

Disrupting development routines with one day sprints

OFFER ZEN

Hi, I'm Elena! I like
to travel and I like
to eat.



Our team is split into squads that focus on different parts of the platform.

Team structure

Product Team

Squad

Squad

Squad

Squad

Team structure

Product Team

Company Product Squad



Product
Manager



Product
Designer



Lead Dev



Dev



Dev

Squad

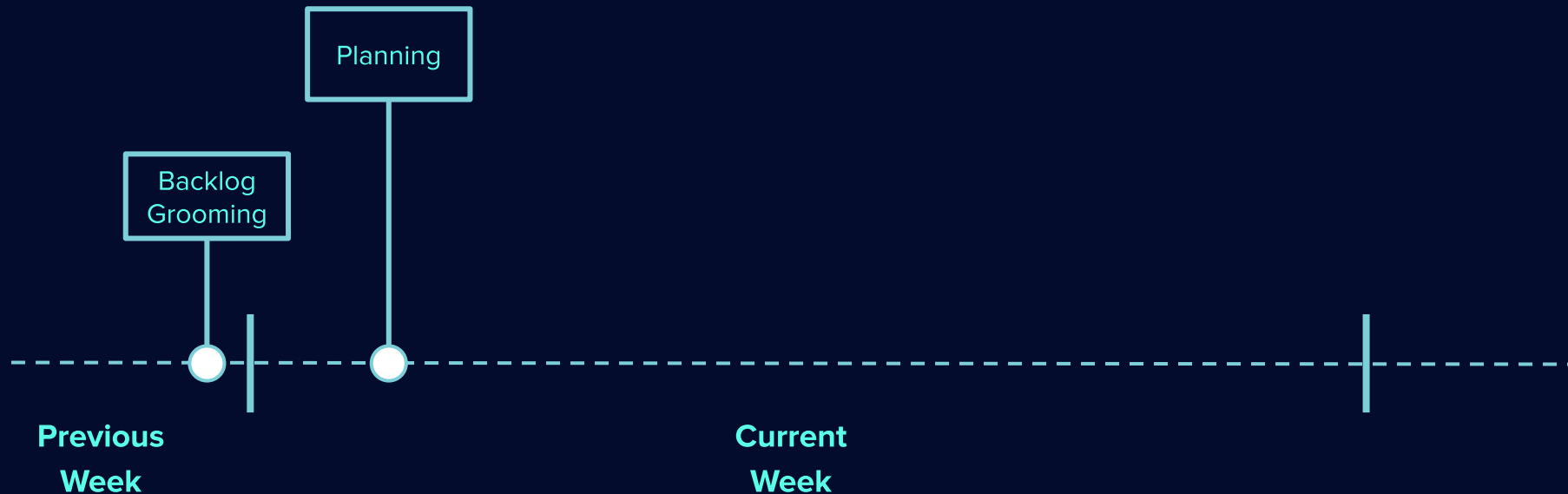
Squad

Squad

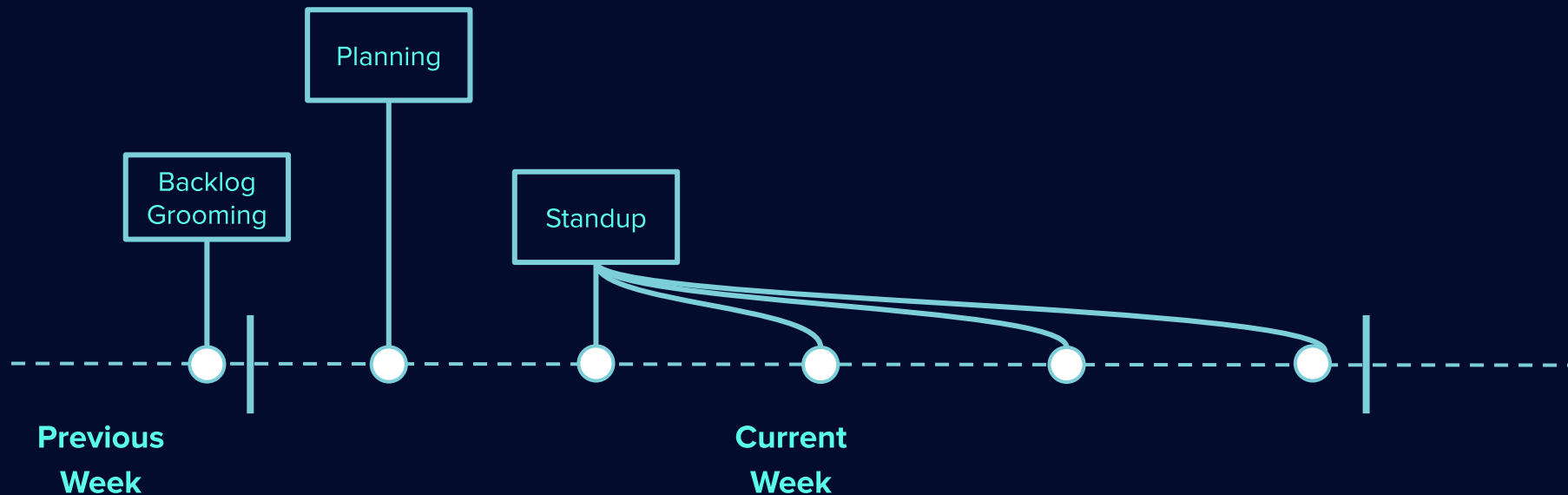
“Delivery Ceremonies”/Routines



“Delivery Ceremonies”/Routines

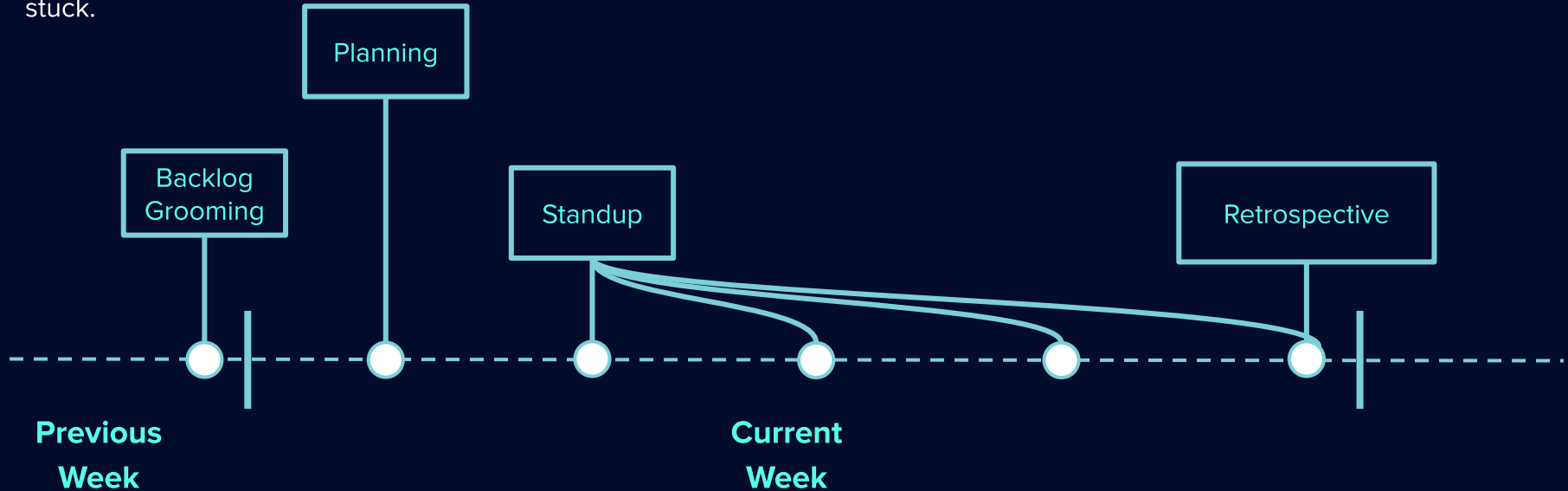


“Delivery Ceremonies”/Routines



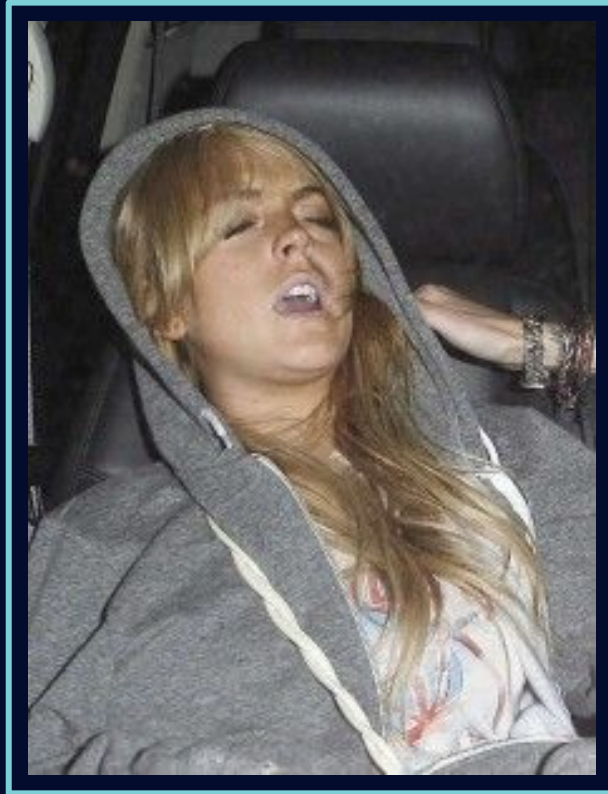
“Delivery Ceremonies”/Routines

This is the flow we were trying to achieve on a weekly basis but we kept getting stuck.



So, we started to
feel like this –
Lindsay Lohan,
circa 2010.

We were
consistently stuck
with what we
called a ‘hangover’
into the next week.





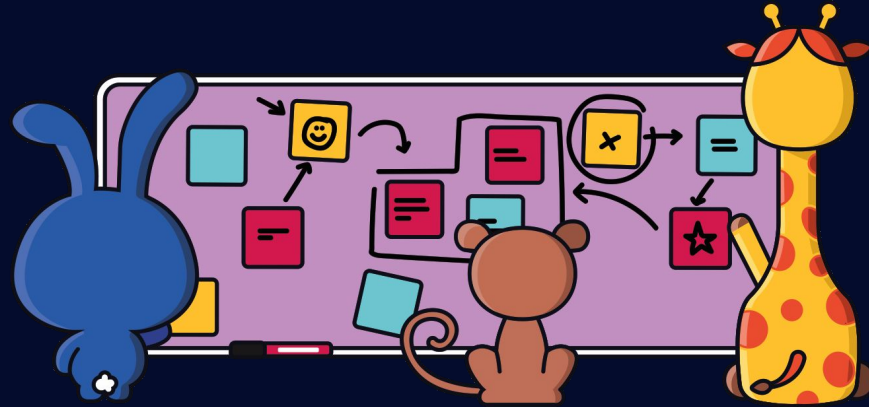
Our problems

- A. Underestimating complexity
- B. Inaccurate time estimation for tasks
- C. Figuring out the dynamics of a new team

Our solution was to
trial one day
sprints for two
weeks.

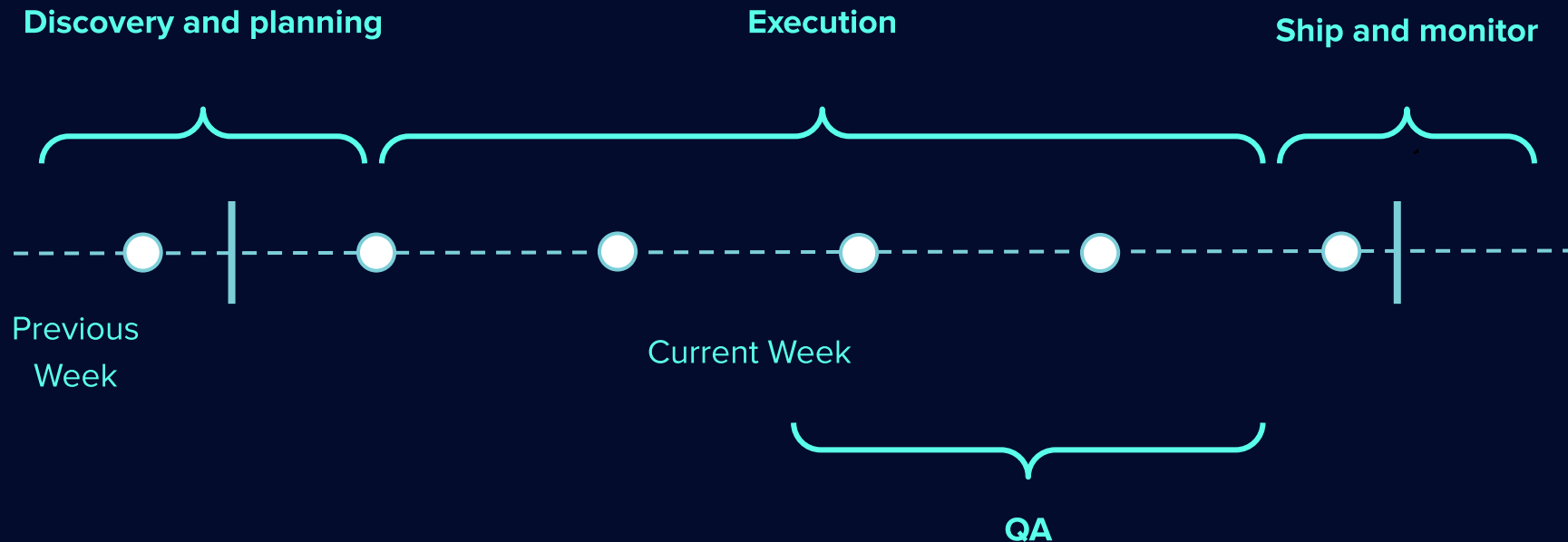
We came out
slightly battered
but mostly
victorious.





What is a one day sprint?

Week sprint



One day sprint timeline



One day sprint setup

The two things that proved to be most important in terms of physical set up were:

- Getting a room to ourselves to work uninterrupted.
- Having as many screens as possible.





Our problems and how one day sprints helped



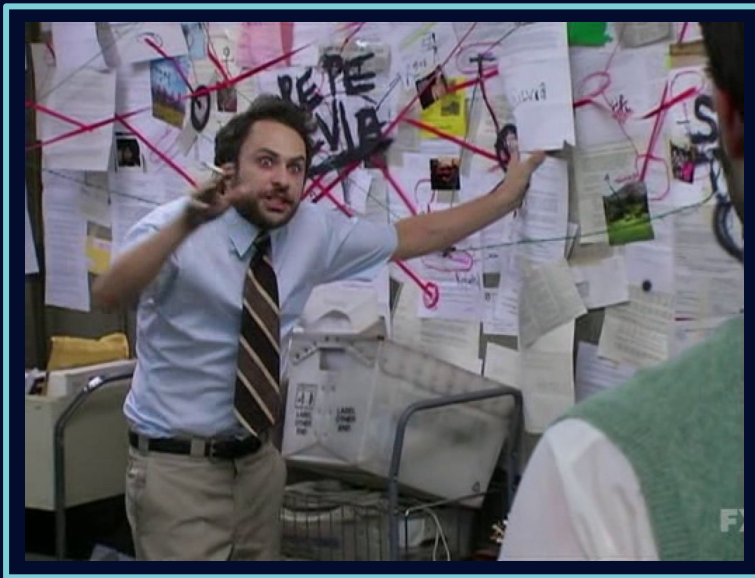
Underestimating complexity

- A. Hidden complexity in the system
- B. Swoop and poop

Underestimating complexity

To try and counteract surprise complexity, our tactic was to figure out how to do the task without writing any lines of code.

This approach forced us to think of a multitude of solutions and then choose the best one.



Hidden complexity in the system

Underestimating complexity



Hidden complexity in the system

Underestimating complexity

‘Swoop and poop’ is how we affectionately refer to stakeholders coming in and changing the scope of the project or revealing some vital information halfway through the task.

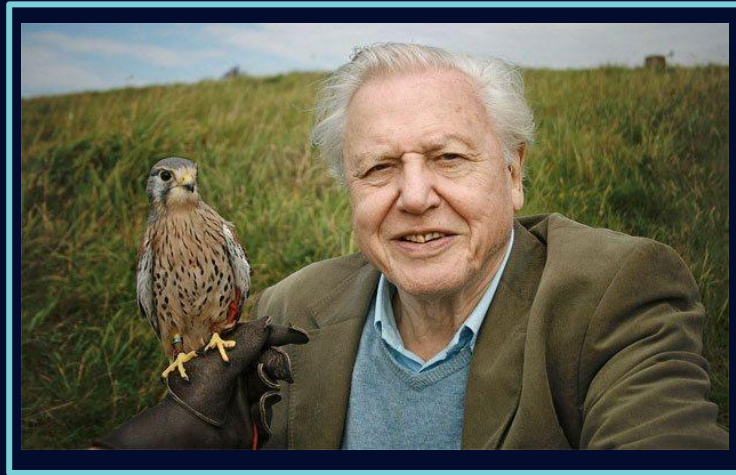
During the one day sprints, we decided not to tolerate any vital info coming late, so we included stakeholders in the non-technical discovery chunk of the day.



Swoop and poop

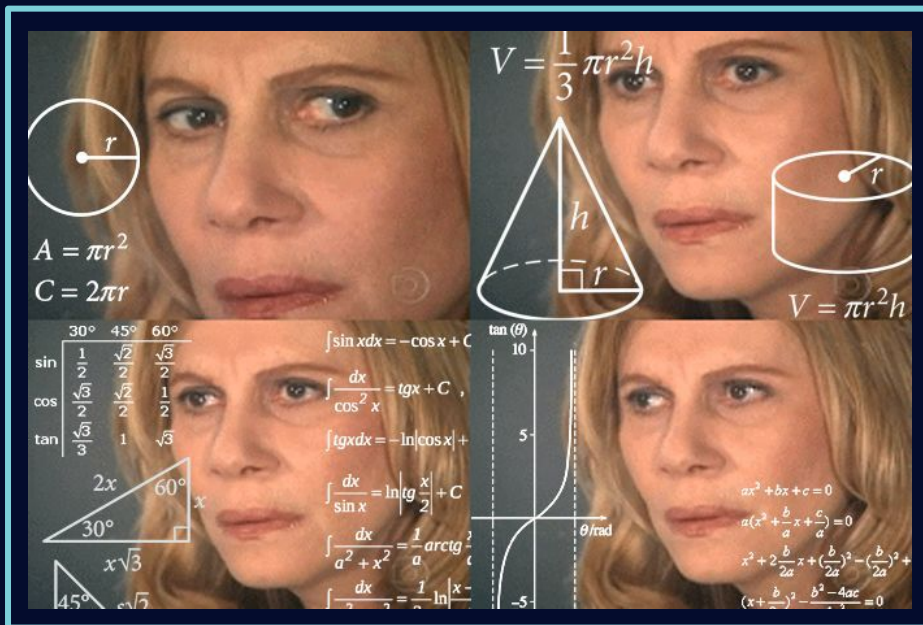
Underestimating complexity

This helped us
tame the
swoopers!



Swoop and poop

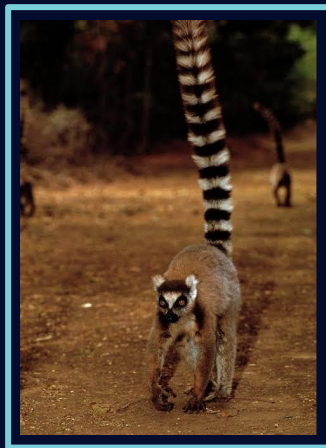
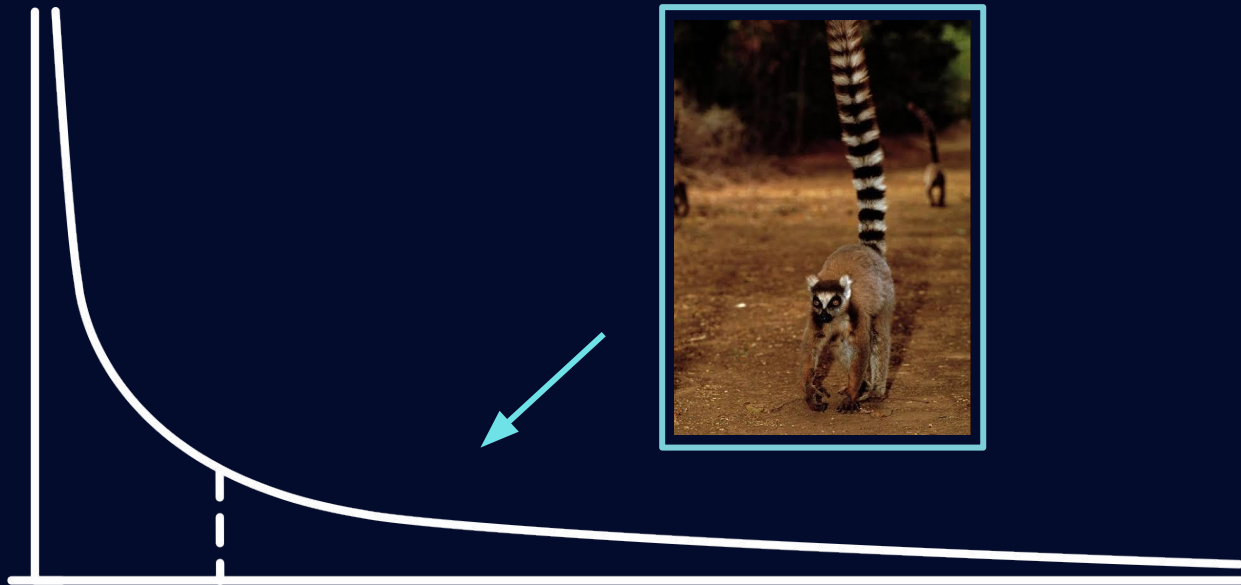
Time estimations are impossible



Time estimations are impossible

The long tail of small tasks at the end of a mission was delaying us before, as we didn't account for them in our planning.

During the one day sprints, we planned a one-day unit of work, factoring these fiddly tasks in.



The long tail of small fiddly tasks

Figuring out **dynamics** of a new team

Most of us who were involved in this experiment were pretty new at OfferZen, so we hadn't quite figured out how best to work with each other yet.



Figuring out **dynamics** of a new team

The one day sprints really helped with this because all our brains were on one problem simultaneously, so we quickly got an understanding of how each person solved problems and where they were knowledgeable.



Figuring out **dynamics** of a new team

Here's a photo of us during the one day sprint experiment.



Figuring out **dynamics** of a new team

We also got to know each other better.

For example, this is Madelein. She is a full-stack developer and plays seven instruments!



Madelein

Figuring out **dynamics** of a new team

And this is Andries,
whose two Maltese
poodles are slowly
sucking away his
life force.

Andries



Madelein

By the end of our one day sprints, we were exhausted.

They required a huge amount of brain power. Two weeks was pretty long to commit all that energy every single day.



But once the exhaustion wore off, we were pretty happy that we'd tried something new that helped us hit the 'reset' button as a team.



We've now gone back to doing week-long sprints but we've incorporated a lot of what we learned during the one day sprints into how we plan and execute on tasks.

Most NB was that one day sprints helped surface invisible issues in our routines.



The aftermath



We learnt about

- A. The complexity of our codebase
- B. Extracting stakeholder opinions and information early
- C. Estimating how long tasks take
- D. Our teammates



The End

OFFER ZEN