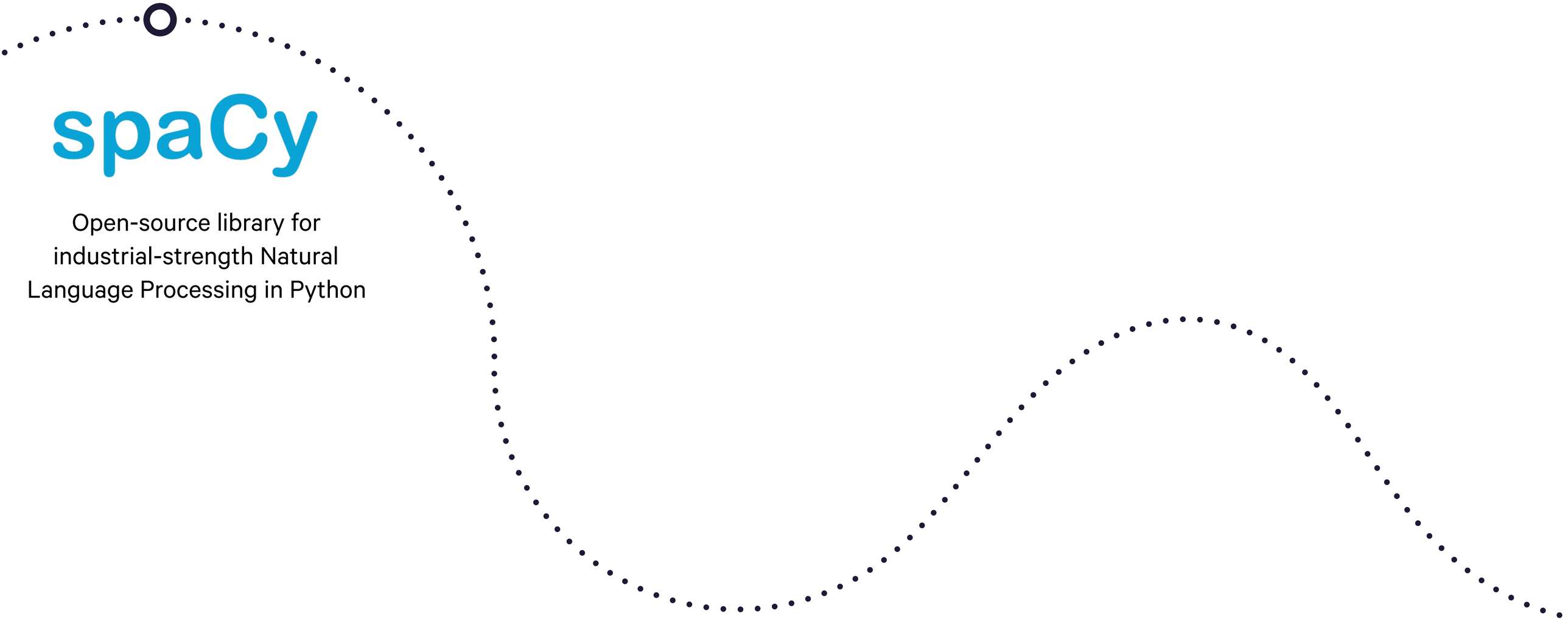
# How to Ignore Most Startup Advice and Build a Decent Software Business

Ines Montani

**Explosion AI**




Open-source library for industrial-strength Natural Language Processing in Python

# spaCy

Open-source library for industrial-strength Natural Language Processing in Python

Company and digital studio, bootstrapped with consulting

# spaCy

Open-source library for industrial-strength Natural Language Processing in Python

Company and digital studio, bootstrapped with consulting

# prodigy

First commercial product: radically efficient data collection and annotation tool, powered by active learning

spaCy

Open-source library for industrial-strength Natural Language Processing in Python

EXPLOSION

Company and digital studio, bootstrapped with consulting

prodigy

First commercial product: radically efficient data collection and annotation tool, powered by active learning

**You are here!**

spaCy

Open-source library for industrial-strength Natural Language Processing in Python

EXPLOSION

Company and digital studio, bootstrapped with consulting

prodigy

First commercial product: radically efficient data collection and annotation tool, powered by active learning

**You are here!**

prodigy

**ANNOTATION MANAGER**

Extension platform with a SaaS layer to help users scale up annotation projects

spaCy

Open-source library for industrial-strength Natural Language Processing in Python

EXPLOSION

Company and digital studio, bootstrapped with consulting

DataStore

*Coming soon:* pre-trained, customisable models for a variety of languages and domains

prodigy

First commercial product: radically efficient data collection and annotation tool, powered by active learning

**You are here!**

prodigy

**ANNOTATION MANAGER**

Extension platform with a SaaS layer to help users scale up annotation projects

# The “startup playbook” isn’t the only way.

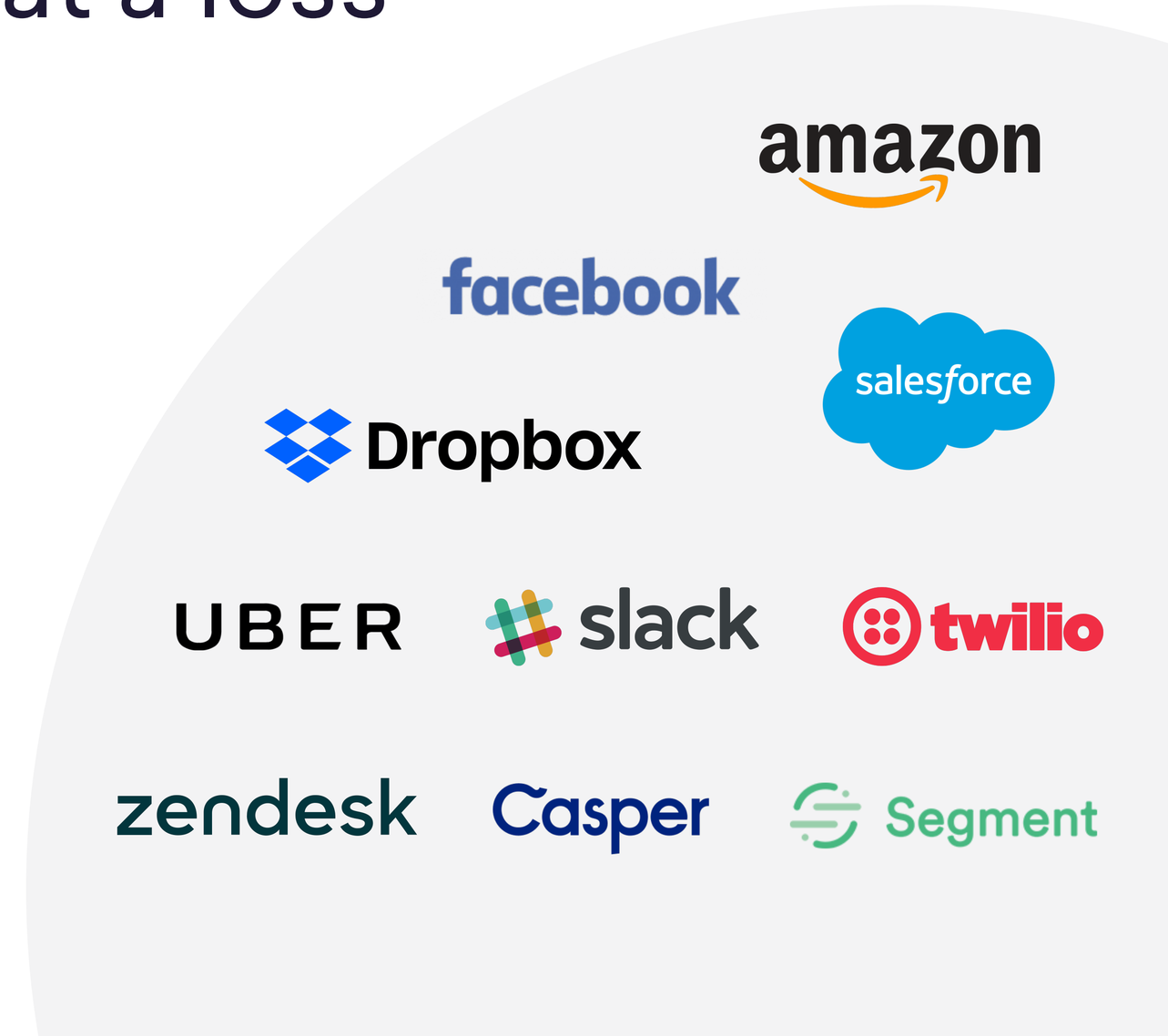
- it’s possible to be profitable early
- it’s possible to keep the team small
- you don’t have to do anything sneaky, you can just make something good

MISCONCEPTION #1

# You need to run at a loss.

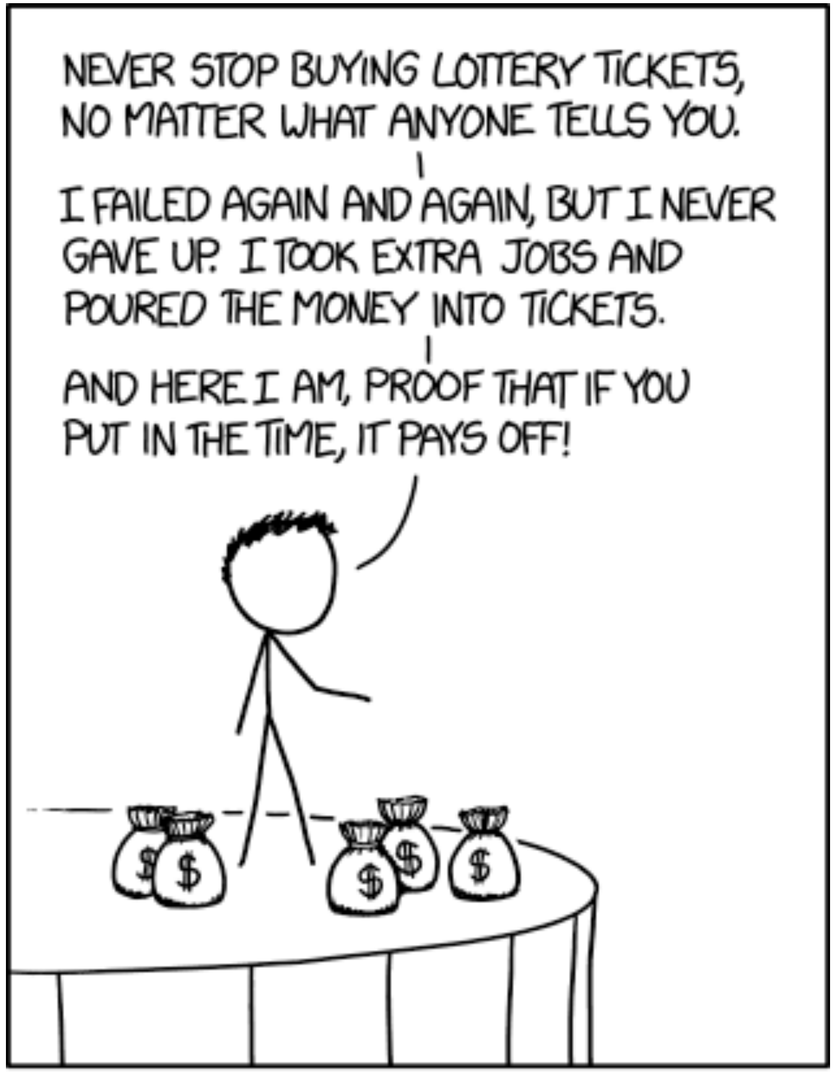
# Reasons to run at a loss

- network effects
- scale operations
- predatory pricing
- enterprise sales



# Bigger isn’t necessarily better.

- software is **more expensive** to build at scale, not less
- most businesses **aren’t** “winner takes all”
- being in a “winner takes all” market kinda sucks anyway




EVERY INSPIRATIONAL SPEECH BY SOMEONE SUCCESSFUL SHOULD HAVE TO START WITH A DISCLAIMER ABOUT SURVIVORSHIP BIAS.

Source: xkcd.com/1827

# The good news is:
# so many opportunities!

- people are drawn to “tournaments” and “winner takes all” markets
- this leaves many other **high-value** opportunities untouched
- optimize for **median** (not mean!) outcome

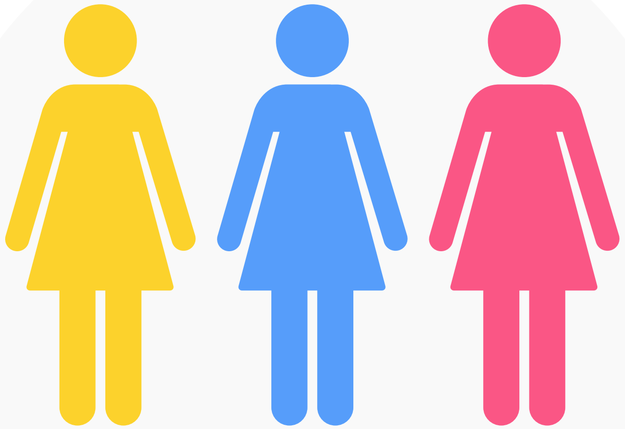
EXPLOSION

**MISCONCEPTION #2**

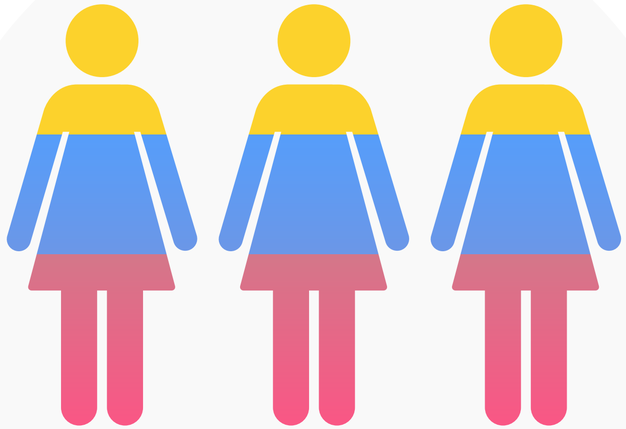
# You need to hire lots of people.

# Good teams can be surprisingly small

- you don’t need to pass the “bus test”
- excellence requires **authorship**, not redundancy or design by committee
- building the **right stuff** matters much more than building lots of stuff

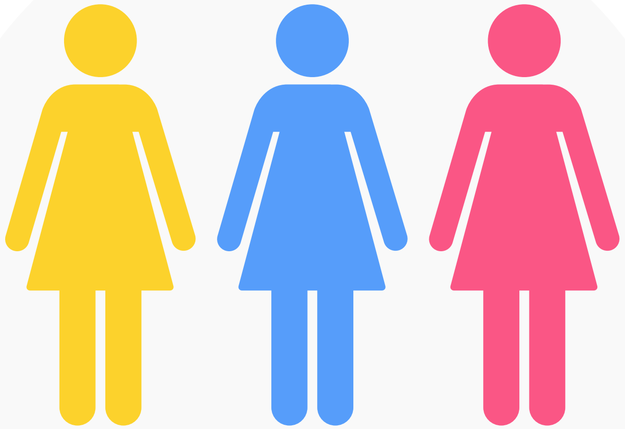


specialists

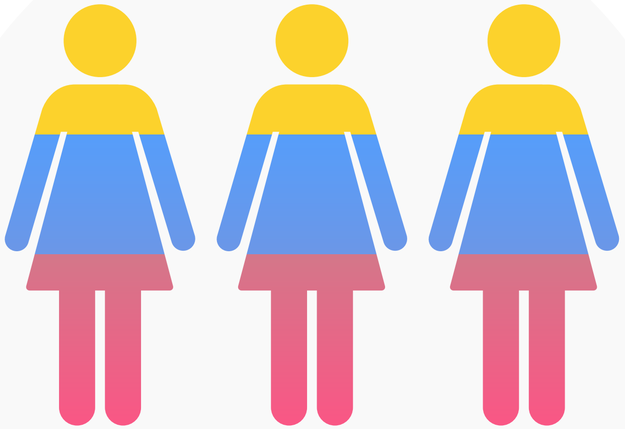


generalists

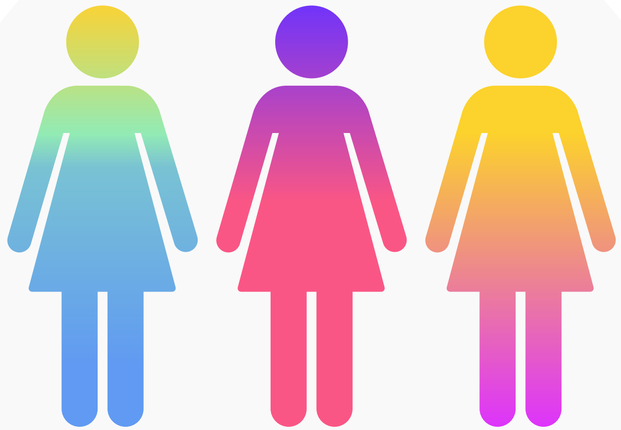
EXPLOSION



specialists



generalists



complementary

T-shaped
skills

tree-shaped
skills

**MISCONCEPTION #3**

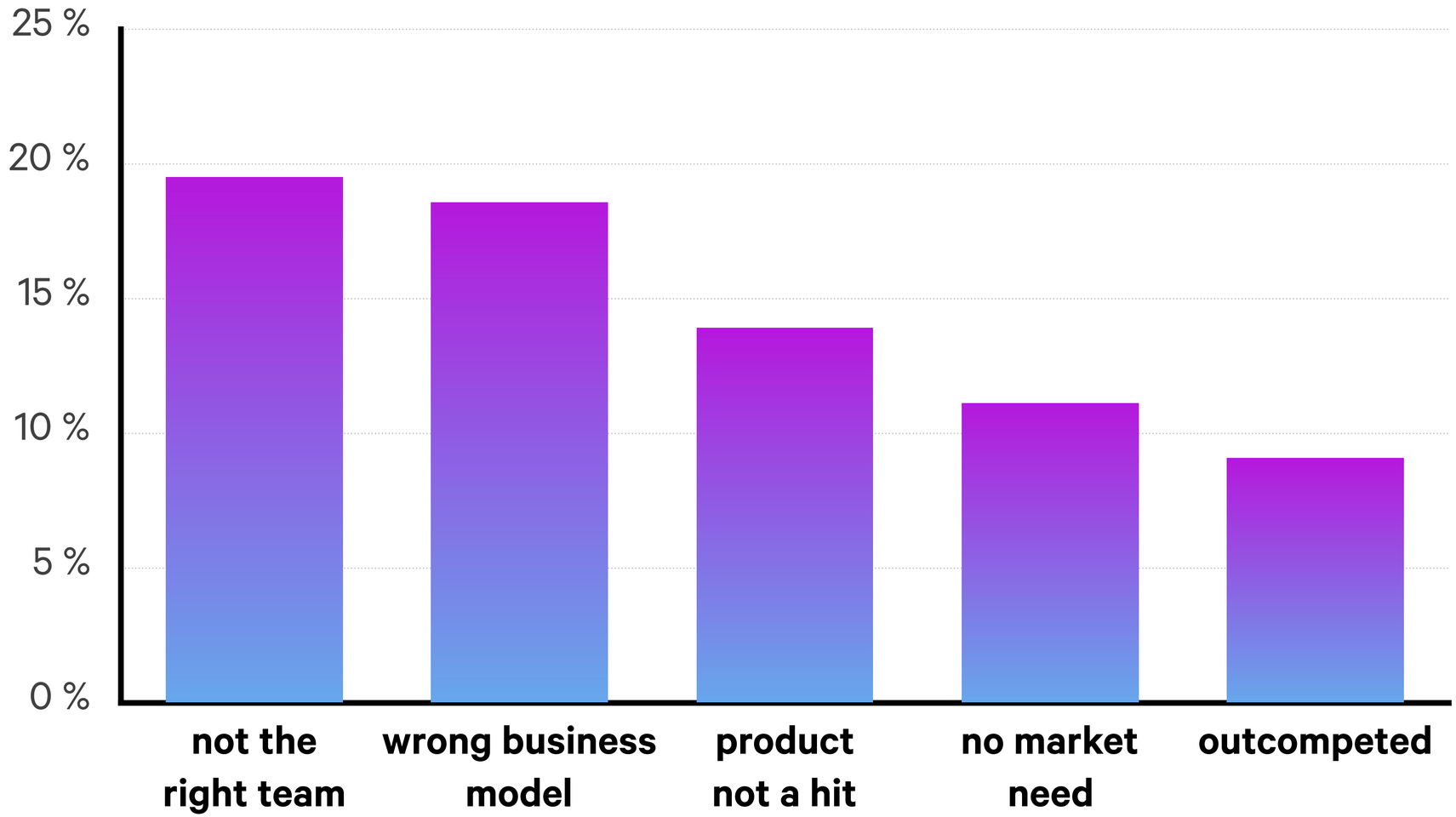
# You can’t make good decisions without testing all of your assumptions.

**“It turned out nobody wanted our product… I wish we’d spent more time validating our ideas! Next time I’m running a 100% data-driven startup!”**

EXPLOSION

- **inverse of survivorship bias:**
  “We didn’t do X and we failed, therefore X would have saved us.”

# Top 5 reasons startups fail based on 300 “autopsies”




Source: autopsy.io

Source: hyperboleandahalf.blogspot.com

Our company Twitter makes us look clueless and insecure. We need to stop retweeting random crap.

Do you have numbers to back that up?

What? No.

Then how do I know you’re right?

By thinking?

# You can’t replace logic with data.

- decisive data is the exception, not the rule
- decisions are mostly based on **reason**
- you’ll **win** if you’re **mostly right**
- build things **you** think are good

EXPLOSION

**MISCONCEPTION #4**

# The true value lies in your users' data.

```
$ prodigy ner.teach product_ner
  en_core_web_sm /data.jsonl
  --label PRODUCT

$ prodigy db-out product_ner >
  annotations.jsonl
```

Prodigy Annotation Tool: prodi.gy

# Sell products, not promises.

- fundraising logic: potential > reality
- focus on what you can *really* charge people money for *right now*
- other objectives not worth adding friction and making your product worse

# Monetize the money

- ship **value**, charge money
- users appreciate **software that works**
- users are not interchangeable test subjects, they’re people and they remember things
- **profit** is the best KPI

# Thanks!

💥 **Explosion AI**
explosion.ai

📲 **Follow us on Twitter**
@_inesmontani
@explosion_ai