

Site Reliability Engineering (SRE): The Big Picture

Introducing Site Reliability Engineering



Elton Stoneman
Freelance Consultant and Trainer
[@EltonStoneman](https://twitter.com/EltonStoneman) | blog.sixeyed.com

What is SRE?

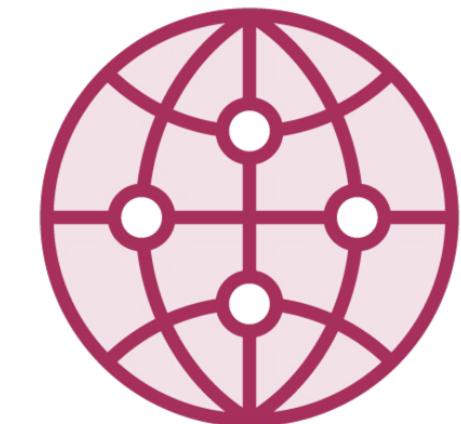
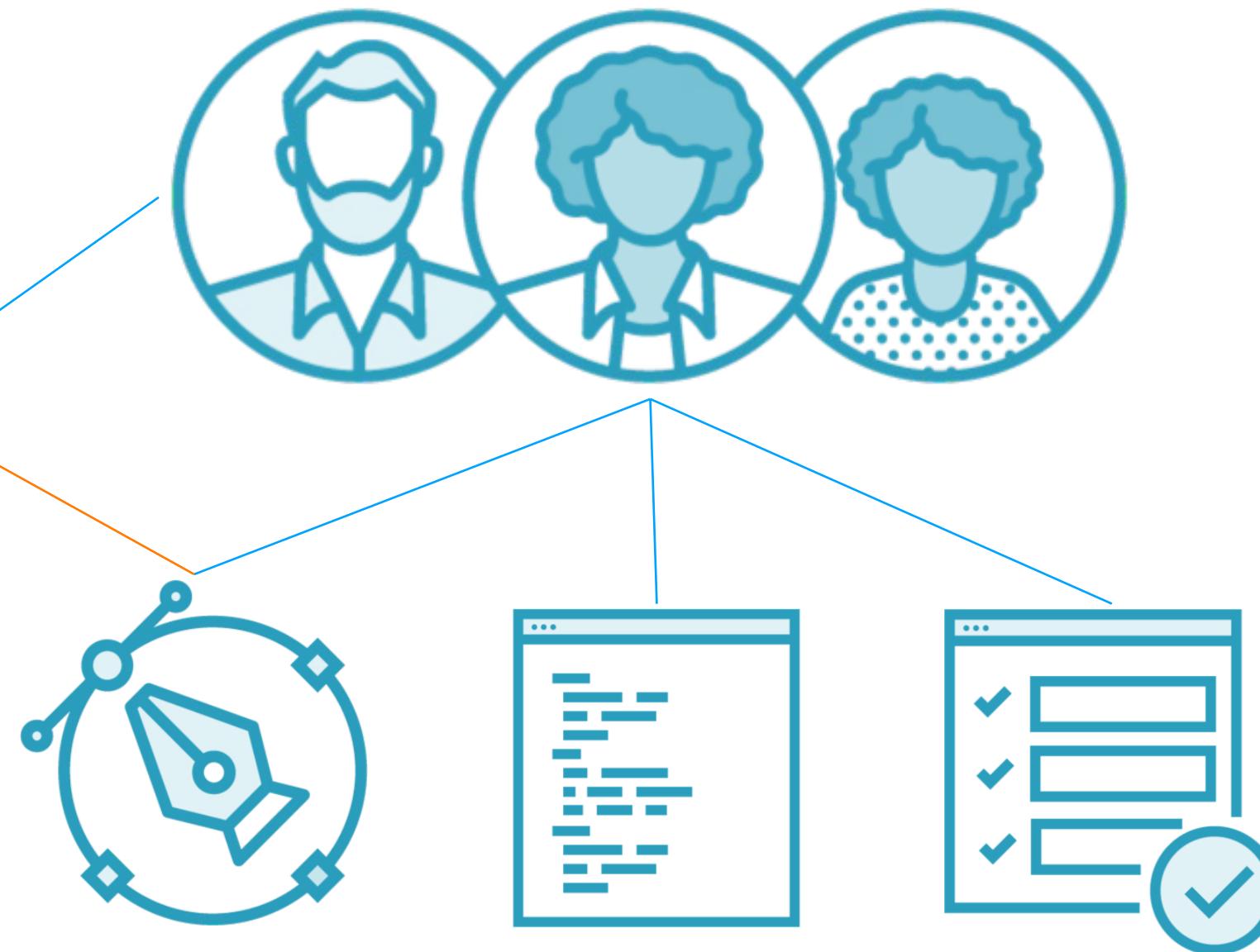
An approach to operations which uses software as the primary tool for managing systems.



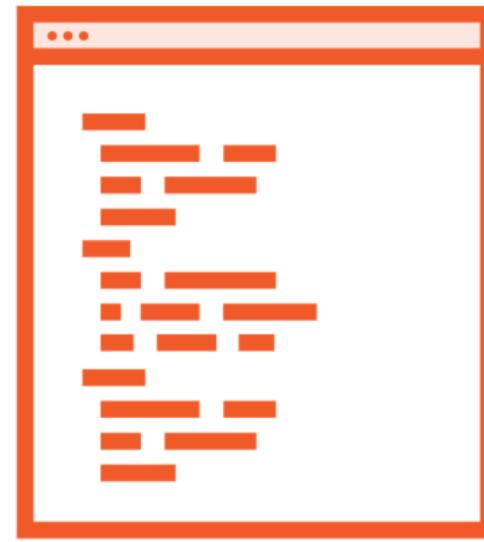
SRE Team



Product Development Team



Functions of Site Reliability Engineering



Eliminating Toil

Automating manual,
repetitive work

Managing Risk

Agreed service levels
with explicit tolerance

Handling Failure

Incident management
and post-mortems





How Google runs production systems

- Engineering approach to ops, 2003
- Site Reliability Engineering, 2016
- The Site Reliability Workbook, 2018

Industry-recognized role

- @SREcon
- Google, Netflix, Amazon

Viable transition from "ops"

- Solves dev vs. ops problems
- Without the upheaval of DevOps



Comparing Traditional Ops and SRE



Ops Team



Development Team



No, thanks

Here's a new thing

Give me reliability

Give me new stuff



"The Business"

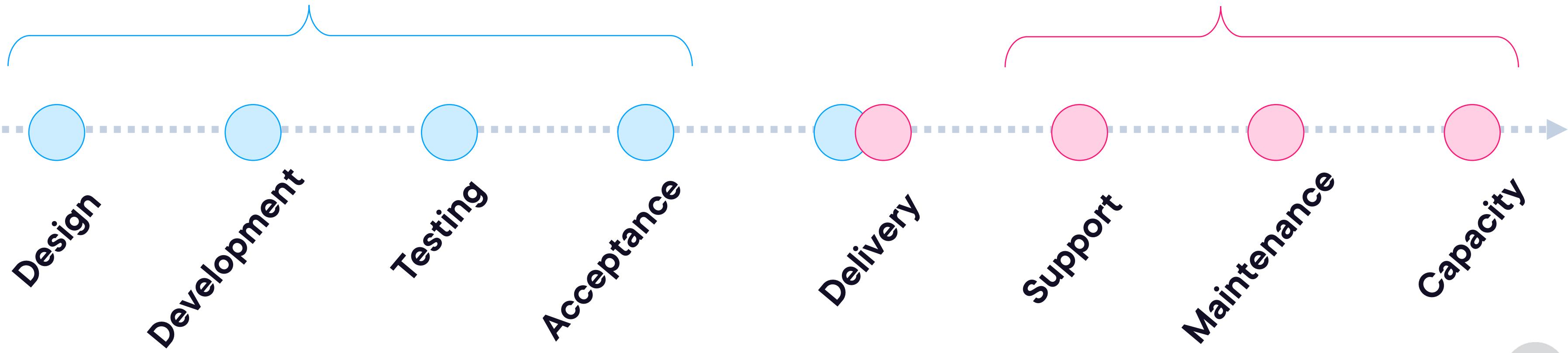


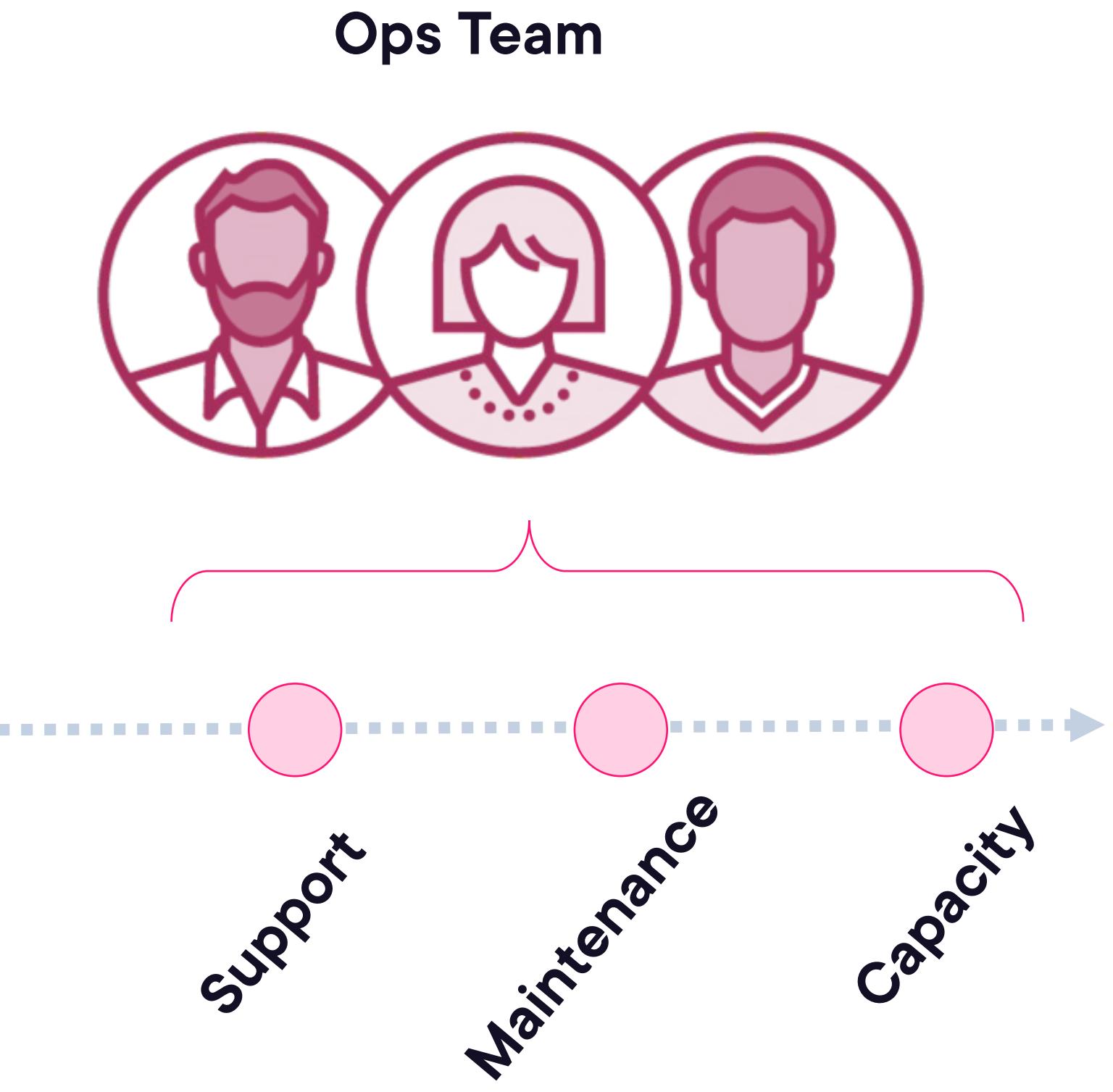
- Different goals
- Different skillsets
- Different tools
- No common ground

Development Team



Ops Team





Black box delivery

- No design input
- Automation options limited
- Infrastructure-level monitoring

Limited agency

- Post-acceptance
- Go or no-go decision
- Conflict guaranteed

After-the-event

- Mandated service levels
- Reactive support

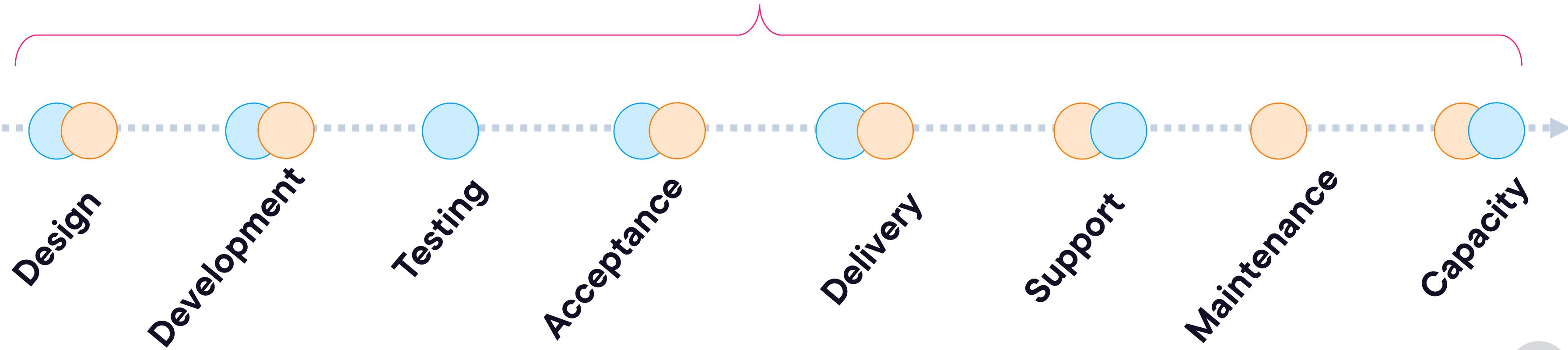


- Shared goal
- Overlapping skillsets
- Consistent tools
- Common basis

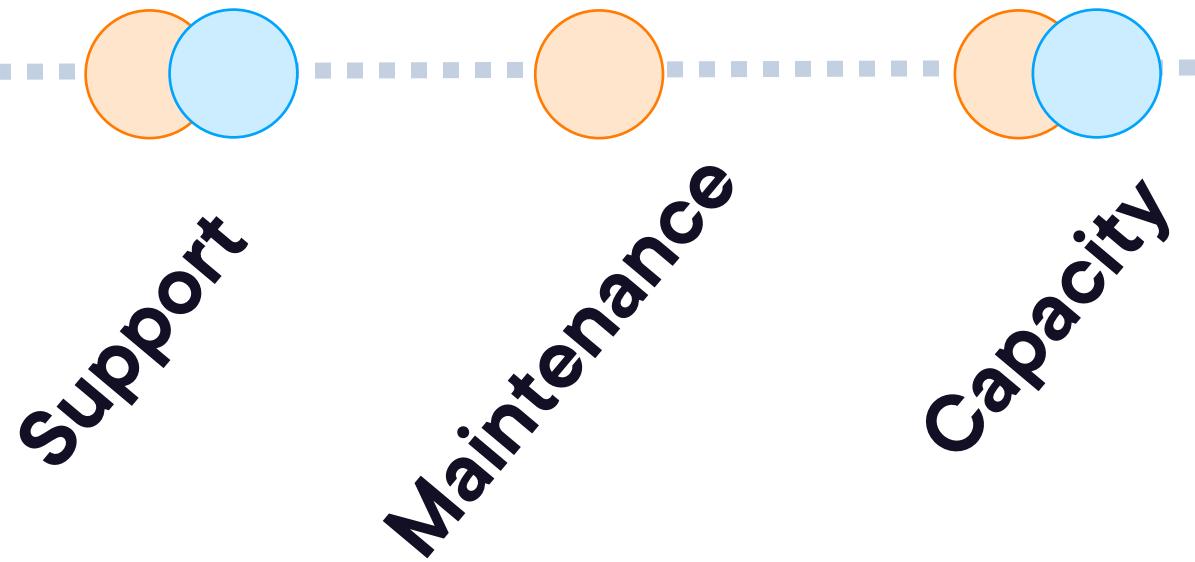
Development Team



SRE Team



SRE Team



Agreed delivery

- Ongoing design input
- Extended automation options
- User-focused monitoring

Full agency

- Dev team support
- Stop deployments
- Hand back the pager

After-the-event

- Agreed service levels
- Limited support time



Comparing DevOps and SRE



DevOps Team



***Give me reliability
and new stuff***

OK



"The Business"



Transitioning-to-DevOps Team



***Give me reliability
and new stuff***



"The Business"

Sorry...

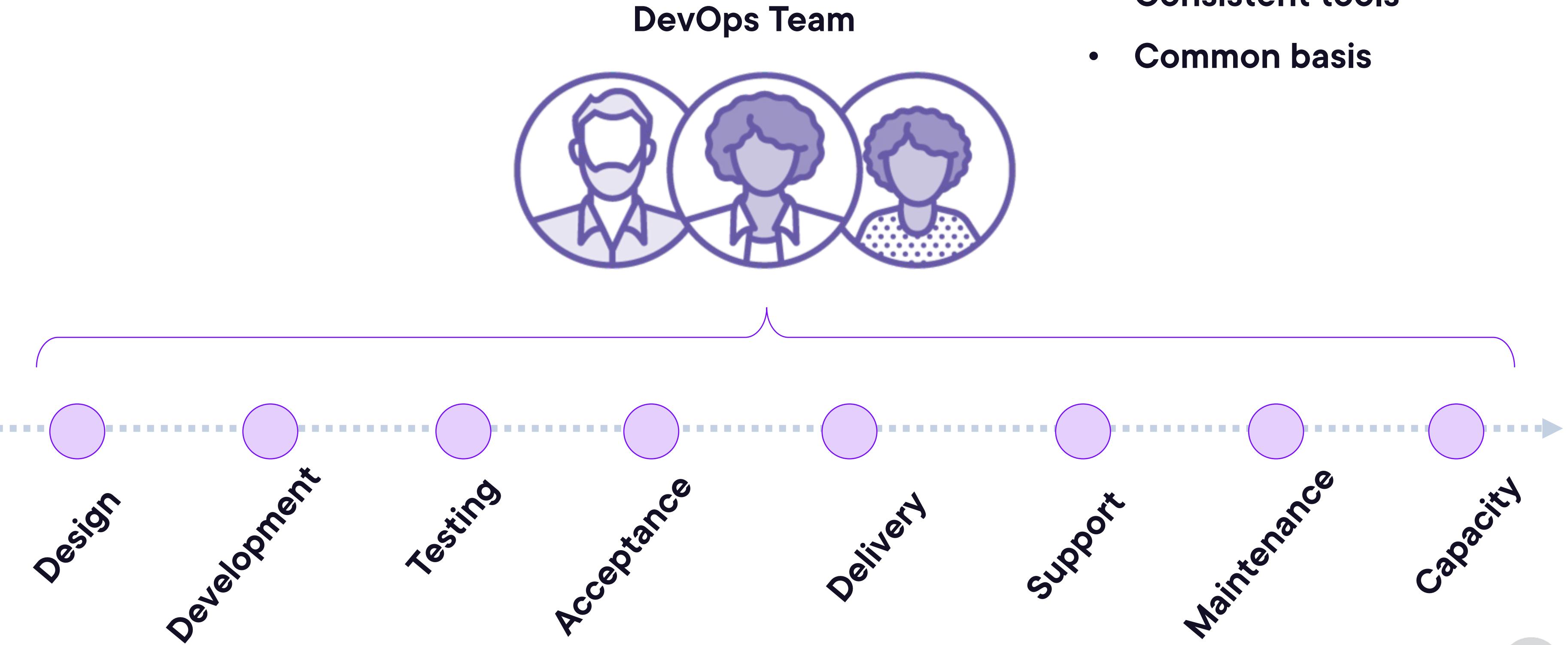
***no new features
until we've
automated the
release process***

***we think the app's
down so we're
adding more
monitoring***

***we're all on DevOps
training this week***



- **Shared goal**
- **Overlapping skillsets**
- **Consistent tools**
- **Common basis**

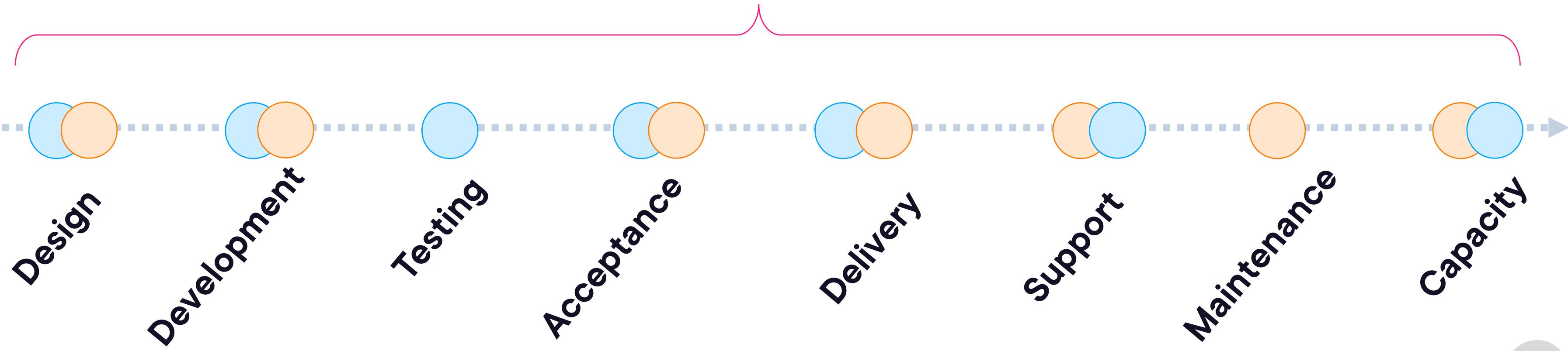


- Shared goal
- Overlapping skillsets
- Consistent tools
- Common basis

Development Team



SRE Team



SRE Team



DevOps Team



Much in common

- Focus on quality & velocity
- Automation & tooling
- Removing silos

But DevOps...

- Has a broader remit
- Significant cultural change

And SRE...

- Is more prescriptive
- Feasible internal migration



Development Team



Ops Team



DevOps Team



- **Significant investment**
- **Acceptance of change**
- **Needs outside help**
- **Uneven rollout**



Development Team



Ops Team



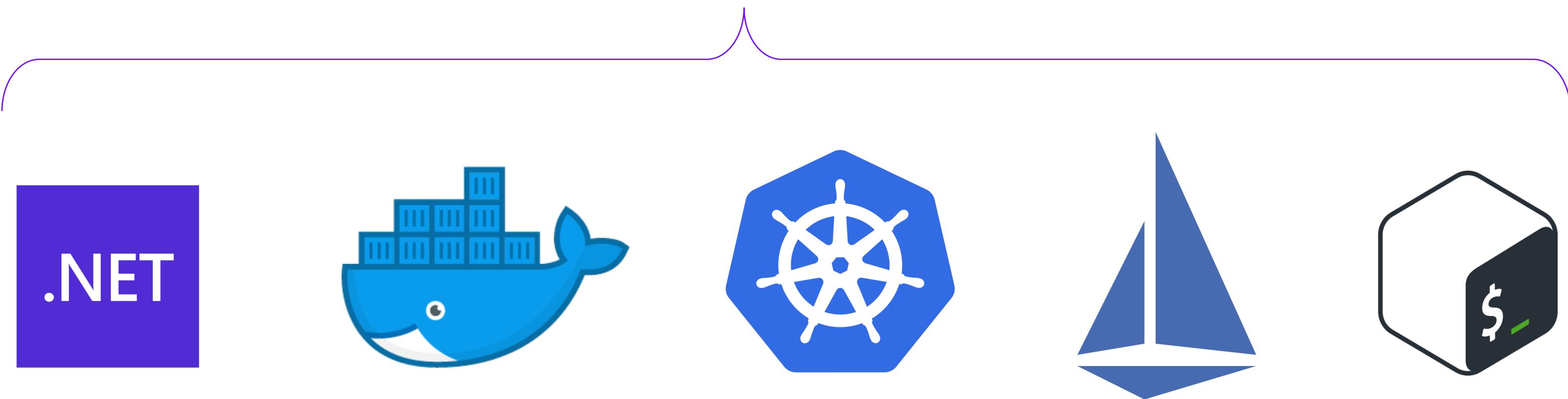
Development Team



SRE Team



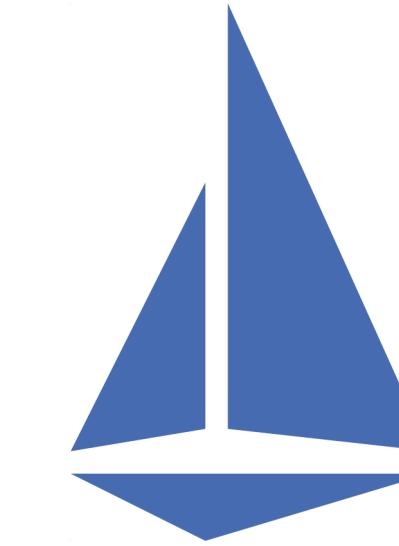
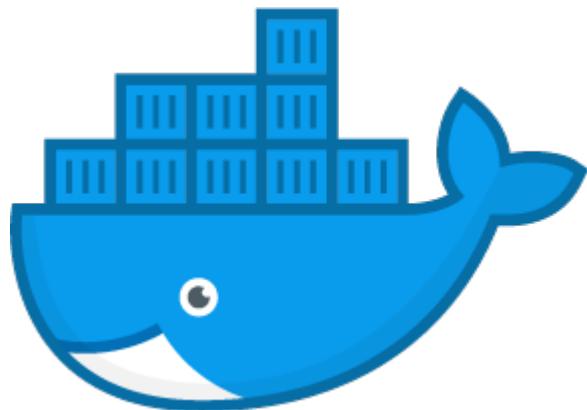
DevOps Team



Development Team



SRE Team



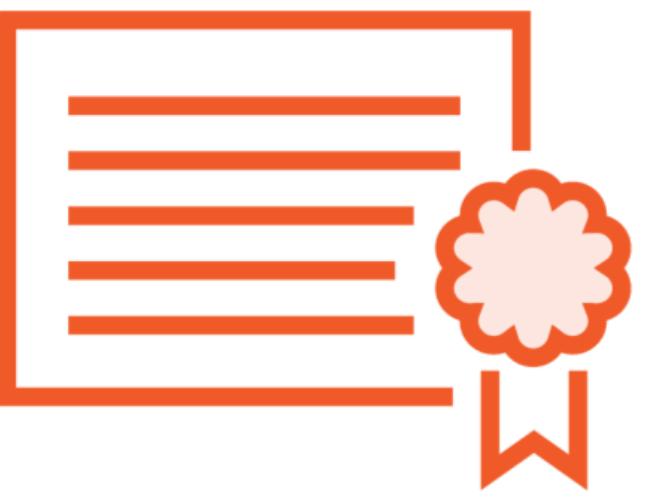


Exploring the Key Tenets of SRE





Eliminating Toil



Working to Service Levels



Managing Failure





**New release;
logs fill disk
daily**

**Log in weekly;
delete old log
files**

**Scale up to 50
servers**

**Script to
delete old log
files; log in and
run daily**

**Work with dev
team to
reduce logs**

**CRON job to
run script daily**





**New release;
bugs cause
latency**

**Service level
maintained**

**New release;
config issue
causes outage**

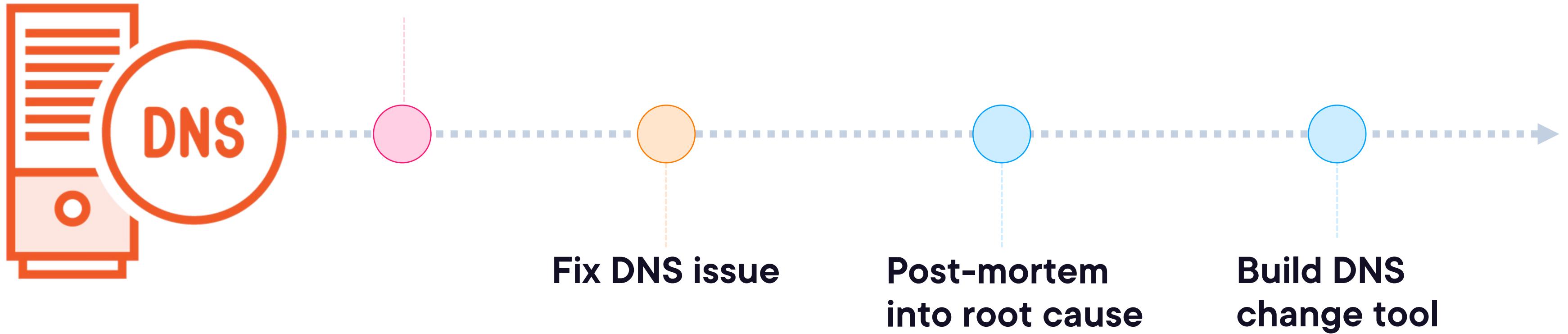
**Fix config and
restore service**

**Block new
releases for
agreed period**

**Increase
capacity;
report issues
to dev team**



**DNS change
causes outage**





Understanding Why SRE Works





vs.



Conflict between dev and ops

- Made explicit
- Contracted service level
- Business buy-in

Integration between dev and ops

- Shared responsibility
- Shared workload
- Common toolkit





Prescriptive practices

- On-call teams and targets
- %age SRE time on toil
- Service level framework

IT-led migration

- Within existing org chart
- Maintain product knowledge
- Start small





SRE empowerment

- Improving systems
- Formal structure
- Tools to meet responsibilities

SRE as an attractive role

- Reduce mundane work
- Expand skills
- Career development





Miles Bryant

@milesbxf

We run an on-call rota [@monzo](#) that is so popular we have a waiting list of people looking to join it. If you'd like to hear more, I'm really looking forward to speaking about it at [@SREcon](#)

Monzo, UK startup bank



**“It's what happens when you ask a
software engineer to design an
operations function.”**

Ben Treynor Sloss, VP Engineering, Founder of Google SRE



Summary



SRE is an approach to IT operations

- Maintains a dev/ops split
- But adds engineering to "ops"

Shares goals with DevOps

- Breaking down silos
- Increasing quality and velocity

Growing adoption

- '00s job roles
- Sympathetic migration



Up Next:

Automation and Eliminating Toil

