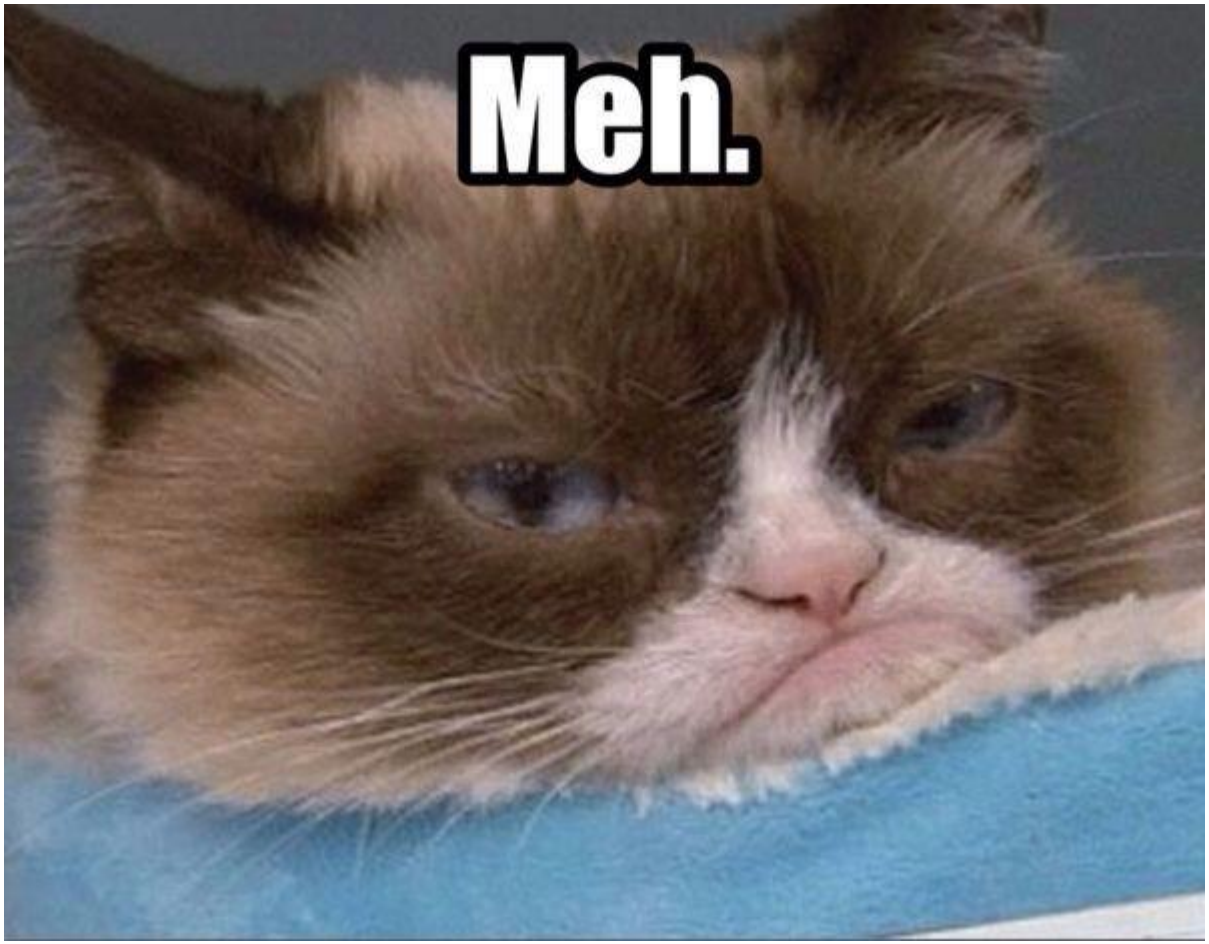
- схема/маппинг/индекс хранилища/БД
- оптимизация хранилищ
- перекладывание из одного места в другое
- удаление неактуальных данных/дедупликация
- бэкапы
- логирование и анализ логов
- мониторинг, алертинг
- метрики, дашборды
- бенчмарки
- оптимизация кода

И даже это не все

- документирование
- администрирование системы/поддержка
- построение процесса разработки
- расчет железа/стоимости облака
- «нетехнические вещи»

* разумеется, все перечисленное на предыдущих слайдах нужно не всегда.

Очень скучно

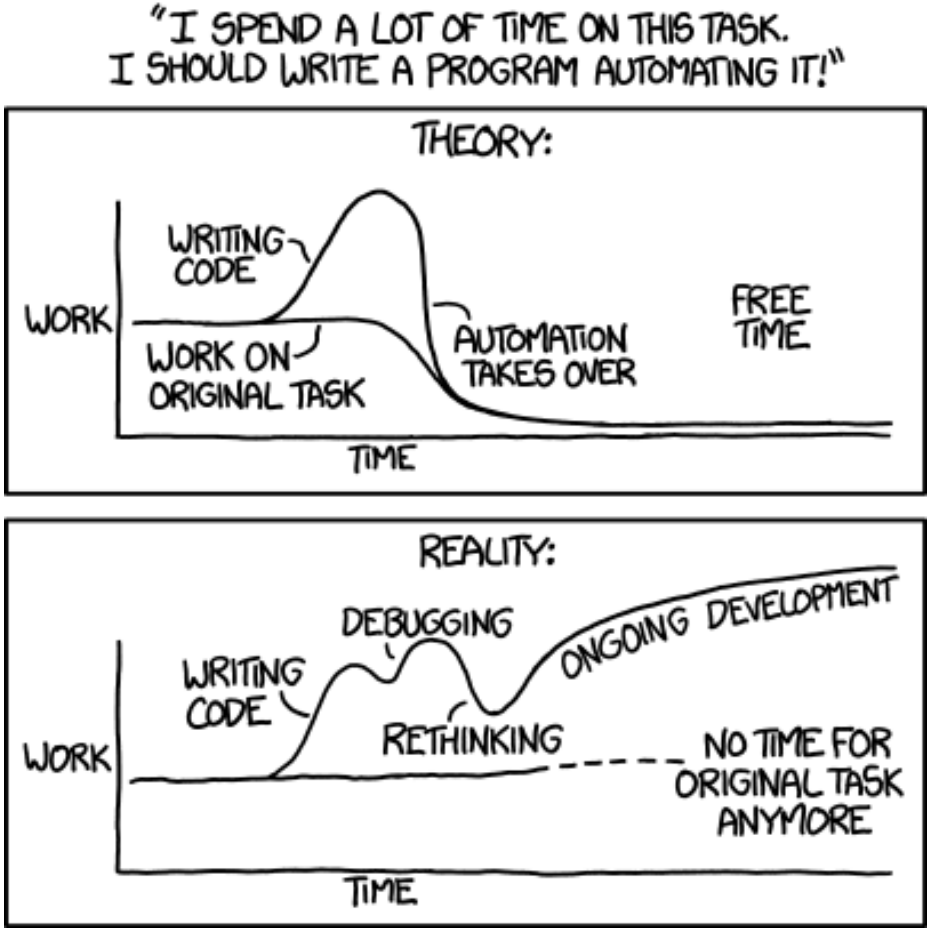


Однако

- каждый проект уникален 
- многие проблемы решены за вас
- надо собирать из того, что есть 
- сжатые сроки!
- долговременная поддержка и развитие
- не только решить проблему, но и предотвратить ее в будущем 
- веселье: изменяющиеся требования

Что делать?

- автоматизировать все, что можно



Что делать?

- автоматизировать все, что можно
- что нельзя – организовать для минимум бойлерплейта и максимум реиспользования
- типовые решения + адаптация + изменение
- база знаний
- постоянная эволюция: технологии, продукт, процессы, люди

Примеры

- кроссвалидация API и клиента
- flaky-тесты
- соответствие маппинга и кода
- сохранение лога ошибки в баг-трекер
- отказоусточивость postgres
- переезд с JS на TS
- data-классы вместо валидации
- публикация только с тестами

Примеры

- кроссвалидация API и клиента
- flaky-тесты
- соответствие маппинга и кода
- сохранение лога ошибки в баг-трекер
- отказоусточивость postgres
- переезд с JS на TS
- data-классы вместо валидации
- публикация только с тестами

Примеры

- кроссвалидация API и клиента
- flaky-тесты
- соответствие маппинга и кода
- сохранение лога ошибки в баг-трекер
- отказоусточивость postgres
- переезд с JS на TS
- data-классы вместо валидации
- публикация только с тестами

Примеры

- кроссвалидация API и клиента
- flaky-тесты
- соответствие маппинга и кода
- сохранение лога ошибки в баг-трекер
- отказоусточивость postgres
- переезд с JS на TS
- data-классы вместо валидации
- публикация только с тестами

Примеры

- кроссвалидация API и клиента
- flaky-тесты
- соответствие маппинга и кода
- сохранение лога ошибки в баг-трекер
- отказоусточивость postgres
- переезд с JS на TS
- data-классы вместо валидации
- публикация только с тестами

Примеры

- кроссвалидация API и клиента
- flaky-тесты
- соответствие маппинга и кода
- сохранение лога ошибки в баг-трекер
- отказоусточивость postgres
- переезд с JS на TS
- data-классы вместо валидации
- публикация только с тестами

Примеры

- кроссвалидация API и клиента
- flaky-тесты
- соответствие маппинга и кода
- сохранение лога ошибки в баг-трекер
- отказоусточивость postgres
- переезд с JS на TS
- data-классы вместо валидации
- публикация только с тестами

Примеры

- кроссвалидация API и клиента
- flaky-тесты
- соответствие маппинга и кода
- сохранение лога ошибки в баг-трекер
- отказоусточивость postgres
- переезд с JS на TS
- data-классы вместо валидации
- публикация только с тестами

Примеры

- кроссвалидация API и клиента
- flaky-тесты
- соответствие маппинга и кода
- сохранение лога ошибки в баг-трекер
- отказоусточивость postgres
- переезд с JS на TS
- data-классы вместо валидации
- публикация только с тестами

Что там у продуктов с ML?

Продукты с ML – не исключение

Те же проблемы + ML-специфичные проблемы

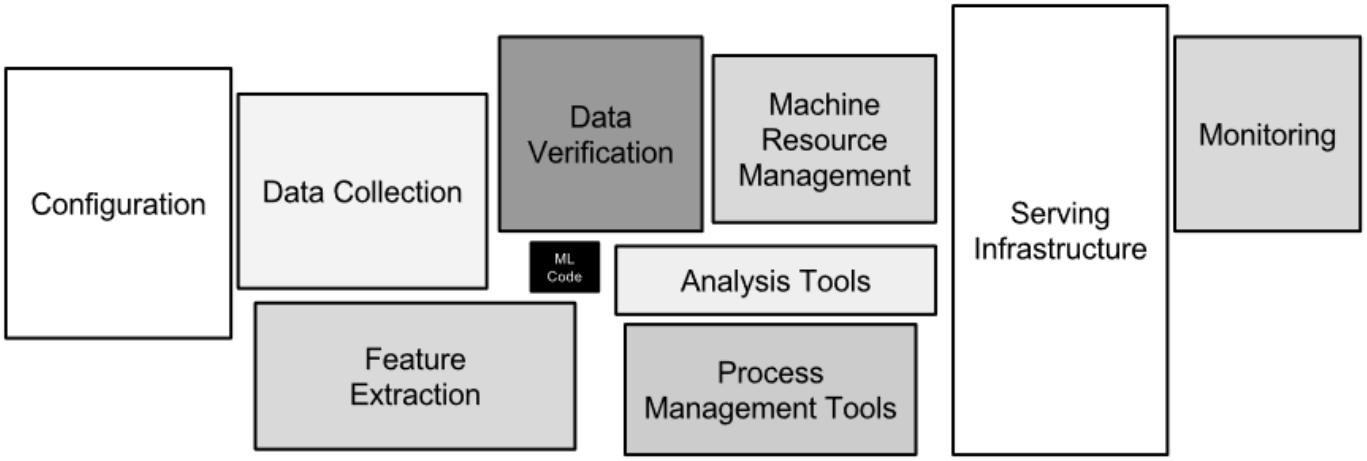


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

https://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf

Данные для ML

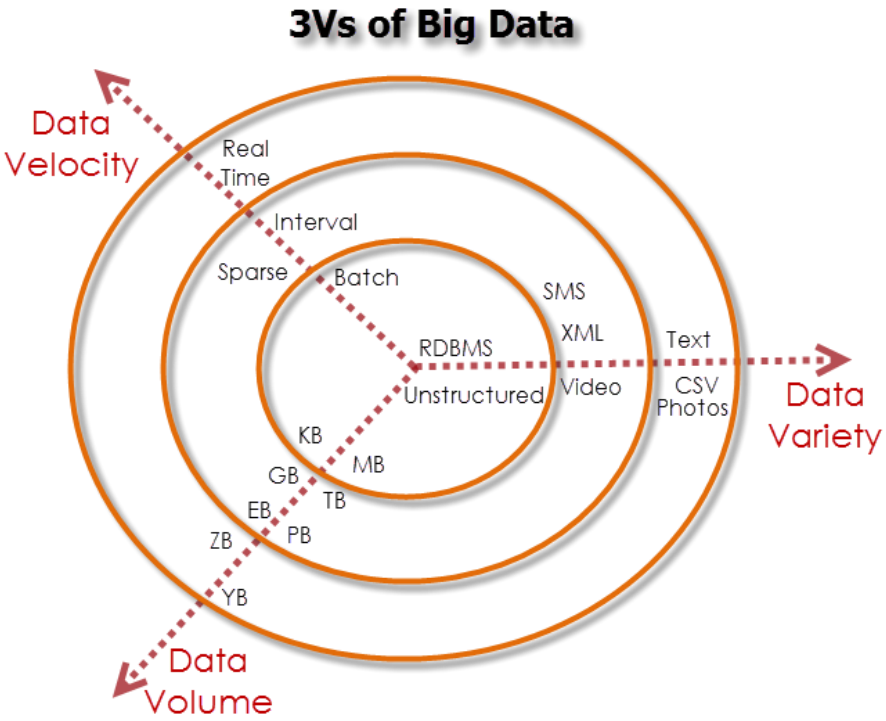
Для обучения модели нужны качественные данные

- для обучения с учителем – много данных
- для обучения без учителя – очень много данных

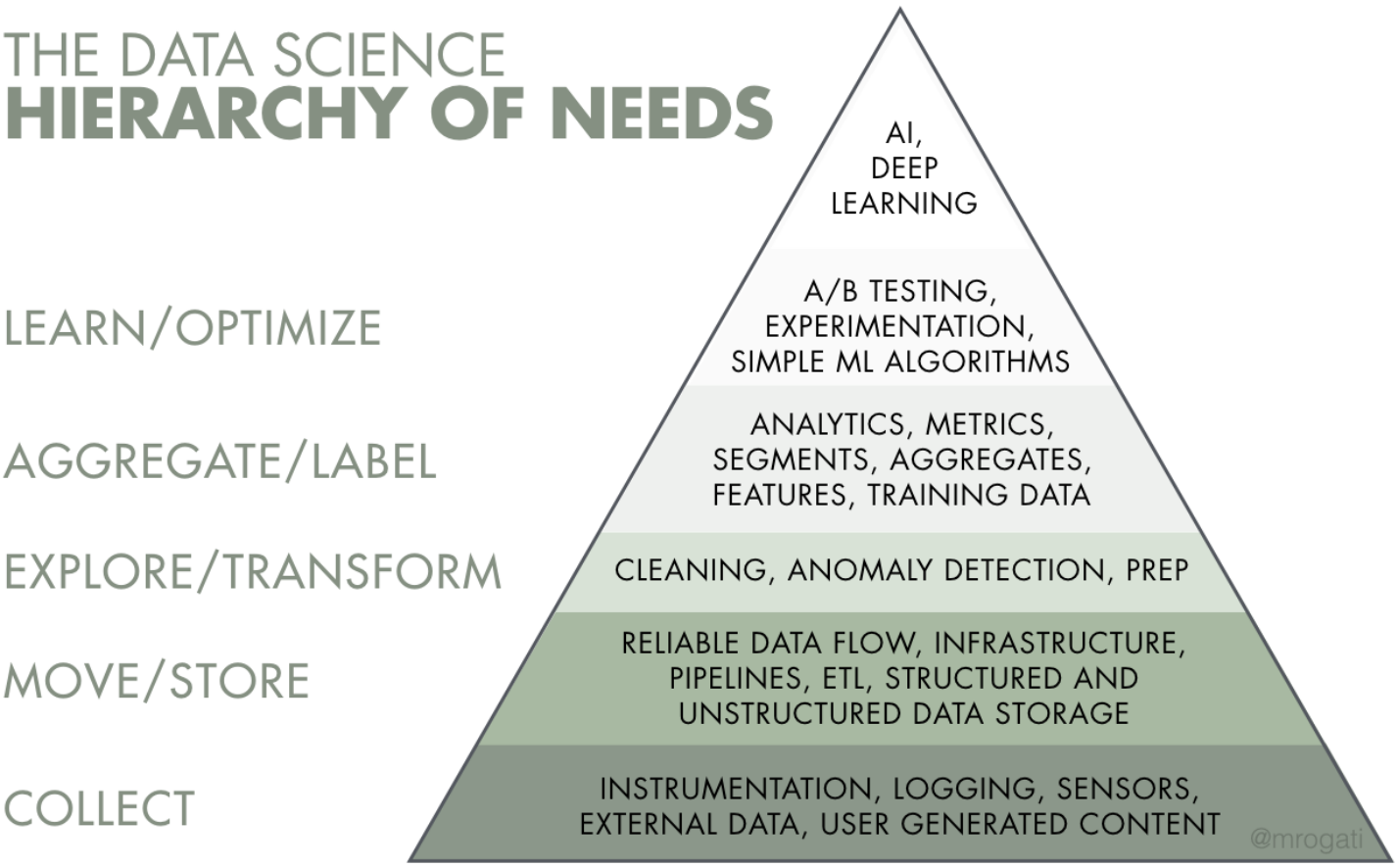
Что такое big data?

Dan Ariely
6 января 2013 г. ·

Big data is like teenage sex: everyone talks about it, nobody really knows how to do it, everyone thinks everyone else is doing it, so everyone claims they are doing it...



Пирамида AI


THE DATA SCIENCE
HIERARCHY OF NEEDS
LEARN/OPTIMIZE
AGGREGATE/LABEL
EXPLORE/TRANSFORM
MOVE/STORE
COLLECT
AI,
DEEP
LEARNING
A/B TESTING,
EXPERIMENTATION,
SIMPLE ML ALGORITHMS
ANALYTICS, METRICS,
SEGMENTS, AGGREGATES,
FEATURES, TRAINING DATA
CLEANING, ANOMALY DETECTION, PREP
RELIABLE DATA FLOW, INFRASTRUCTURE,
PIPELINES, ETL, STRUCTURED AND
UNSTRUCTURED DATA STORAGE
INSTRUMENTATION, LOGGING, SENSORS,
EXTERNAL DATA, USER GENERATED CONTENT
@mrogati

Кто такой data engineer?

Понемногу каждого:

- инженер

Кто такой data engineer?

Понемногу каждого:

- инженер
- архитектор

Кто такой data engineer?

Понемногу каждого:

- инженер
- архитектор
- программист

Кто такой data engineer?

Понемногу каждого:

- инженер
- архитектор
- программист
- devOps

Кто такой data engineer?

Понемногу каждого:

- инженер
- архитектор
- программист
- devOps
- администратор БД

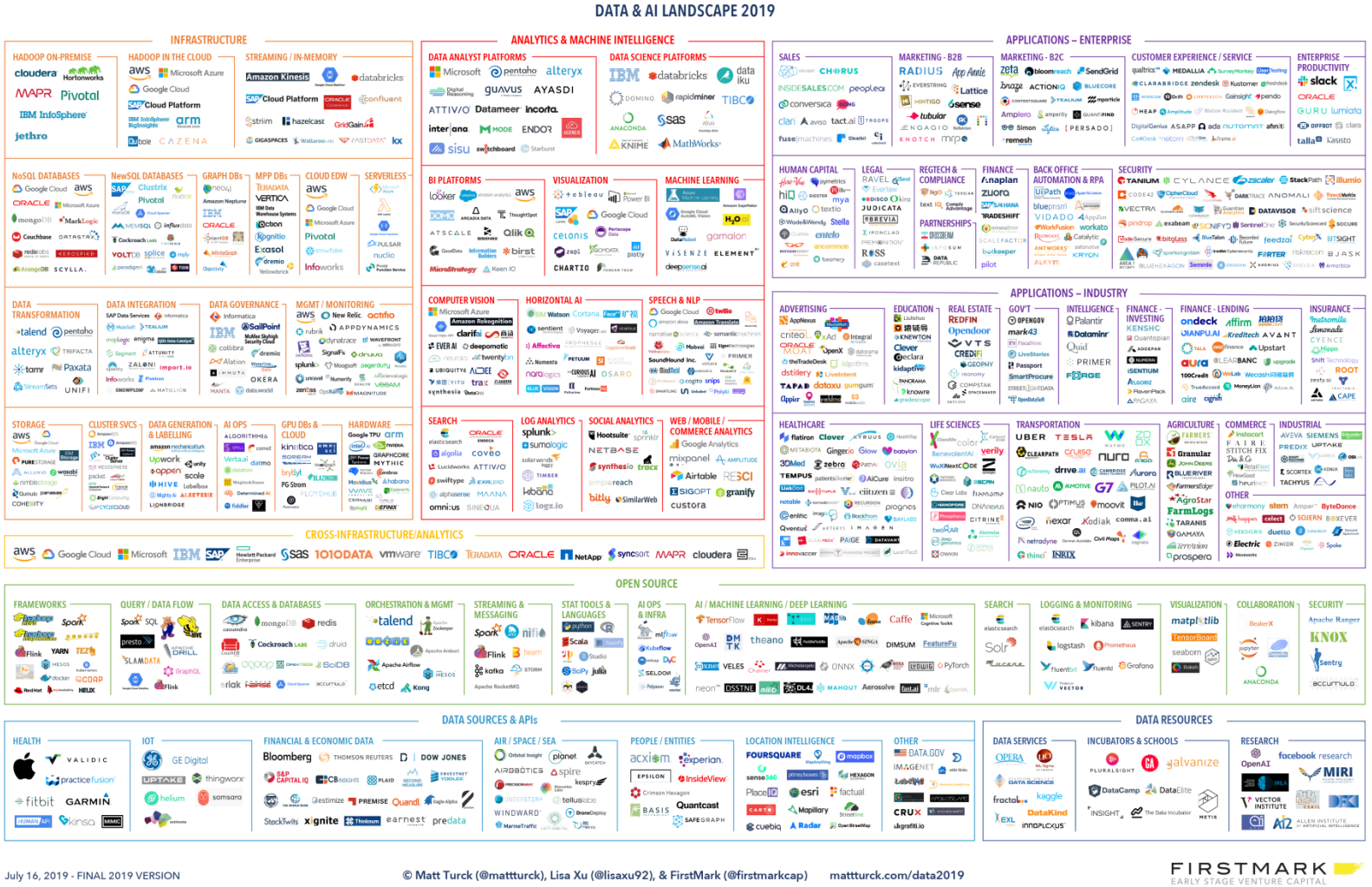
Кто такой data engineer?

Понемногу каждого:

- инженер
- архитектор
- программист
- devOps
- администратор БД

Делает «прод» для data scientist

Что надо знать?



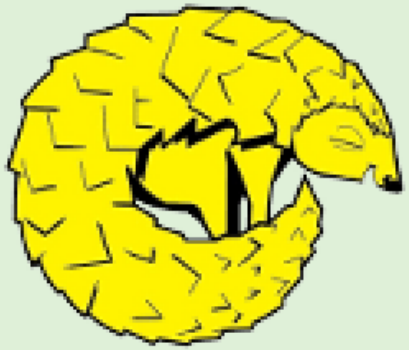
https://mattturck.com/data2019/

Технологий очень много

Azurill is Pokemon!

Azurill's tail is large and bouncy. Azurill can be seen bouncing and playing on its big, rubbery tail. On sunny days they gather at the edge of water and splash about for fun.

Pangool is Big Data!

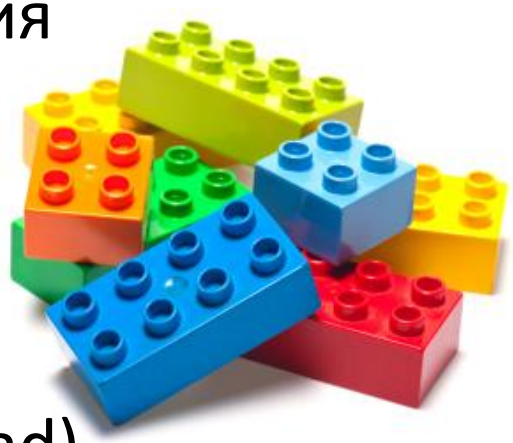


It's like Sandlash, but it will help us to develop map reduce jobs for Hadoop.

https://pixelastic.github.io/pokemonorbigdata/

Как быть?

- строить из готовых блоков
- понимать, какие блоки выбрать (какая решается задача), знать по 1-2 на каждый случай
- уметь формулировать требования
- понимать, что внутри
- не поддаваться хайпу
- правильно склеить
- не переусердствовать (см. left-pad)
- понимать, когда сделать свое/форк/плагин



Построение решения: хранилища

- реляционные БД – PostgreSql, H2, …
- NoSql – MongoDB, CouchDB, …
- key-value – Redis, Memcached, …
- полнотекстовый поиск – ElasticSearch, Solr, …
- колоночные БД – Cassandra, ClickHouse, …
- графовые БД – Neo4j, RedisGraph, …
- распределенные ФС – HDFS, DFS, Ceph, …
- очереди сообщений – Kafka, RabbitMQ, …
- …

Построение решения: инструменты

- пакетная обработка – Spark, Flink, …
- потоковая обработка – Spark Streaming, Kafka Streams, Samza, …
- MapReduce – Hadoop, Riak, …
- синхронизация состояния – Zookeeper, etcd, …
- управление ресурсами – YARN, Mesos,…
- …

База

- программирование – python/scala
- алгоритмы и структуры данных
- форматы данных – json, csv, avro, …
- SQL
- теория распределенных систем

Деплой

- CI/CD – Teamcity, Travis, Jenkins…
- развертывание – Puppet, Ansible, …
- контейнеры – Docker, Podman, …
- оркестрация – Kubernetes, Docker Swarm, …
- облака – AWS, Google Cloud, …
- логи – ELK stack, …
- метрики – TICK stack, …
- …

Заключение

- от прототипа до готового качественного продукта на проде – большая пропасть
- много «вспомогательного кода»
- там тоже есть интересные и сложные проблемы
- эти проблемы для дата сайентистов обычно решают дата инженеры
- зоопарк технологий, все надо знать хотя бы по чуть-чуть

Литература

- Типичные проблемы кода, который работает с ML: https://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf
- Kleppmann. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems
- Обзорная статья про то, что нужно знать дата инженеру https://khashtamov.com/ru/data-engineer/
- Интерактивная карта технологий Big Data http://xyz.insightdataengineering.com/blog/pipeline_map/
- Big data & AI landscape https://mattturck.com/data2019/
- Покемон или бигдата: https://pixelastic.github.io/pokemonorbigdata/

Спасибо за внимание!