



GOPHERCONF POLAND 2019

GDANSK — 27 JUNE

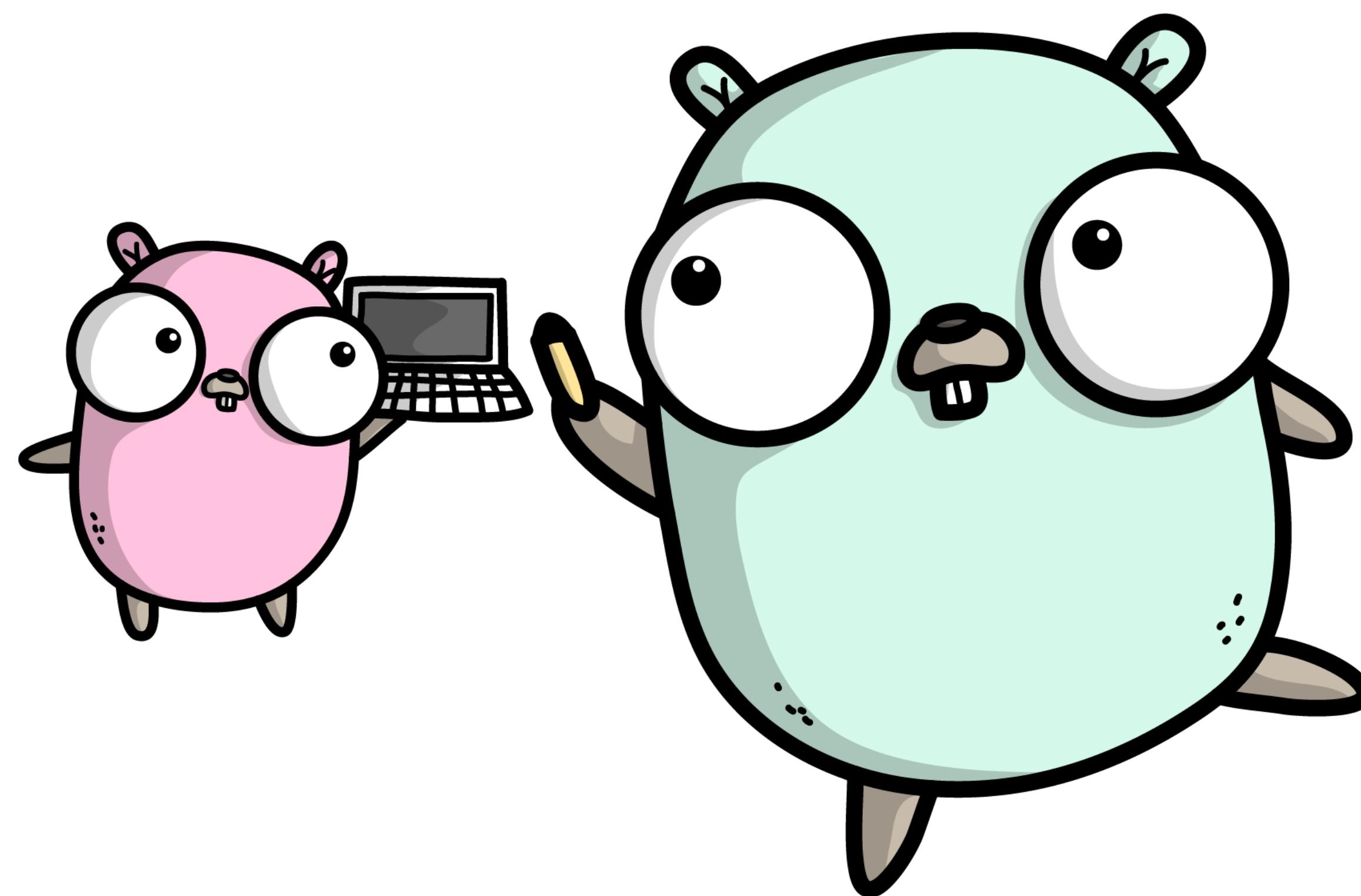


Wojciech Barczyński [Head of Engineering]

[SMACC.io](https://smacc.io) | [Hypatos.ai](https://hypatos.ai)

Gdańsk, 27 July 2019

Golang



What did it teach me? And what was next.

Beginnings

- from Berlin to Indonesia
- Successful mobile e-commerce
- Legacy developed in death march 💀



Plan

- New features in Golang
- Move to Kubernetes (GCP)
- Get team on board



WE MOSTLY LIKE

- Standard library
- Composition
- ...
- Idiomatic way
- go proverbs



Simplicity

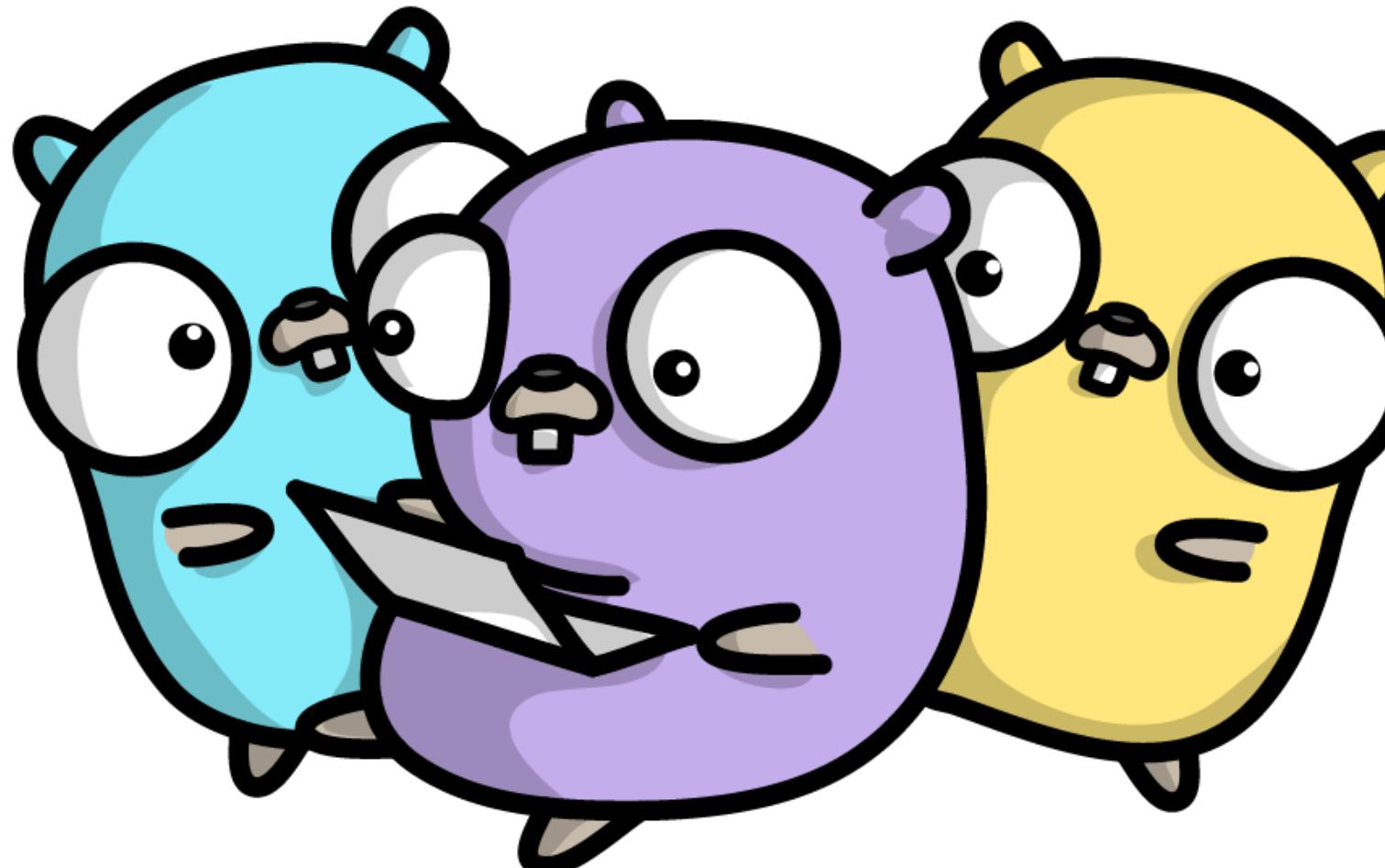
- Simplicity = low cognitive load
- Easy to start
- Increase readability

Low Cognitive

- People better feel about tech
- Faster adoption
- Less resistance

Golang culture

- Ruby?
- Docker?
- ...
- Golang



Redundant code

- A fair price fighting complexity
- 9 years with `err != nil`
- Repeat yourself (in tests) by Hashicorp

Mitchell Hashimoto - Advanced Testing with Go

Experience-driven

- Observation what works what not
- With Proof-of-Concepts
- Less dogmatic

Next stop

SaaS and ML Fintech

- [SMACC.io](#)
- [Hypatos.ai](#)

New lessons

- Simple (common) project structure works the best
- Review libraries before using them
- Have internal training

Culture

Golang is simple.
So we dive directly into talking about
problems to solve.

Borrowed from [Tomasz Grodzki](#)

Culture shock

- No mocks?
- No DSLs
- No lib for X
- JSON parsing :)
- Not flexible



Golden mean

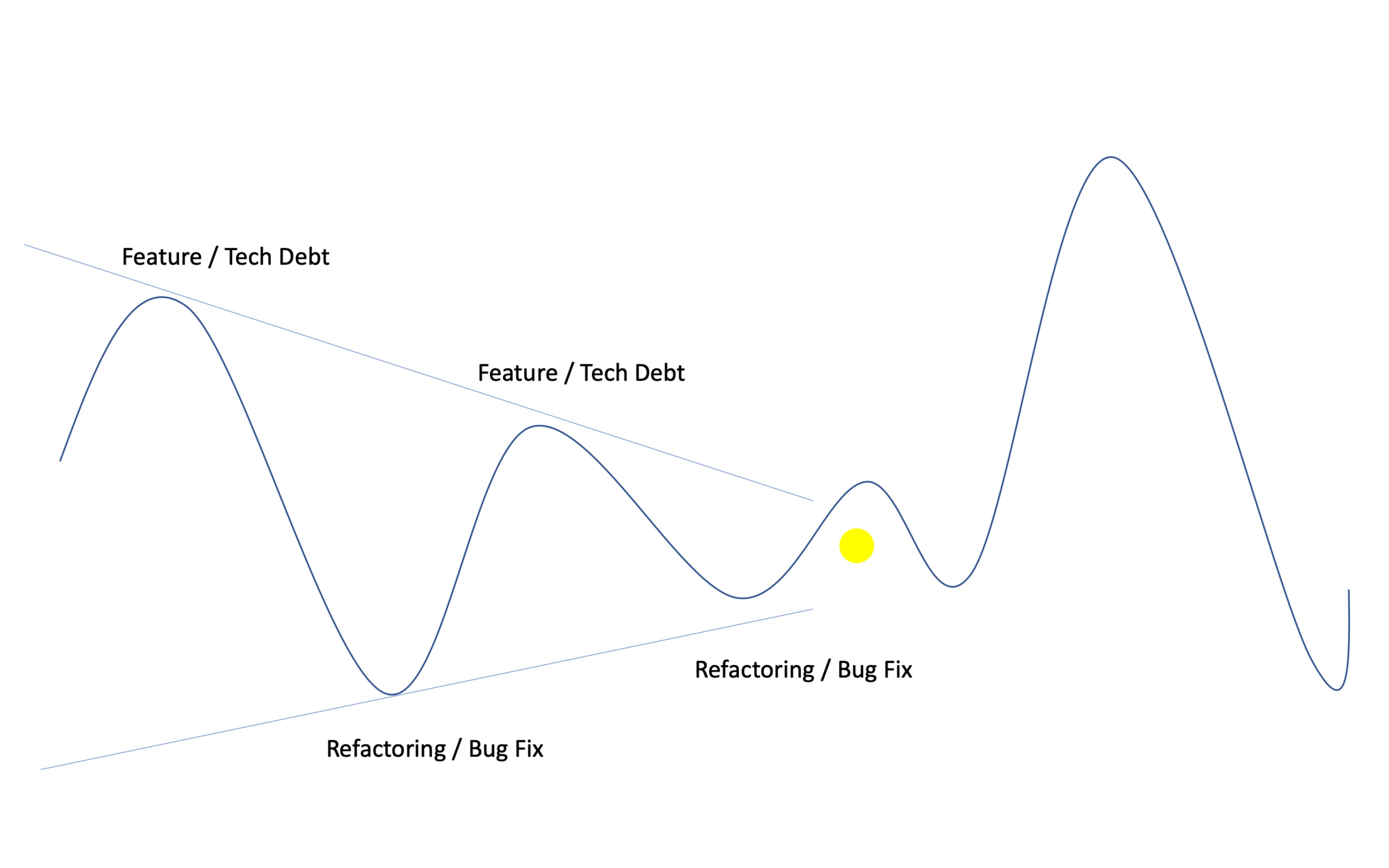
1. Guidelines,
not rules

[[Dave Cheney](#)]

2. Hard to find
3. It moves all the time



https://www.flickr.com/photos/bruno_brujah/

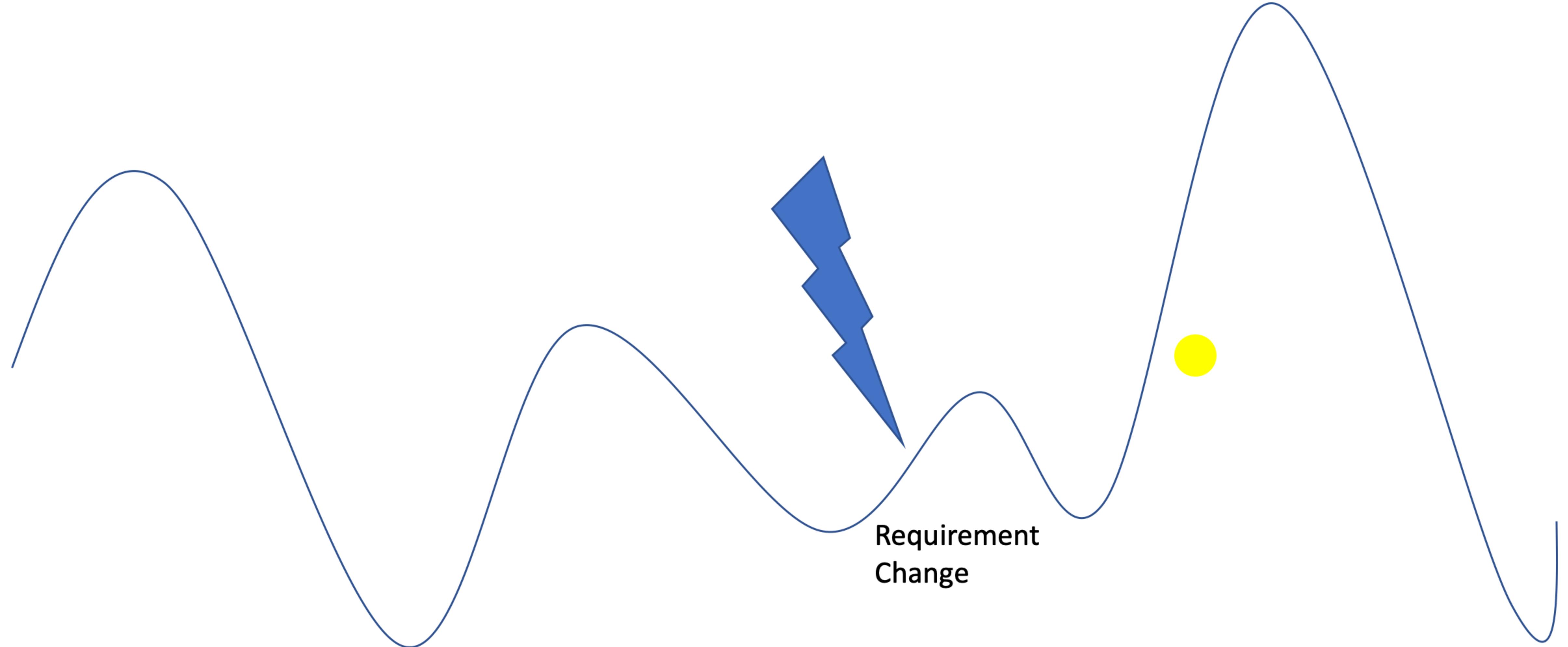


Feature / Tech Debt

Feature / Tech Debt

Refactoring / Bug Fix

Refactoring / Bug Fix



Requirement
Change

How to find it?

1. Team
2. Inclusion
3. Diversity



https://www.flickr.com/photos/bruno_brujah/

Methods?

- One-on-ones
- Engineering hour
- Moderate discussions
- Communication is everybody responsibility

More inspiration: [golang conduct](#)

Stuck? Let us know

- Lower the threshold
- Time-boxing
- Trigger-based [1]

[1] Cooper Code

Assume good intentions

ALWAYS

**Strong opinions
weekly hold**

Ship it

1. Ship it
2. ...
3. The rest
4. Finding balance

Kick Butt, Have Fun
Sun Microsystems

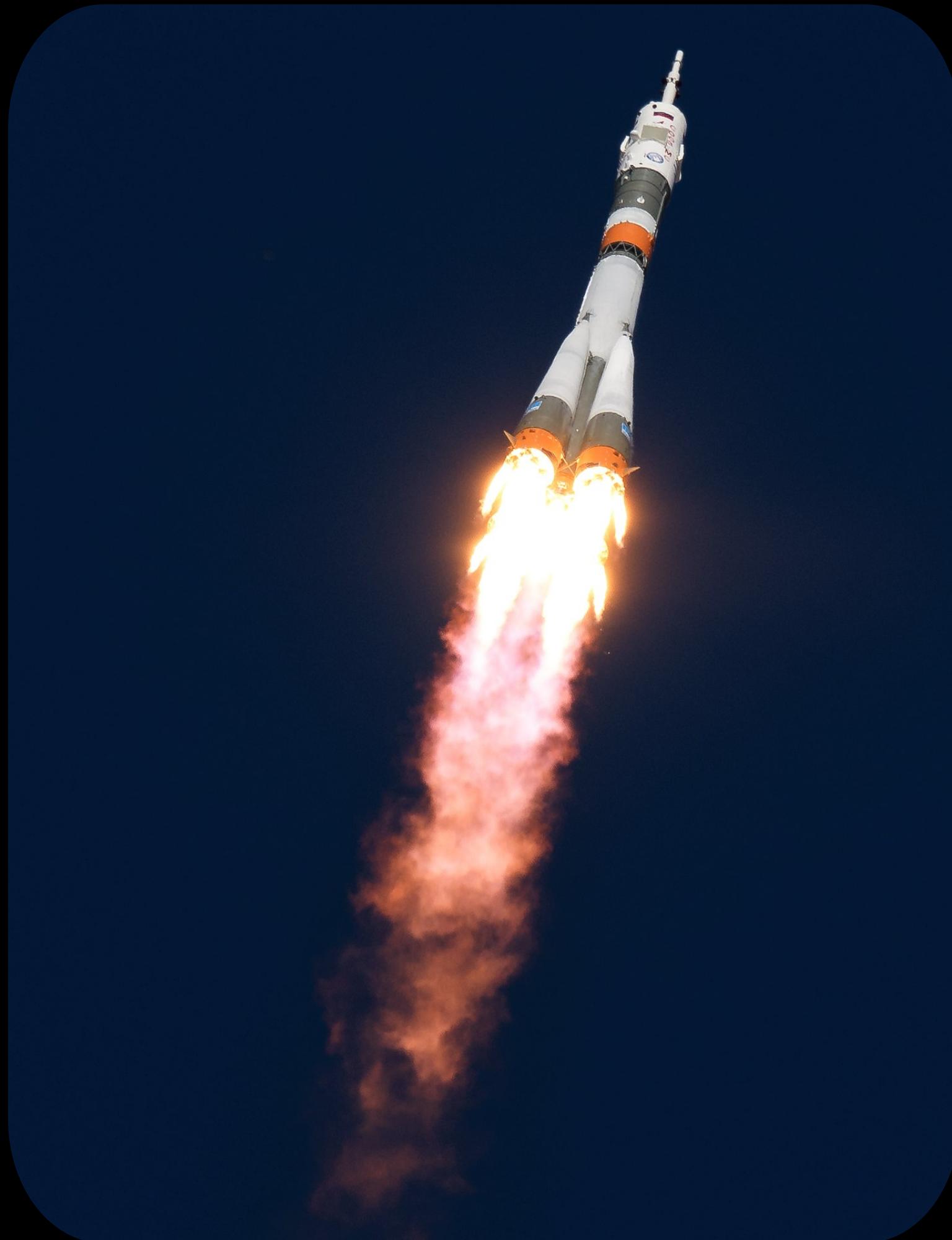


Photo by Nasa

Ship it

1. Cut the scope
2. Cut the quality
3. Delegate sth to the floor

version 1 → version 2

- v1 = fast, ~MVP
- v2 = right

Deliver! Patch. Patch

[Kate Matsudaira's blog](#)

version 1

- Understand
- Validate
- Avoid over-engineering

version 2

- Restructure
- Fix
- Re-factor

version 1 → version 2

- Will take more time
- Find answers faster
- Might give us more time

Idea?

- Demonstrate it without negative effect on your work
- Make it Real!

[Heroku Values](#)

Problem?

- come with a possible solution

- Powerful language
- and tools (k8s, cloud)
- **combined experience**



https://www.flickr.com/photos/bruno_brujah/

Get know each other

- Talk with people around you

Example of good questions

THANK YOU. QUESTIONS?

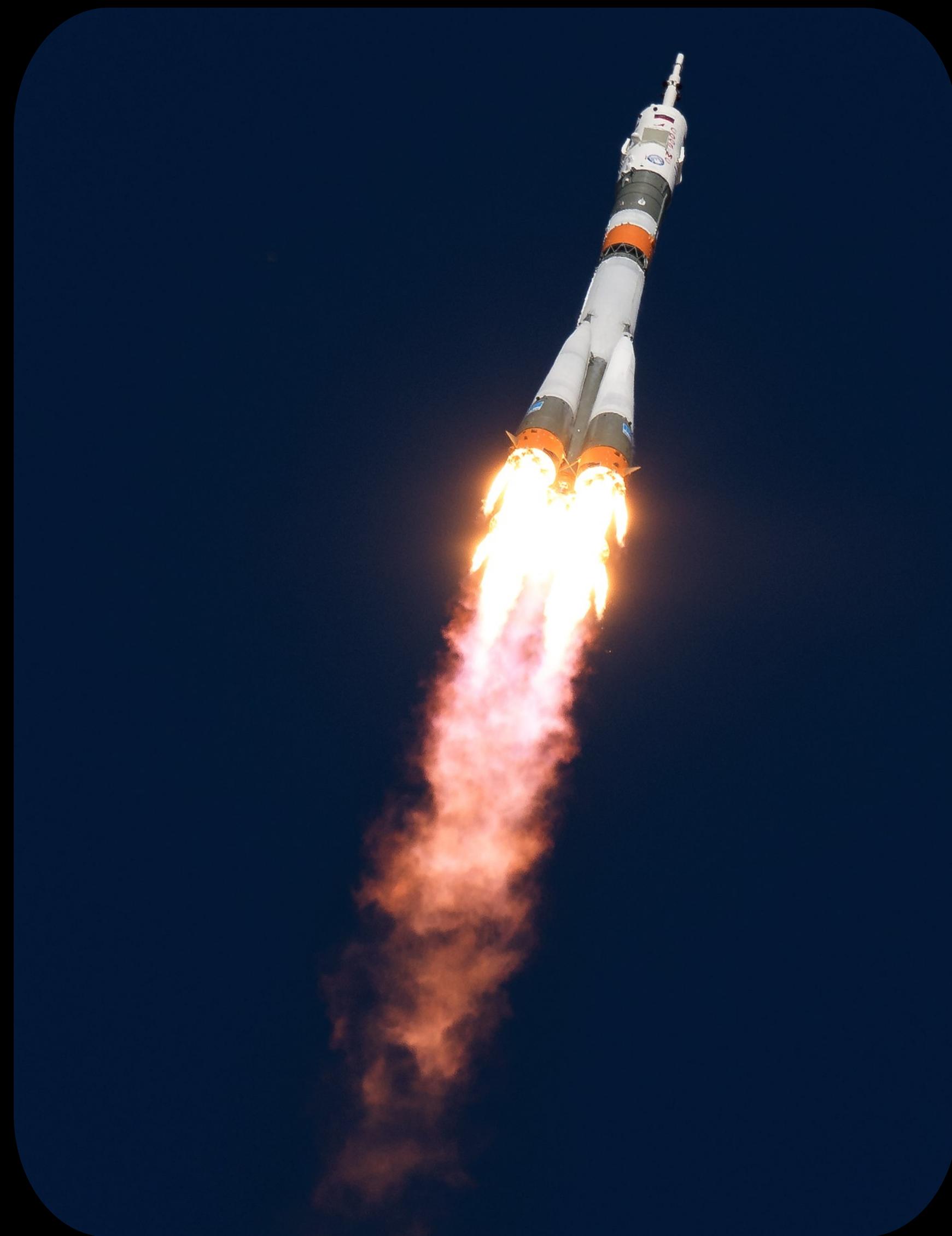
<https://github.com/wojciech12>

```
123 def distance_matrix(regions):  
124     """ Computes a distance matrix against a region list """  
125     tuples = [r.as_tuple() for r in regions]  
126     return cdist(tuples, tuples, region_distance)  
127  
128  
129 def clusterize(words, **kwargs):  
130     # TODO: write a cool docstring here  
131     db = DBSCAN(metric="precomputed", **kwargs)  
132     X = distance_matrix([Region.from_word(w) for w in words])  
133     labels = [int(l) for l in db.fit_predict(X)]
```



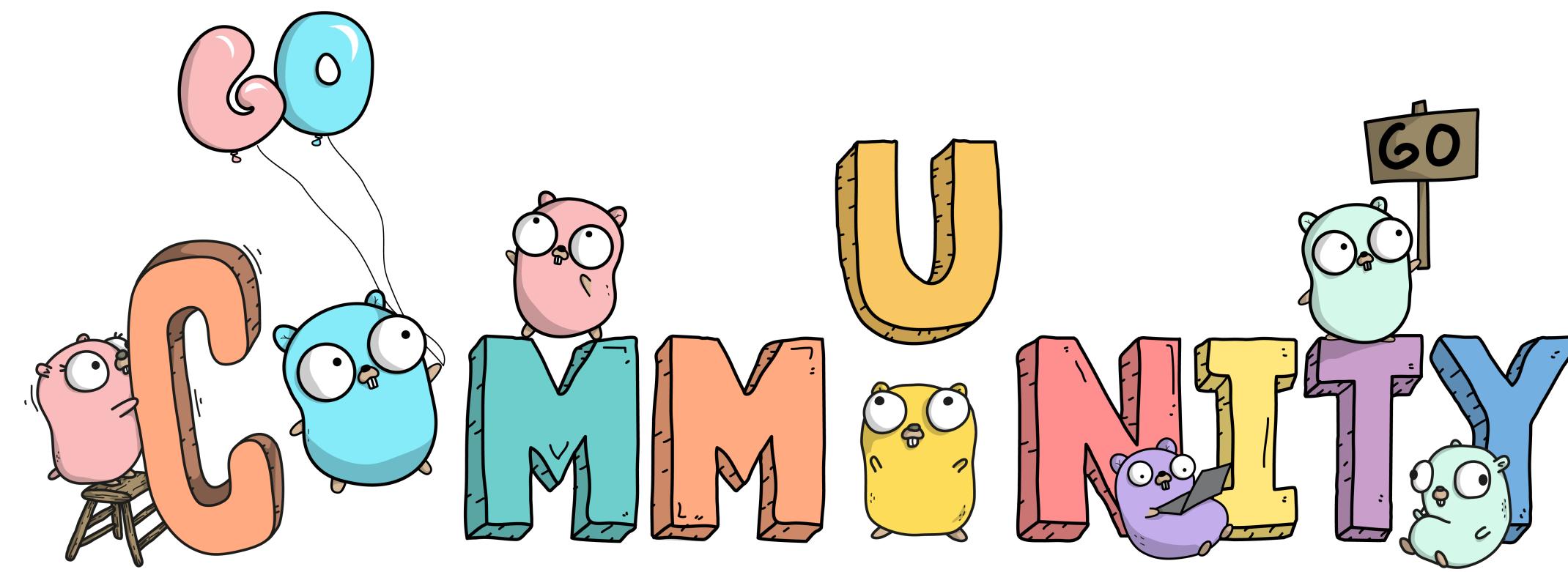
DOING GOOD

- Jetbrains:
most promising
- IEEE:
5th fastest growing
- Kubernetes
- CloudNative
- Next enterprises?



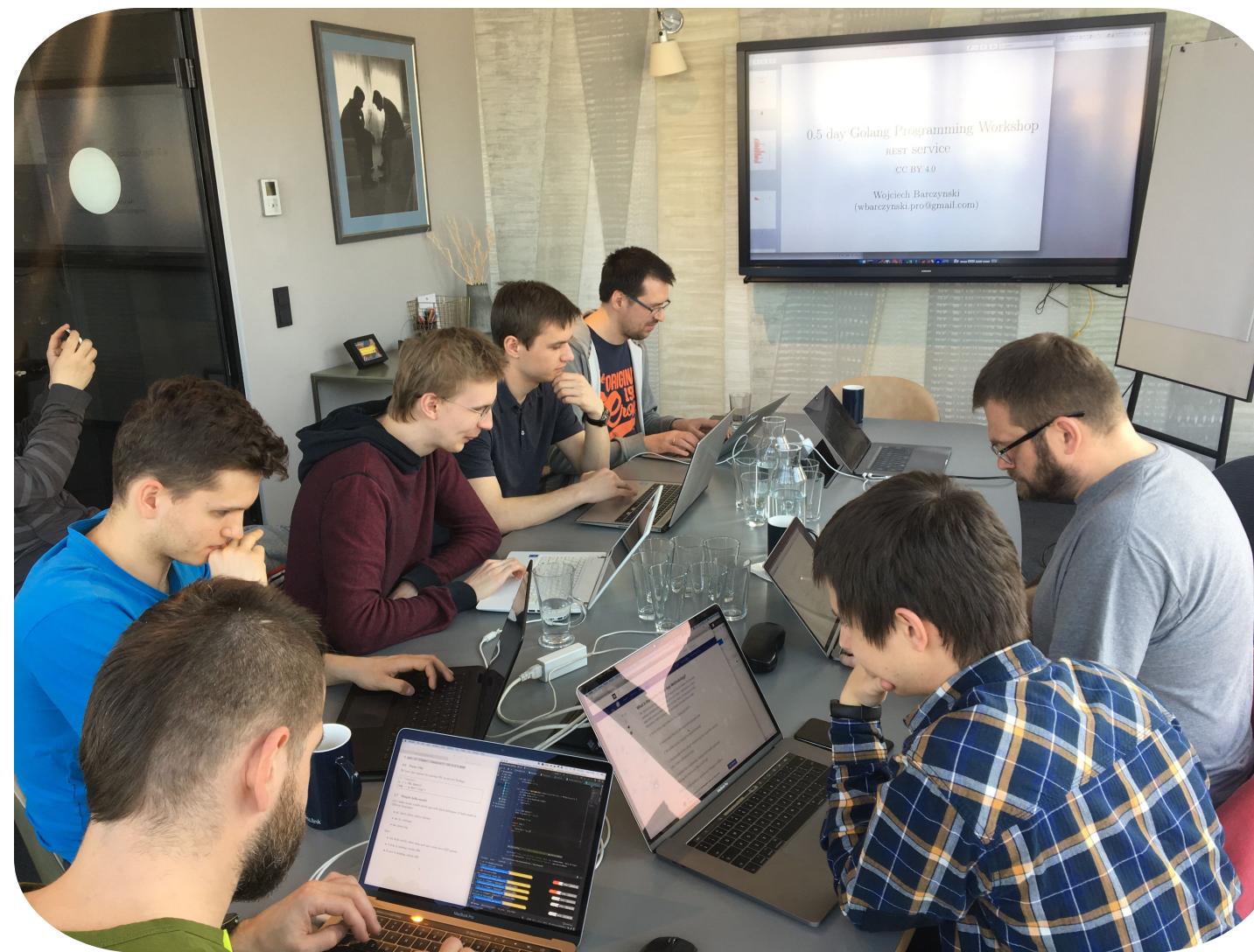
COMMUNITY

- Gopher Warsaw
- GoCracow
- Gopher Silesia
- Golang UG
Gdańsk



IDEAS

- Talks
- Workshops
- Speed get-know each other
- ...



BACKUP SLIDES

```
123 def distance_matrix(regions):
124     """ Computes a distance matrix against a region list """
125     tuples = [r.as_tuple() for r in regions]
126     return cdist(tuples, tuples, region_distance)
127
128
129 def clusterize(words, **kwargs):
130     # TODO: write a cool docstring here
131     db = DBSCAN(metric="precomputed", **kwargs)
132     X = distance_matrix([Region.from_word(w) for w in words])
133     labels = [int(l) for l in db.fit_predict(X)]
```

