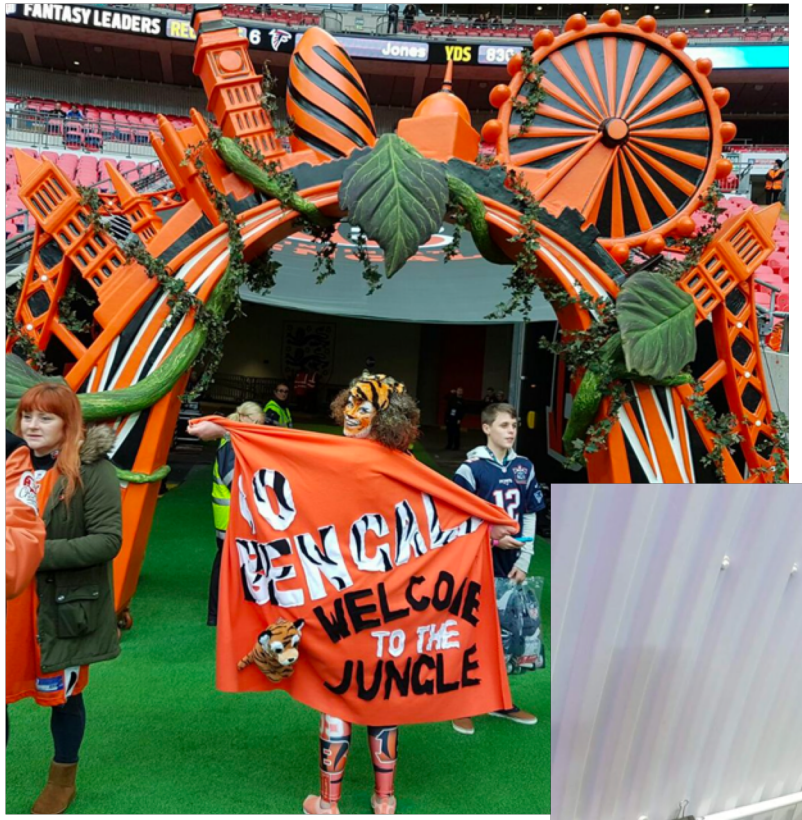
HELLOSODA

Shipping a Machine Learning model to Production; is it always smooth sailing?

LEANNE FITZPATRICK | HEAD OF DATA

I <3 Data

- Head of Data at Hello Soda
- Previously used C++, SAS, Python
- Avid fan of American Football
- Enjoy trying to get a handicap in golf(!)
- Enjoy beer and whisky (and have started writing about beer!)



JBCA mlCon

THE BUSINESS

Hello Soda was born out of a sense of frustration with the amount of poor business decisions being made based on traditional, unsophisticated and impersonal data.

Our services are successfully utilised across a wide range of industries including insurance, finance, travel and gaming.

Through explicit consumer consent, Hello Soda harnesses unstructured data to provide businesses with usable insights into consumers and their behaviour.

Hello Soda removes the disparity between available data and decision making processes of businesses through the use of advanced analytic techniques.

Machine Learning In Production…?

- Traditional Data Science
  - Prototyping
  - Explorative
  - Functional
  - Ease for data scientist
- Solutions to machine learning problems readily solvable with exploratory functionality
- More and more courses, materials and resources encourage data science
- 3rd parties offerings with full functionality in the last 1-2 years

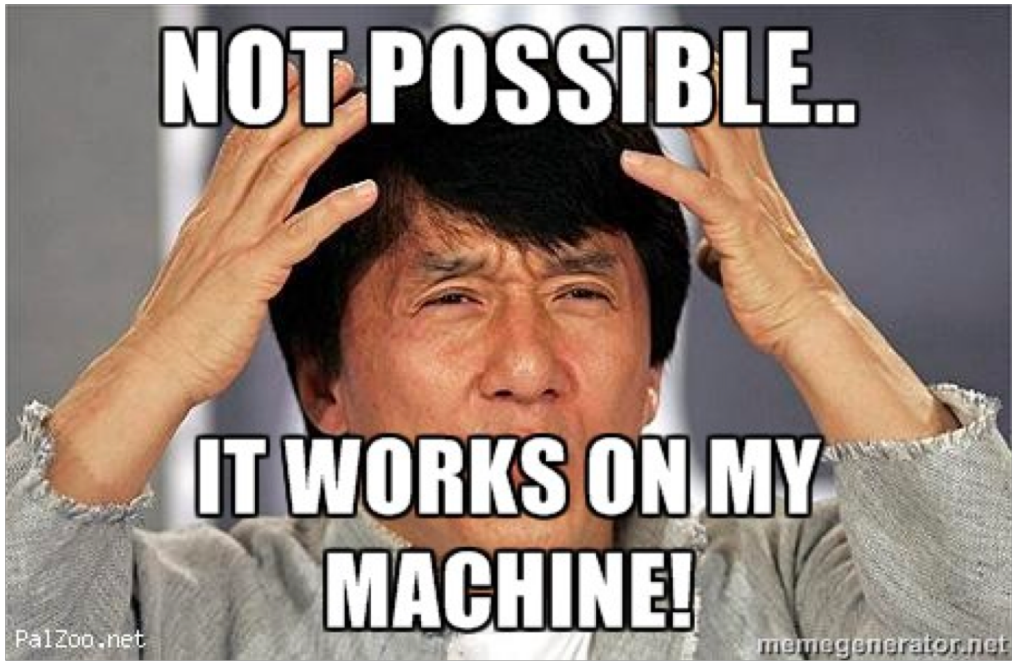
Occasional reaction from engineers…

Problem: Machine Learning Stack

Is varied, fairly vast, and forever growing

- Open Source
- Continually growing
- Specific tools, specific task
- Puzzle of solutions
- Suitability to data, paradigms, resource

JBCA mICon



Problem: Realizing Value

A perfect model, but can it be used?

- Standalone models may be ok but…
- How will your product will be consumed
- Will it stay relevant?
- Deprecation is a risk
- Forever re-training
- Manual, very manual
- Inconsistencies

Machine Learning in Production:
...What is Production Ready?

Depends who you ask

https://softwareengineering.stackexchange.com/questions/61726/define-production-ready

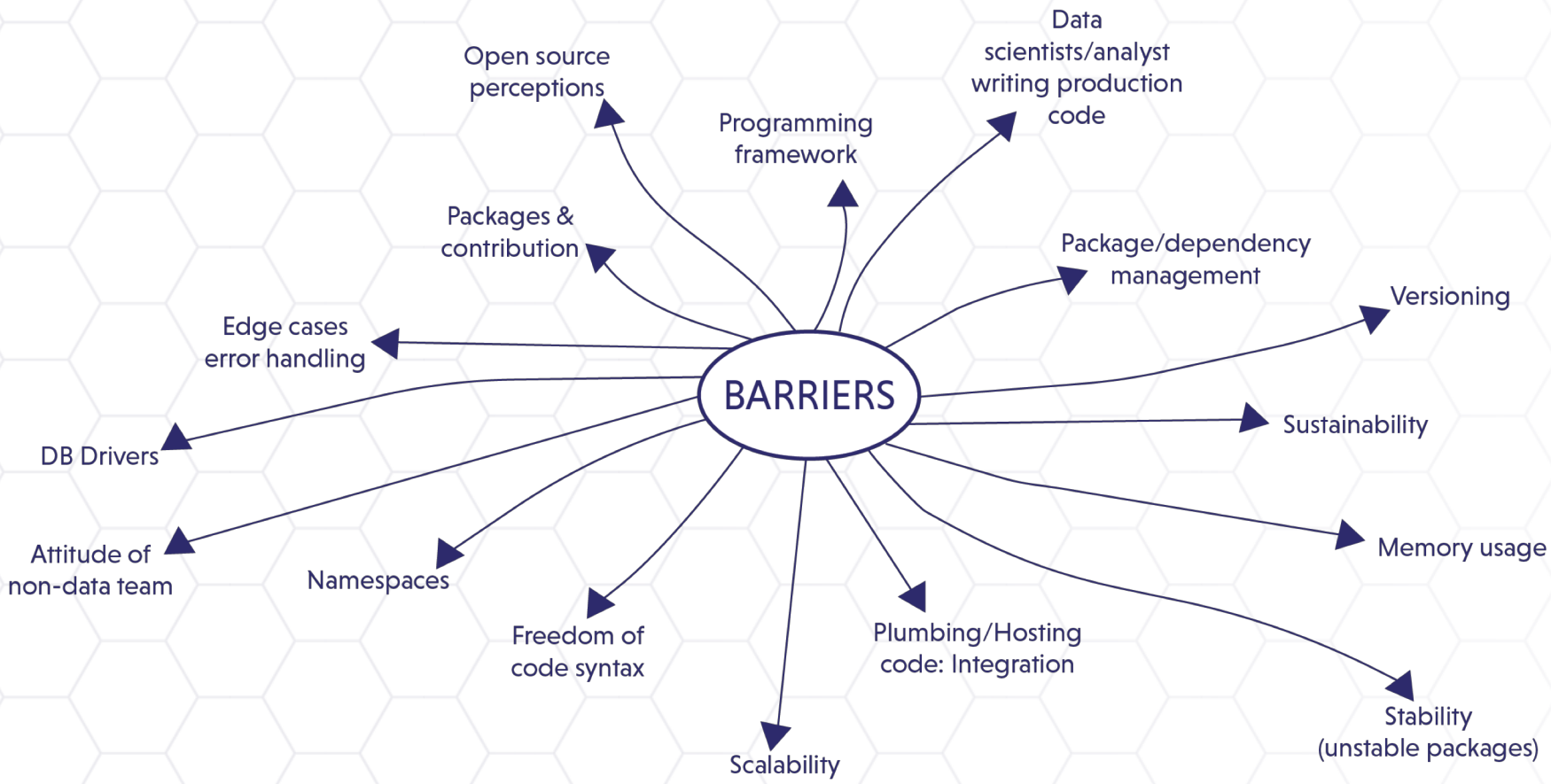
Programmer's definition of "production-ready":

- it runs
- it satisfies the project requirements
- its design was well thought out
- it's stable
- it's maintainable
- it's scalable
- it's documented

Management's definition of "production-ready":

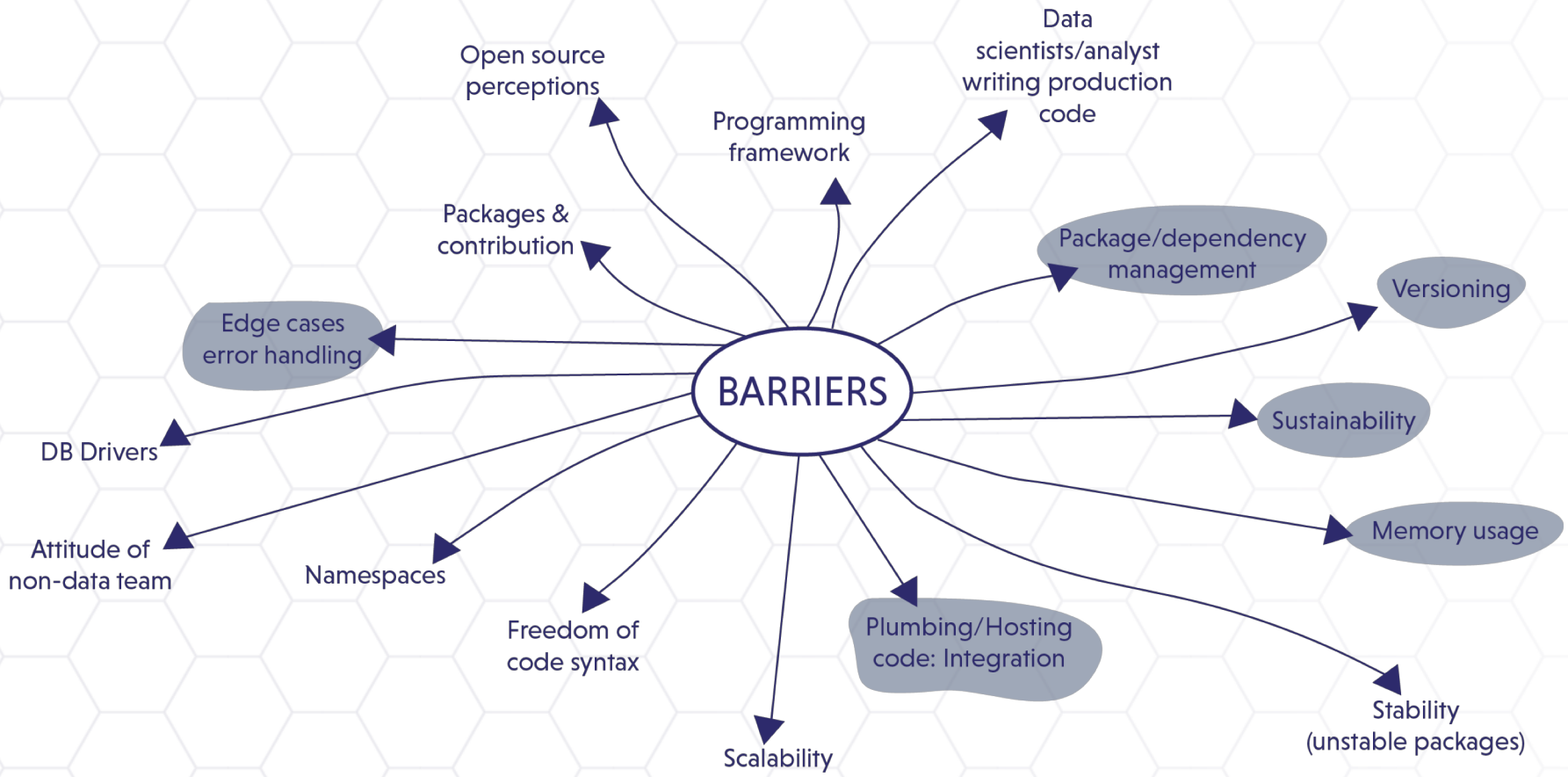
- it runs
- it'll turn a profit

Barriers to Entry: Data Science in Production



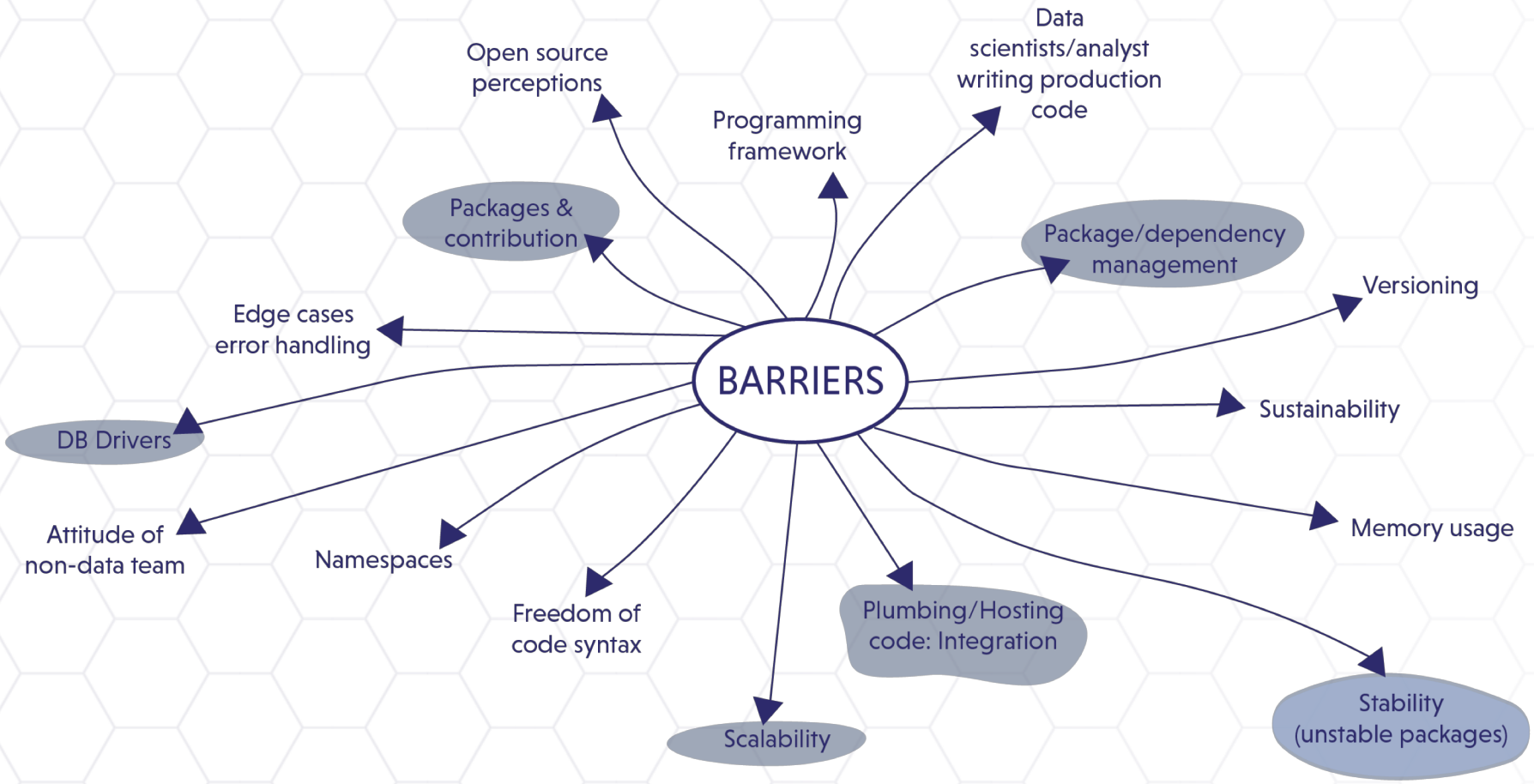
Barriers to Entry: Data Science in Production

"ENGINEERING" BARRIERS



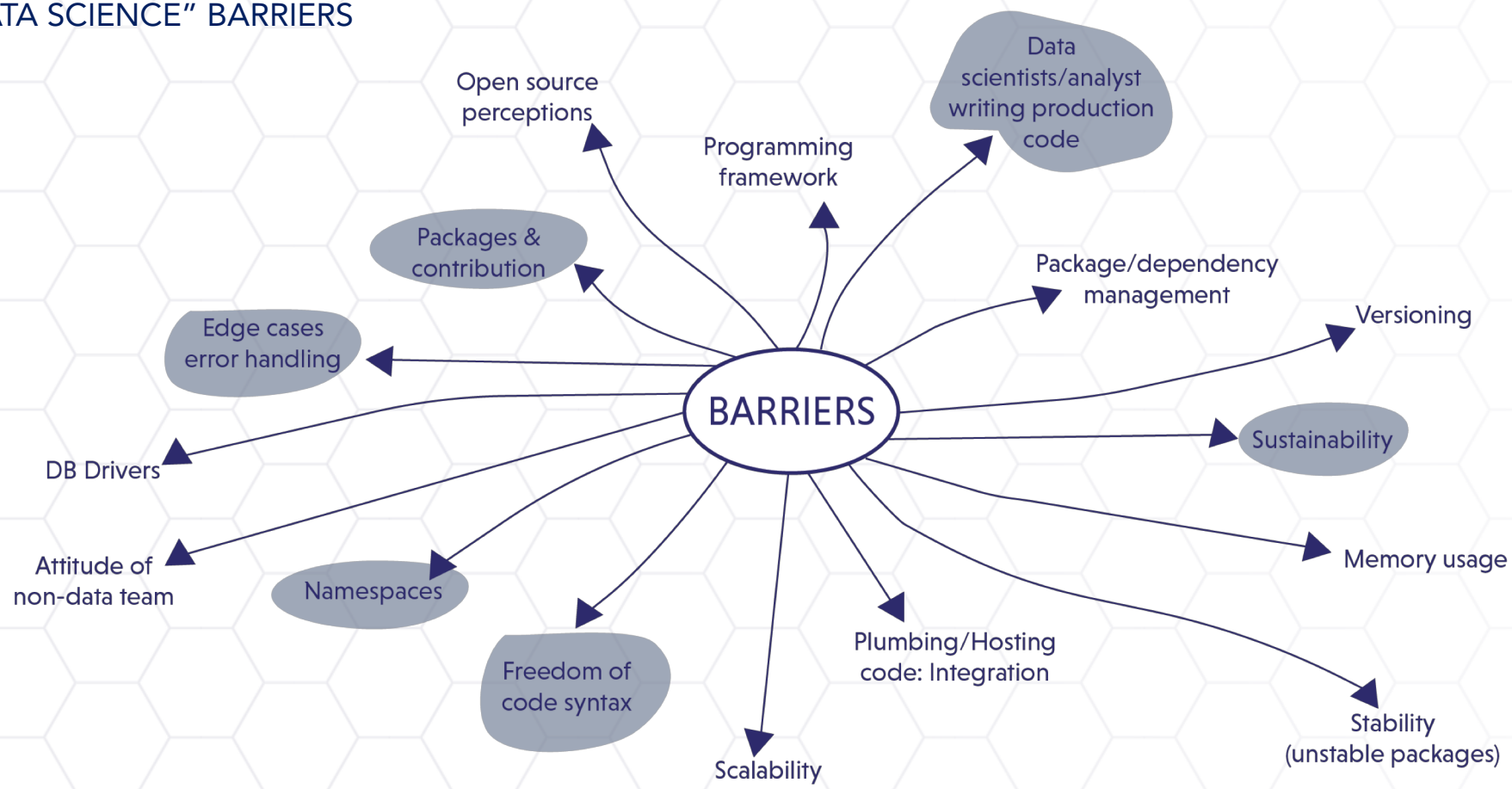
Barriers to Entry: Data Science in Production

"INFRASTRUCTURE" BARRIERS



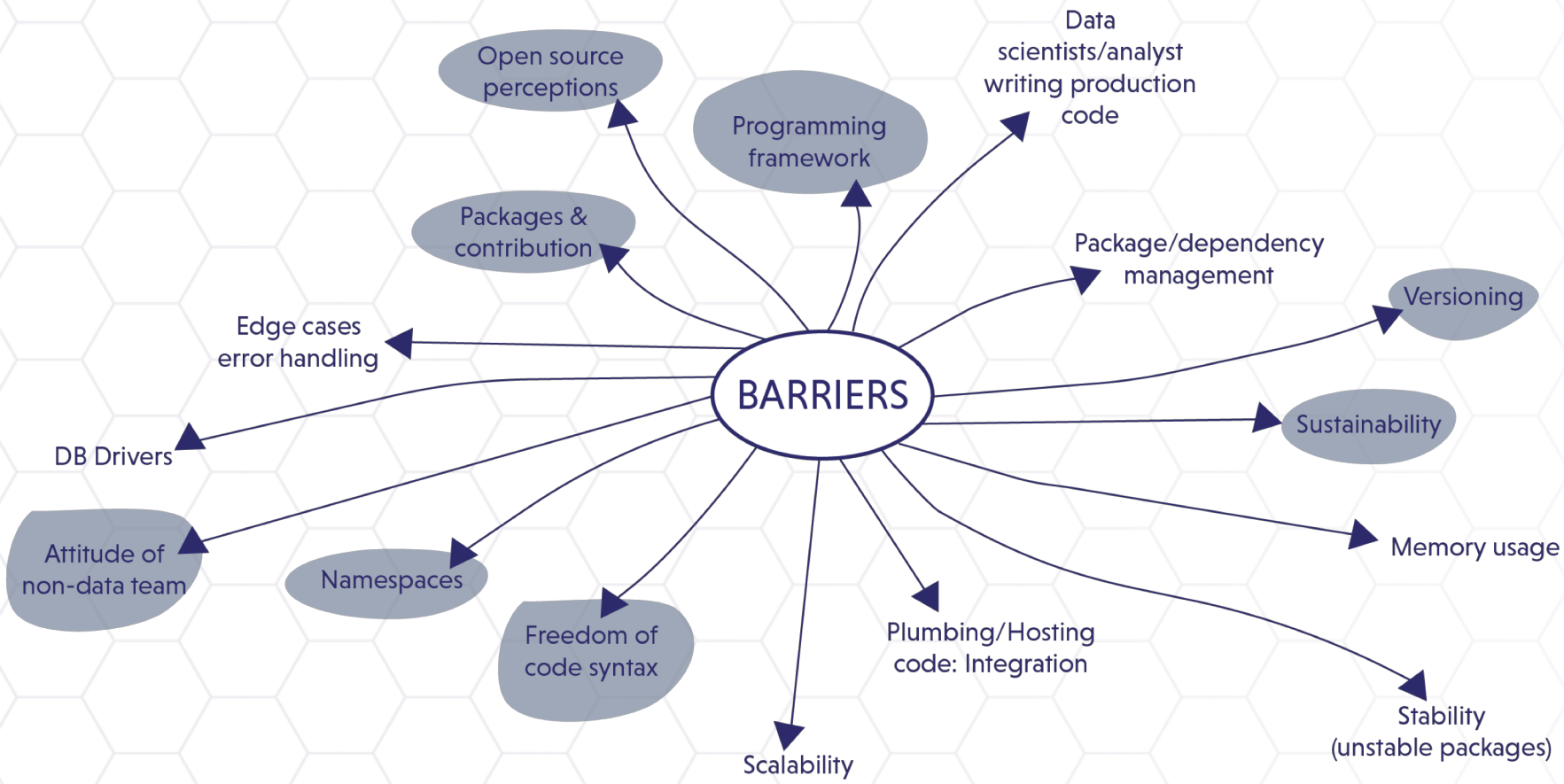
Barriers to Entry: Data Science in Production

"DATA SCIENCE" BARRIERS



Barriers to Entry: Data Science in Production

"CULTURAL" BARRIERS

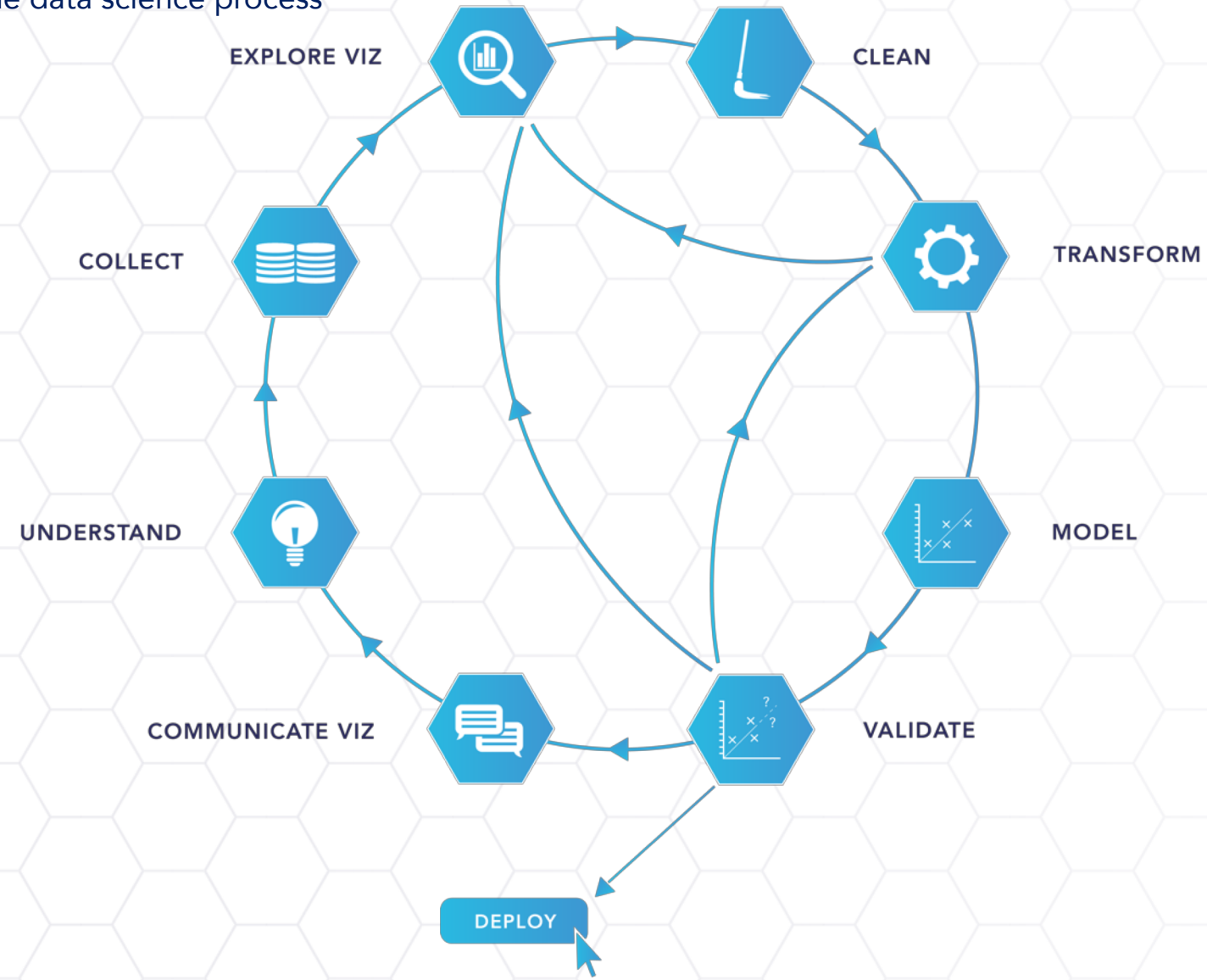


Overcoming Barriers:

… The Hello Soda Data Journey

Barrier: Deployment

Central to the data science process



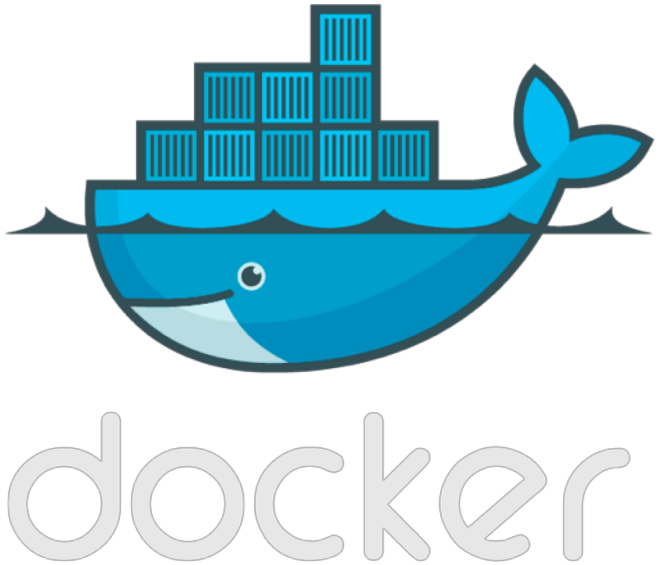
Barrier: Deployment

Central to the data science process

Solution: Docker

Barrier: Development Environment

- Docker Image contains the dependencies and environment requirements
- Underlying concept: Containers
- Container: boxes of self-contained software
- Containerization: boxing up the software
- Used for:
  - Set up and distribution of tools and software
  - Sharing reproducible analysis and code via Docker Images
  - Sharing applications directly
- Deployment as microservice



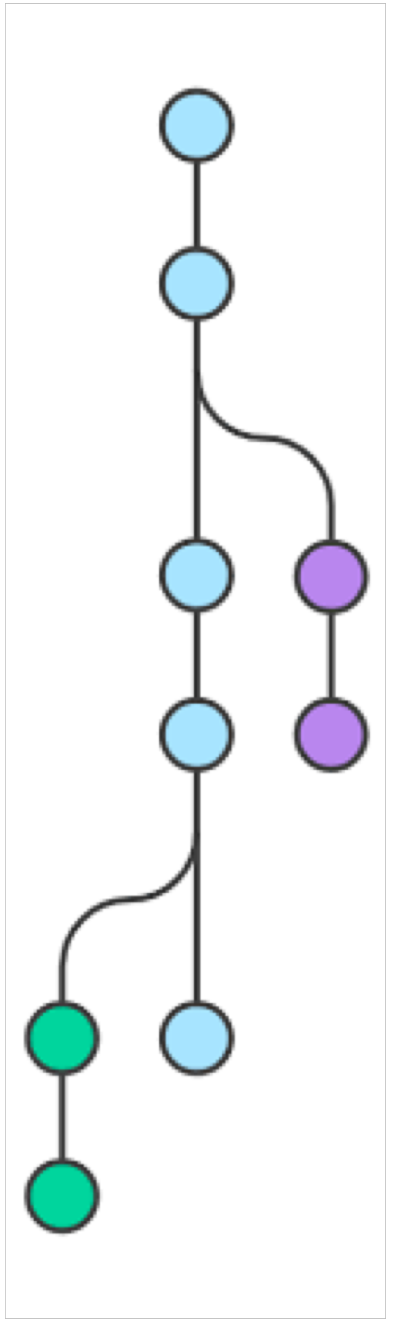
Solution: Code as a Service with Tornado

Barrier: Plumbing / Integration

- Wrap code as http endpoints; serving prediction scripts
- Expose prediction scripts as a REST API without any changes to the data scientist workflow
- Standardised framework with a wrapper outside of the model code
- JSONlite as JSON deserialiser to pass data to prediction script
- Could use flask, plumber, native code to API

Solution: Version Control

- Not expected to be a software engineer but…
- Your code is part of your data / machine learning product
- Model versioning
- Sharing
- Legacy
- Improvements
- GitFlow for Data Science



JSON

Copy JSON

```json
{
  "json": [
    "rigid",
    "better for data interchange"
  ],
  "yaml": [
    "slim and flexible",
    "better for configuration"
  ],
  "object": {
    "key": "value",
    "array": [
      {
        "null_value": null
      },
      {
        "boolean": true
      },
      {
        "integer": 1
      }
    ]
  },
  "paragraph": "Blank lines denote\nparagraph breaks\n",
  "content": "Or we can auto\nvert line breaks\nsave space"
}
```

YAML

Copy YAML

```yaml
---
# <- yaml supports comments, json does not
# did you know you can embed json in yaml?
# try uncommenting the next line
# { foo: 'bar' }

json:
  - rigid
  - better for data interchange
yaml:
  - slim and flexible
  - better for configuration
object:
  key: value
  array:
    - null_value:
    - boolean: true
    - integer: 1
paragraph: >
  Blank lines denote

  paragraph breaks
content: |-
  Or we
  can auto
vert line breaks
save space
```

```yaml
$schema: 'http://json-schema.org/draft-04/schema#'
type: object
additionalProperties: false
required:
  - parameters
  - data
  - format
properties:
  parameters:
    type: object
    required:
      - clientId
    properties:
      clientId:
        type: string
      productId:
        type: string
      weights:
        type: object
        required:
          - name
          - surname
          - email
          - birthdate
          - taggedLocation
          - declaredLocation
        properties:
```

Solution: Schemas with YAML

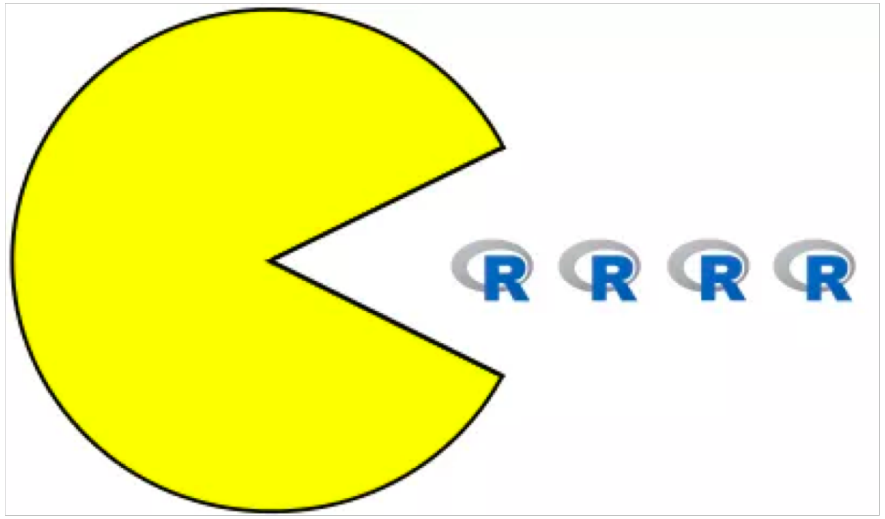
Barrier: Edge Cases, Error Handling & Versioning

- JSON data schema defined as YAML
- Data Scientist generates schema for request & response of prediction API
- Written as YAML (previously as JSON Schema)
- Validated by the code wrapper upon request / response call
- Enforces mindset around a release framework and functionality
- Future: Data Validation

Solution: Package Management

Barrier: Package and Dependency Management

- Options with pip env, packrat and others
- Ensure libraries captured not just in requirements file but exact versioning and dependency
- Differentiate between development environment / exploratory / production
- Leakage prevention
- Open Source “registry”



Solution: Cookie Cutter

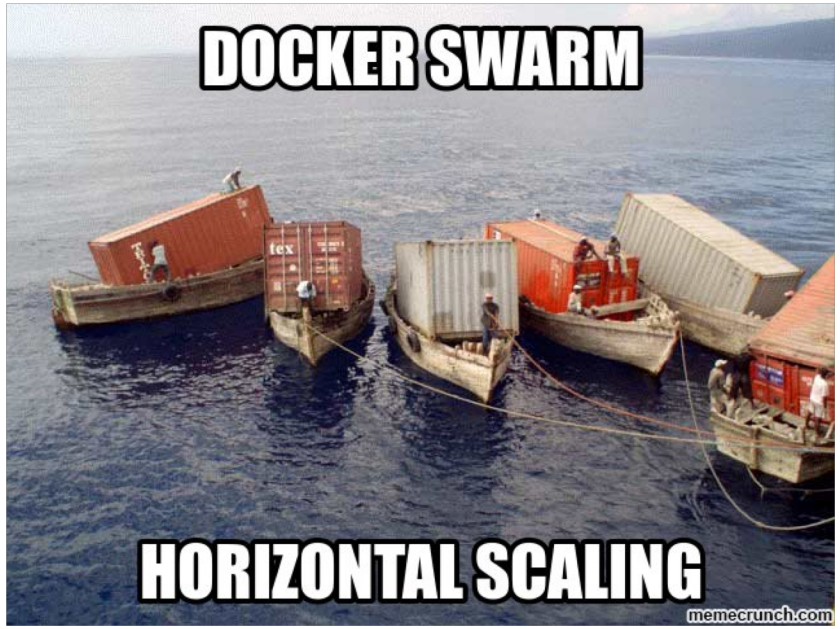
Blocker: Reproducible Framework

- Project Template that goes beyond a folder structure
- Can service rather than copy & paste
- Can handle common mistakes or misconceptions
- Framework for successful project
- Use a standard, then adapt your own
- Example: http://projecttemplate.net/index.html

Solution: Testing & CI

Barrier: Stability & Error Handling

- testthat and pytest
- Continuous Integration framework
- Alternatively, engineer to develop a test framework
- Ensure common *data* mistakes captured in testing
- Predictive model tests vs unit tests vs integration tests vs resource tests
- Test cluster and replica sandbox!



Solution: Don't worry about scaling… yet

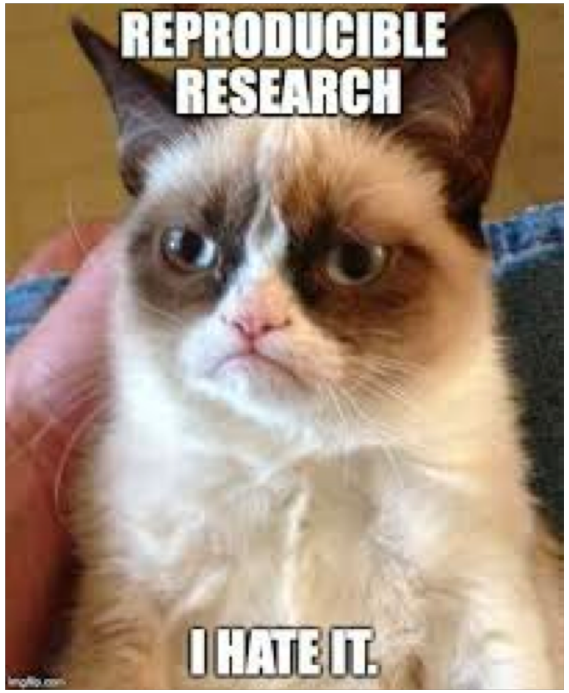
Barrier: Scalability

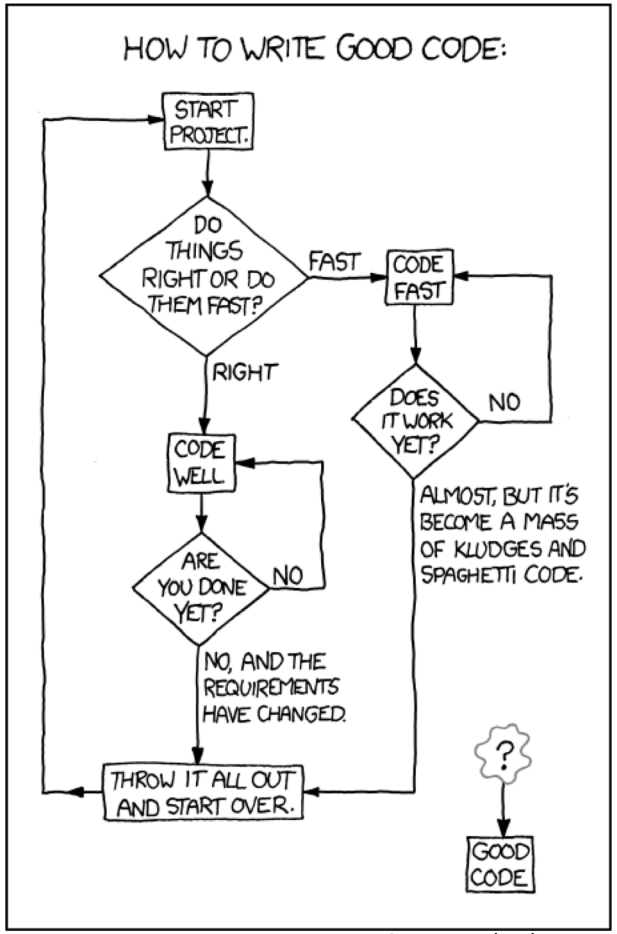
- "Solve first, scale later"
- Ease of scaling via Docker
- Containers of "solutions" can be replicated to manage load
- Kubernetes to manage scaling (previously Amazon ECS)
- No longer technically a "problem for the Data Scientist"

Solution: Collaboration

Barrier: Culture

- Iterative Prototyping; it's a mindset
- Version Control; getting the core principles right
- Support Network; code reviews, style guides
- Documentation
- Package recognition
- Empower and enablement
- Singular Team mentality; business goals >> team goals
- Skill depth; sharing





Source: xkcd.com

SUMMARY

BLOCKERS:

- “Engineering”
- “Infrastructure”
- “Data Science”
- “Cultural”

COLLABORATION IS KEY:

- Allow Data Scientists and Analysts to solve data problems in a native way
- Think solution, then scale
- Not just a data science problem, think about who else can help! (team!)

QUESTIONS

https://www.eventbrite.co.uk/e/all-the-women-in-tech-christmas-bonanza-tickets-51544619344