Cookbook for Building a DS Team

Ronert Obst

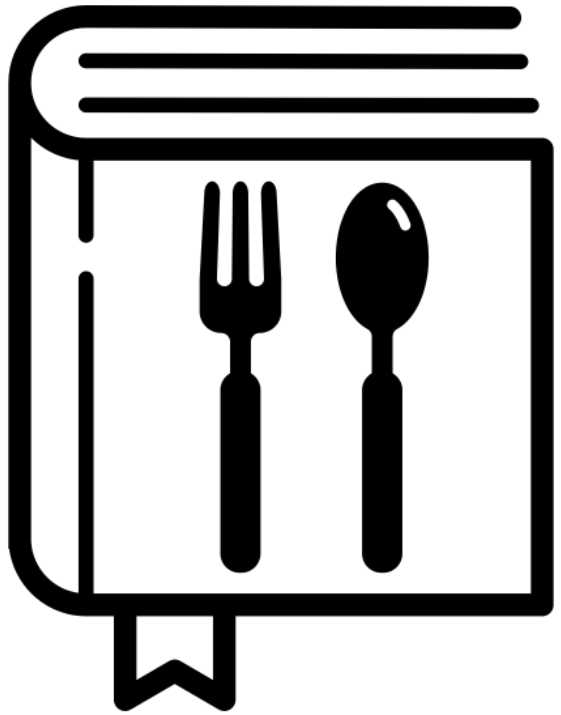
Introduction

- German fashion retailer
- Established 1971
- More than 1000 stores
- 40 countries, 4 continents
- Over 16 000 employees
- Over 850 000 square meters of retail space
- 37 million customers per month

Cookbook for Building a DS Team

Built a data science team of 12 in a year with services running in production.

1. Hiring
2. Tech Stack
3. Building the Right Things
4. Practices
5. What DS expect from companies



Hiring a DS Team

Getting the People

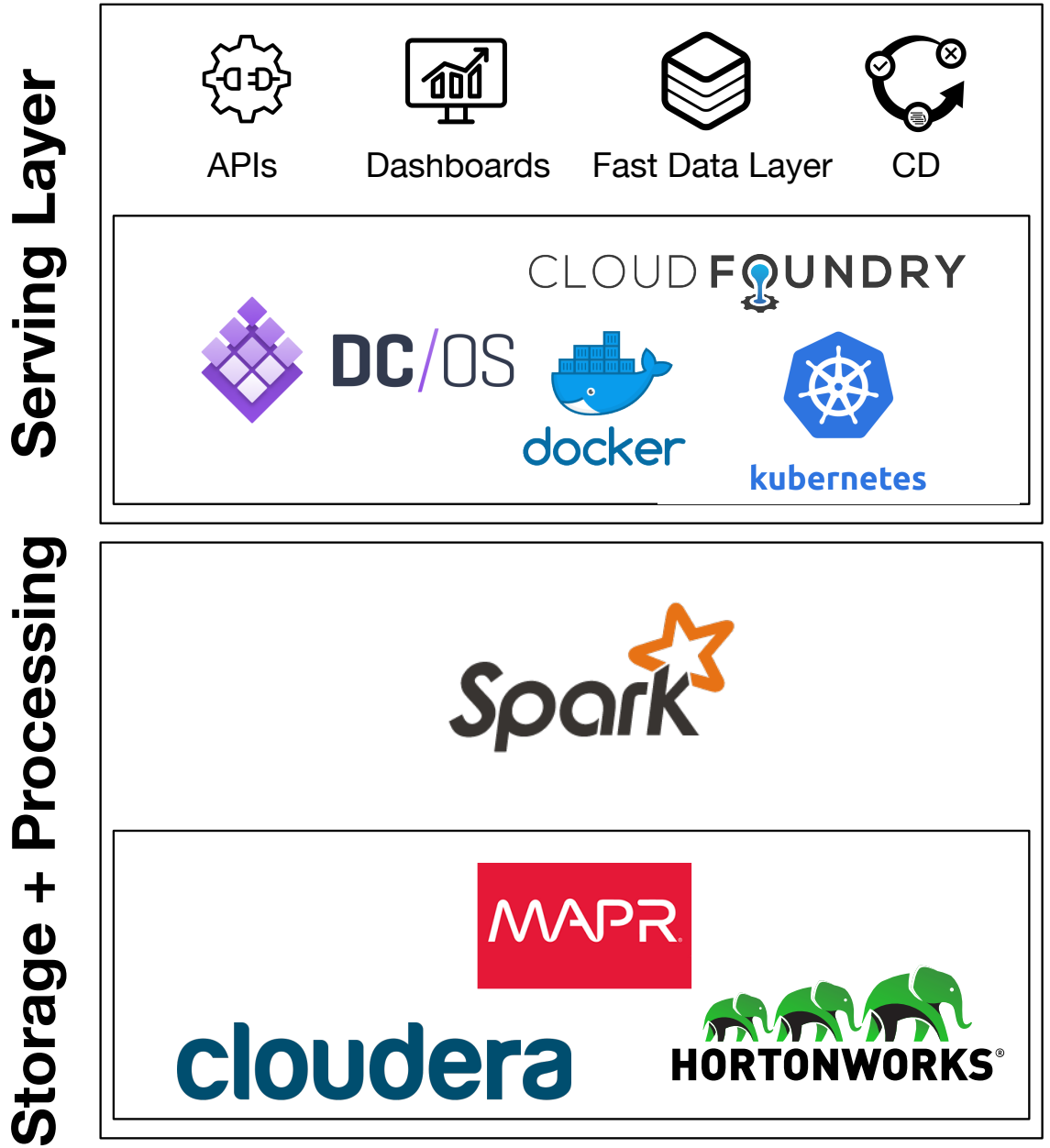
- Job Ads
  - Easy and relatively cheap
  - Lots of unqualified candidates
- Direct Search
  - Checked over 1000 profiles on LinkedIn + Xing
  - Low response rates
  - Lots of work
- Head Hunters
  - High success rate
  - Very expensive
- DSR
  - Excellent for motivated junior DS
  - By definition no senior people
  - No Data Engineers

- Over 100 phone interviews to hire 12 people
  - Technical phone screen
  - Programming exercise
  - Onsite Interview
  - Given a CV looks good and you invite a candidate for a phone screen, roughly only 1/10 – 2/10 are successful
- If you don't already have a good data scientist, try to get outside help for interviewing
- It is easy to hire incompetent people that look good on paper
- 2/3 Data Scientists, 1/3 Data Engineers is a good mix
- Get a dedicated DevOps/SysAdmin

Choosing the Right Tech

Tech Stack

- On-premise vs. cloud
- (Usually) Hadoop vendor
  - Ingest and store data
  - Run Spark for data munging and feature engineering
  - Maybe some SQL queries
- Container orchestration framework for O16N
  - Kubernetes in its various incantations
  - DC/OS
  - Cloud Foundry
  - Docker Datacenter
- Think about where and how to integrate GPUs



Building the Right Thing

Discovery and Framing

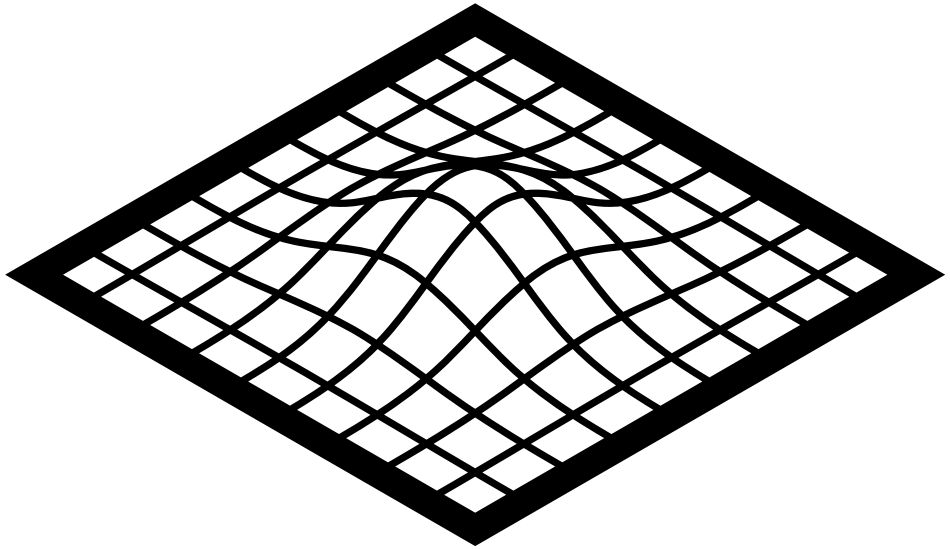
- Often it is not clear what needs to be built
- Come up with use cases together with the business
- Talk to people and figure out the pain points
- Also need vision
- Rank the use cases together with the business in two dimensions
  - Impact
  - Technical feasibility
- Define clear project goals

Extract, Clean and Understand the Data

- Extract, Load, Transform
- Data are almost never clean and well documented
- Need to talk to a lot of people to even understand everything
- In Kaggle and university you get a nice dataset to start modeling with straight away
- In reality you get 50 tables distributed over 5 different databases with no data dictionary and only a partial ERD
- Be prepared to decipher cryptic variable names and become a data janitor and data detective

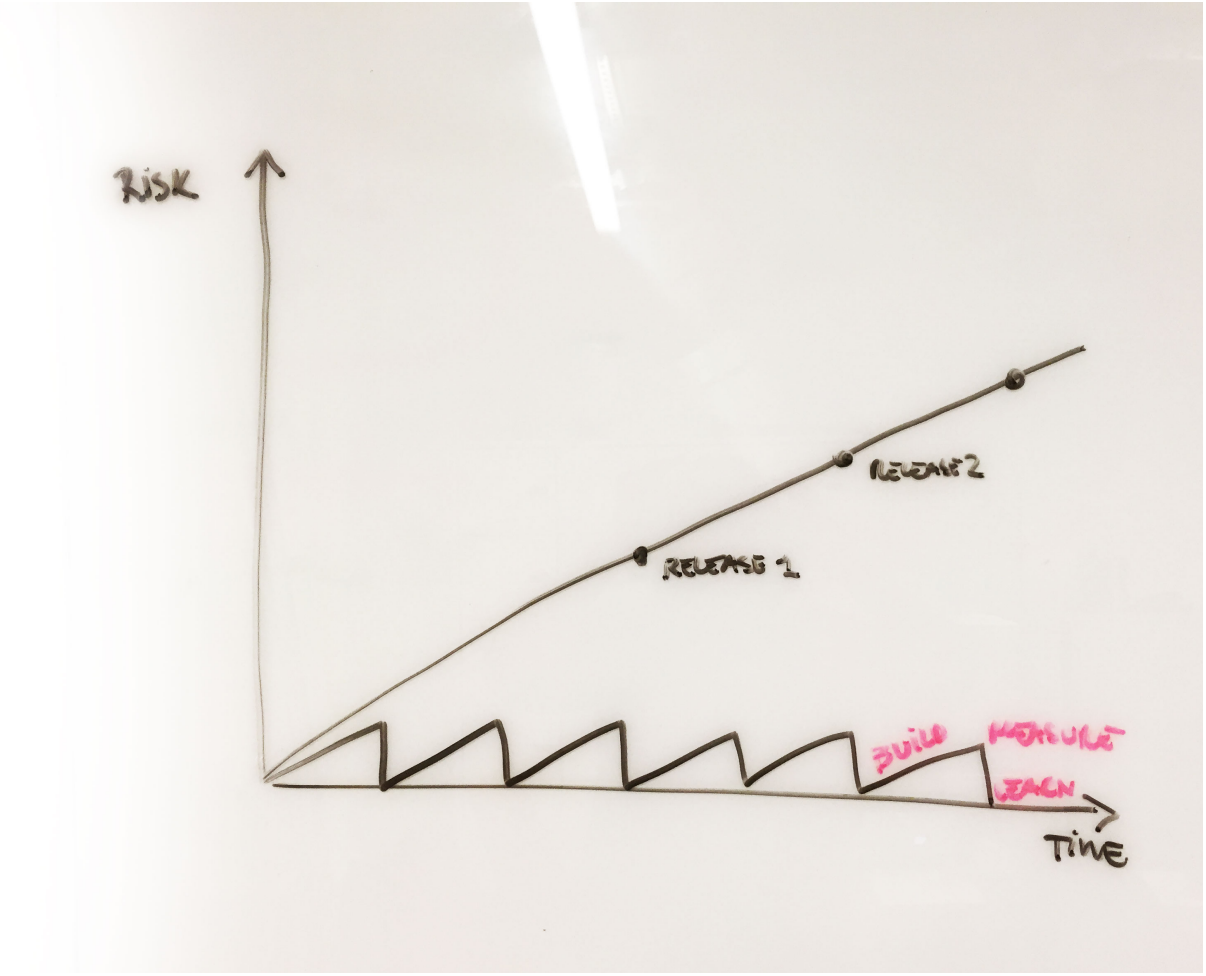
Modelling

- ~ 20 % of your time
- Lots of feature engineering
- for p in problems:
  - Build simple base line model
  - Afterwards try xgboost
  - Then import keras
- If it is some specialised task like image recognition where deep learning is known to outperform then
  - Figure out human error as a base line
  - Use a pre-trained model (Inception, ResNet etc.)
  - Do transfer learning
- Some exceptions
- Optimisation and Operations Research often missing



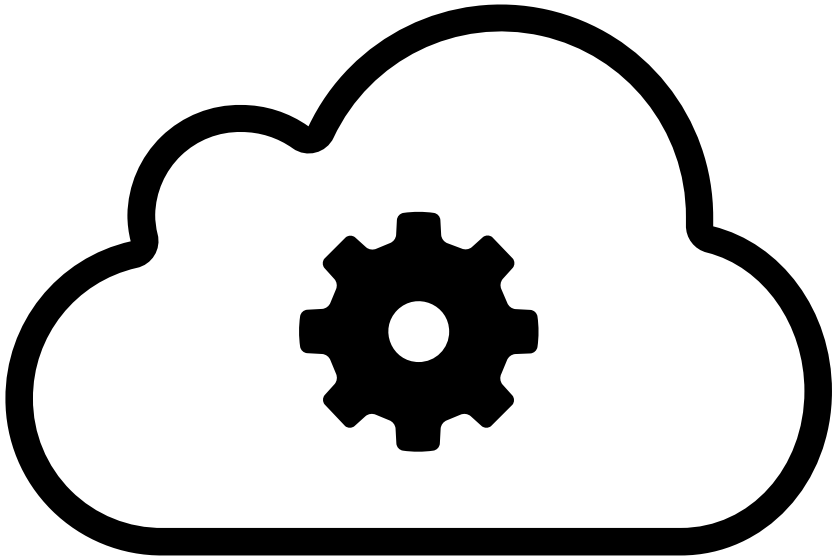
Present your Results!

- Results have to be presented frequently to management and business stakeholders
  - Visualisations
  - Good explanations for non-experts
  - Show relevance
- Understand company politics
- Keep in touch
- Frequent user tests to get feedback early



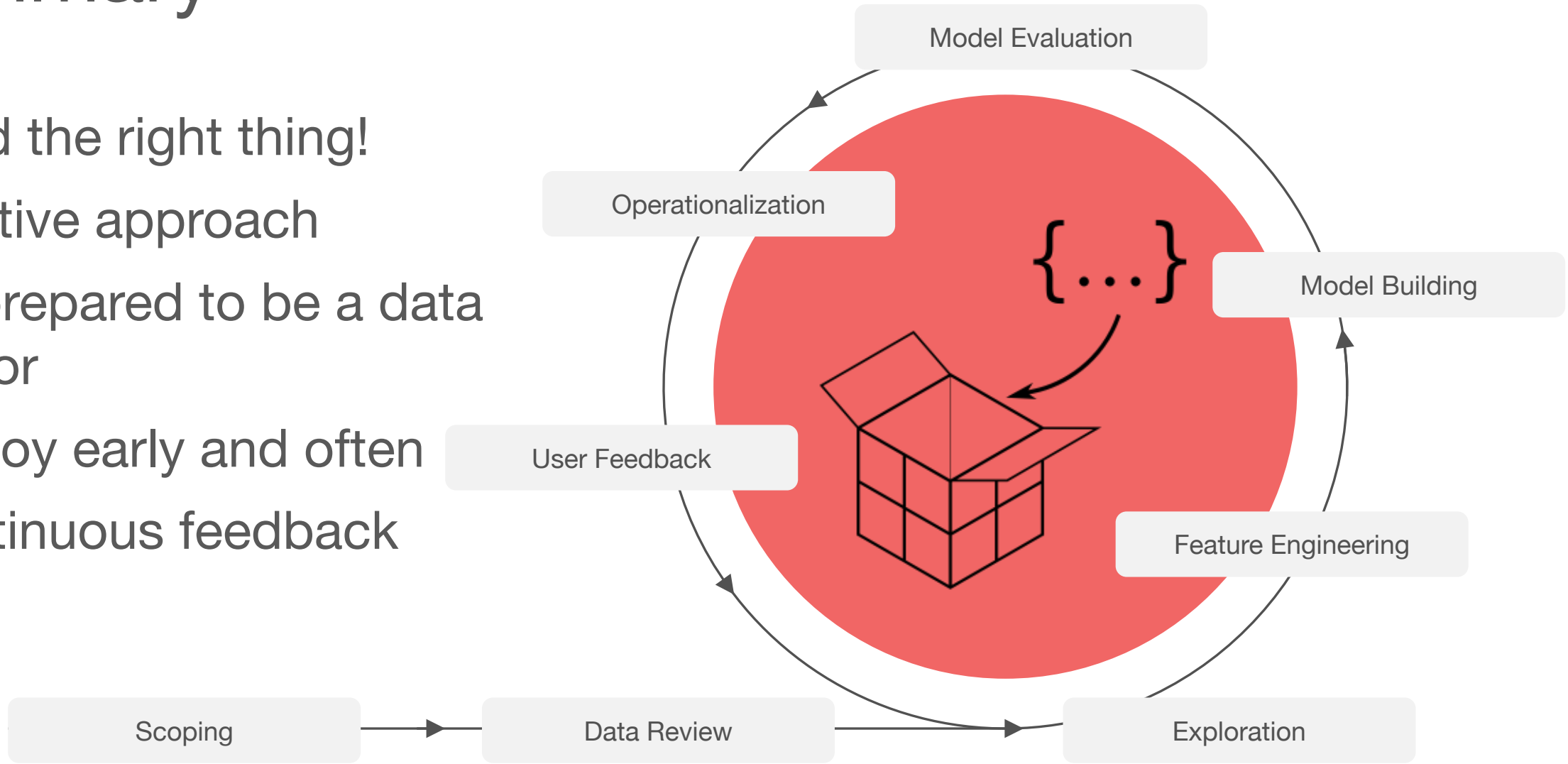
Operationalisation

- Reliable data pipelines
- Batch vs. real-time
- APIs
  - Request-Response
  - Streaming (Pub-Sub)
- Smart Apps
- Dashboards
- Health monitoring
- Logging
- Performance monitoring
- A/B-Tests



Summary

- Build the right thing!
- Iterative approach
- Be prepared to be a data janitor
- Deploy early and often
- Continuous feedback



Practices to Follow

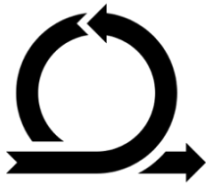
Practices from Software Engineering

Pair Programming

Test Driven Development

Standups

Retros



Continuous Integration /
API First

Project Management

Learn from Software Engineers

- Documentation, documentation, documentation
- Tests, tests, tests
- Follow style guides and clean code principles
- Code reviews
- 2+ people on a project
- Use git properly
- Use a pipeline management tool like Luigi/AirFlow etc.
- 12-factor apps for O16N
- Knowledge sharing process is critical

What DS Expect from Companies

What DS Expect from Companies

- Interesting technologies and problems
- Lunch and learns
- Team chat (Slack)
- Frequent 1-1s
- “Nice” offices
- Lots of whiteboard space
- Ergonomic furniture
- Home office
- Clear career paths
- Competitive salaries