On-call is ruining my life

And other tales about holding the pager

- Co-founder and CEO of Hotpot
- hotpot.works
- Been on call for ~30 years
- SRE, o11y, reliability @ Twitter, Stripe, Splunk, etc

A story about me and Kafka.

Most **organizations** are **not** set up to support engineers handling the workload, interruptions, and stress of on-call.

But the teams are doing it anyway…

“As far as leadership is concerned, ***heroics*** *[are] a viable, long term strategy”*

The talk

- The research
- The findings AKA the “no good” nine
- Recommendations
- Next steps

Research

Literature review

30+

academic papers

40+

industry materials

Survey

65

questions

53

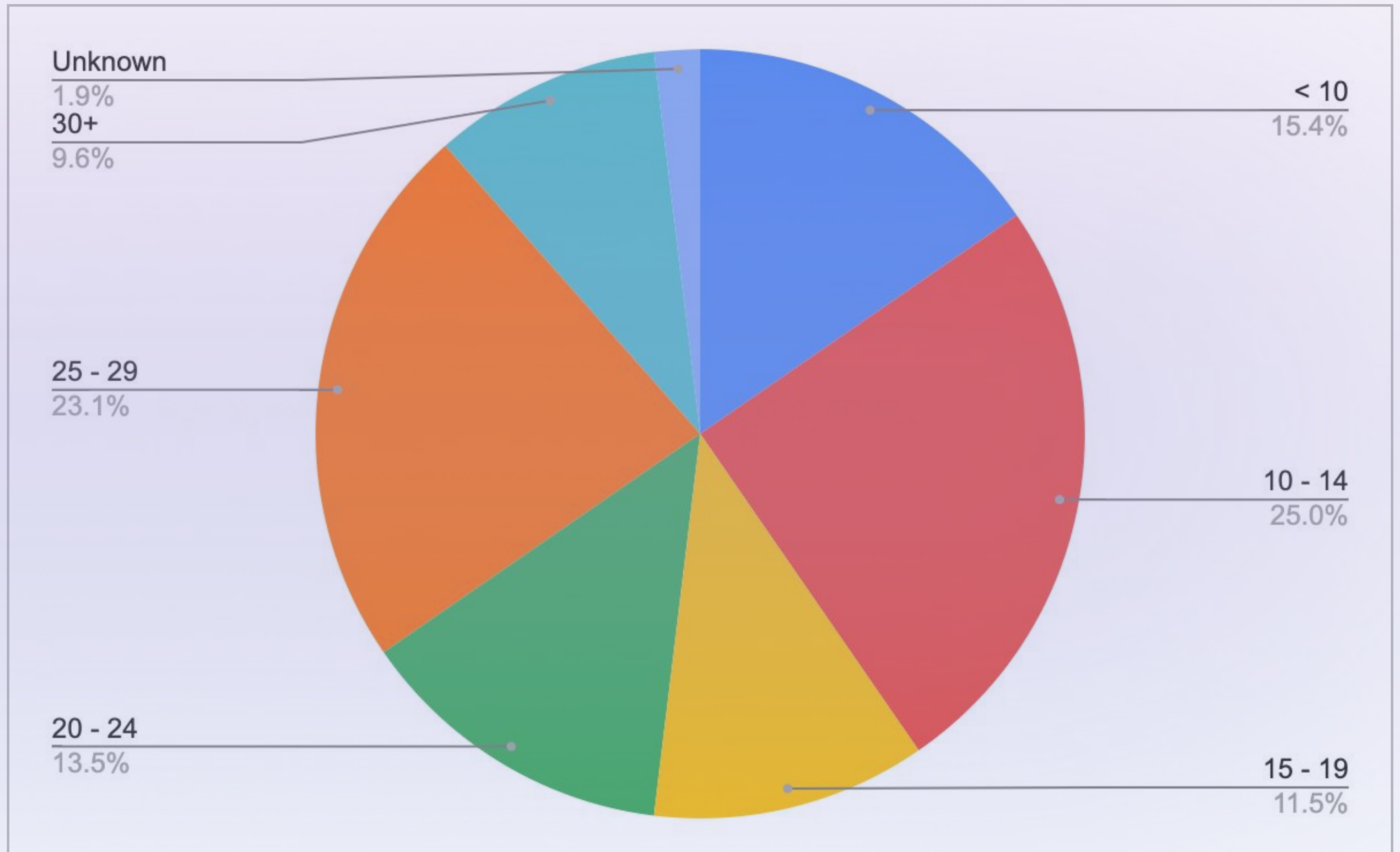
survey responses

- Role at company
  - **SRE (43%)**
  - Backend (17%)
  - Full stack (15%)
  - Other (25%)
- Years of experience:
  - **Mean: 18 years**
  - Range: 6 – 35 years
- Time at current company:
  - **Mean: 4 years**
  - Range: 2 months – 15 years
- 75% men, 10% women, 4% non-binary, 12% unanswered
- Size of on-call rotation
  - **1–5 (33%)**
  - **6–10 (46%)**
  - 11+ (17%)
  - Unknown (4%)
- Size of engineering org
  - < 10 (6%)
  - 10 – 49 (13%)
  - 50 – 99 (17%)
  - 100 – 499 (21%)
  - **500+ (42%)**

Team Size vs. Org Size

Team size	Org > 10	Org 10-49	Org 50-99	Org 100-499	Org 500+
1-5	2	4	5	2	5
6-10	1	2	3	8	10
11-15	0	1	1	0	3
16-20	0	0	0	1	1
More than 20	0	0	0	0	2
Unknown	0	0	1	0	1

Years of experience

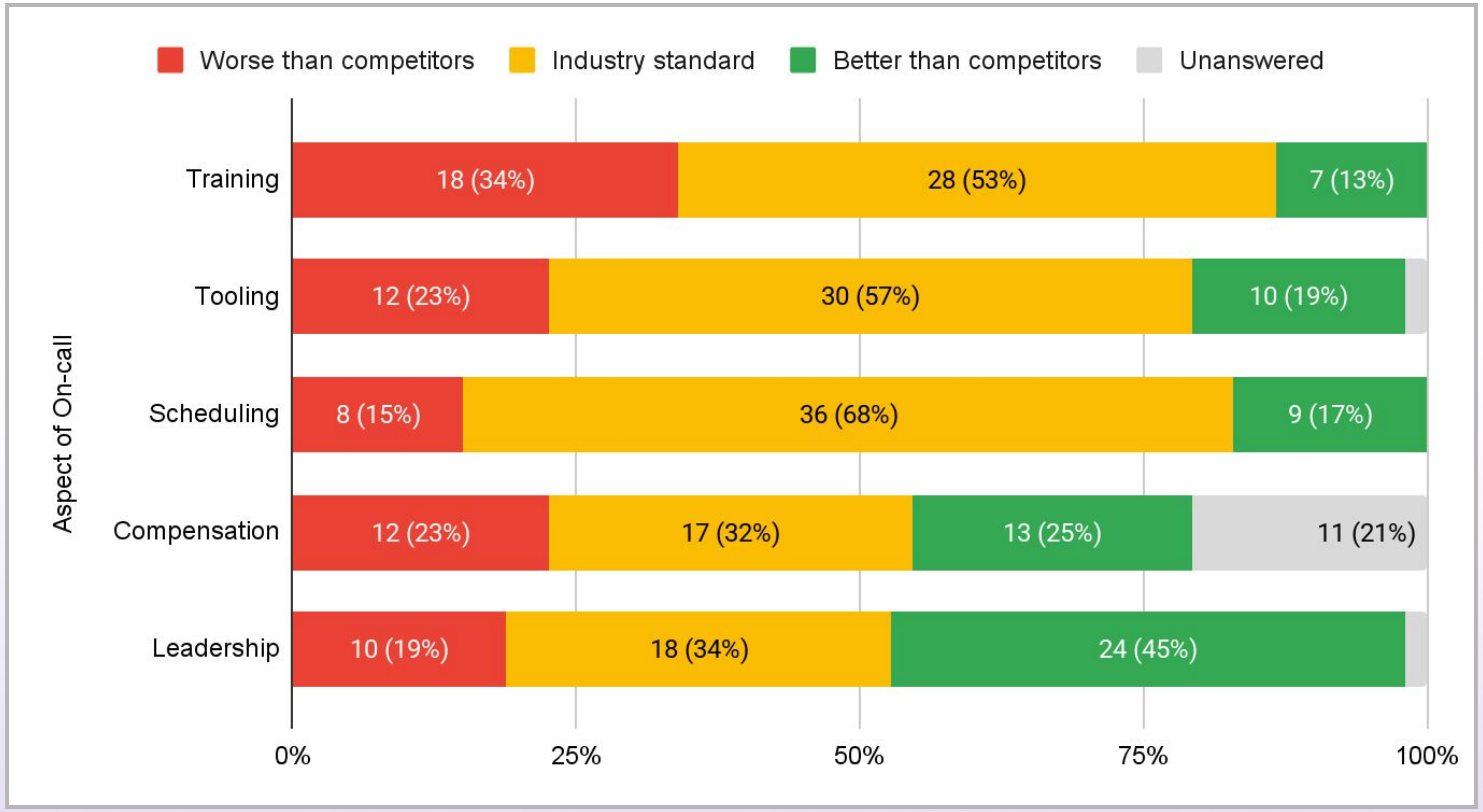


87% unhappy with status quo

👎

57% + 30%

😑 🦕



Majority of respondents were not happy with the industry standard… in any category!

Findings

The “no good” nine

Clunky tools

Noisy alerts

Limited agency

Onboarding & training

Reactive improvements

Complex or no process

Unsupported handoffs

Idealistic scheduling

Too many responsibilities

Onboarding & training

more unhappy with this than any other

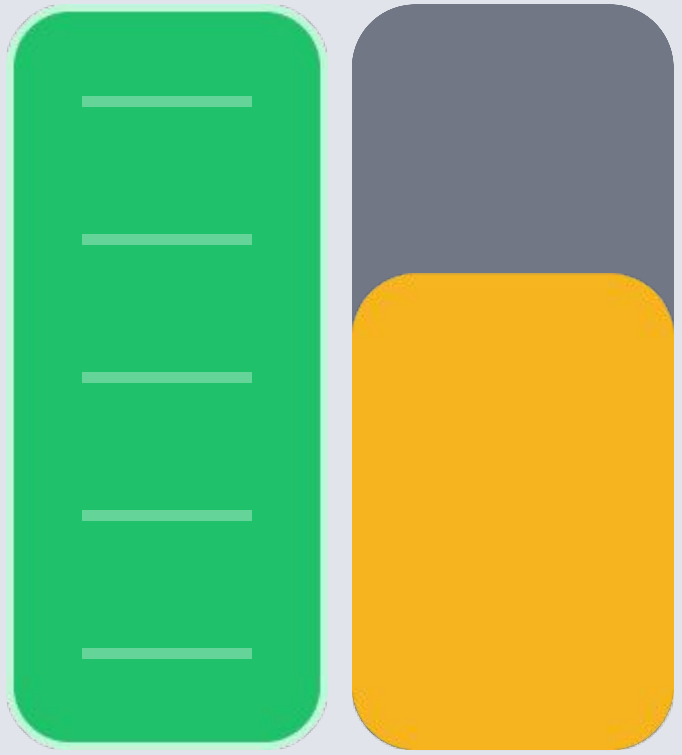
★★★★

2.35 of 4

Most 1 star ratings.

34%

worse than standard



only

13%

better

“We never get new team members, ***we just lose them*** *.”*

Reactive improvements

Fix what breaks

regular meetings?

404

No strategy

404

No measurement

404

No investment

“**On-call is a dynamic, living thing**. Due to the law of entropy, and the fact that computers are terrible, on-call has a tendency to get worse over time if you don’t take **proactive** steps to make it better.

You must build enough slack into the system to allow people to properly deal with the work that will come out of on-call.

- Ryn Daniels

How to create sustainable on-call rotations

Limited agency

20%

can’t or won’t update alerts

🚨 ALERT! 🚨 ALERT! 🚨 ALERT! 🚨 ALERT! 🚨 ALERT! 🚨 ALERT!

T! 🚨 ALERT! 🚨 ALERT! 🚨 ALERT! 🚨 ALERT! 🚨 ALERT!

permissions blocking improvement

can't fix

only escalate

stakeholders blocking

only 30% support overload strats

(well... sometimes)

Complex or no process

7 said

improvements were not prioritized at all

deployment tracking

response SLOs

escalation paths

what to alert on

34% said

team doesn't monitor how their on-call practices/tools are working

“It’s atrocious; it’s too complex, on-call is hampered by too much process and too few permissions, and customers shoot themselves in the foot ***but it’s cloud so it’s our foot.*** *”*

Unsupported handoffs

28%

Never expect a handoff

11%

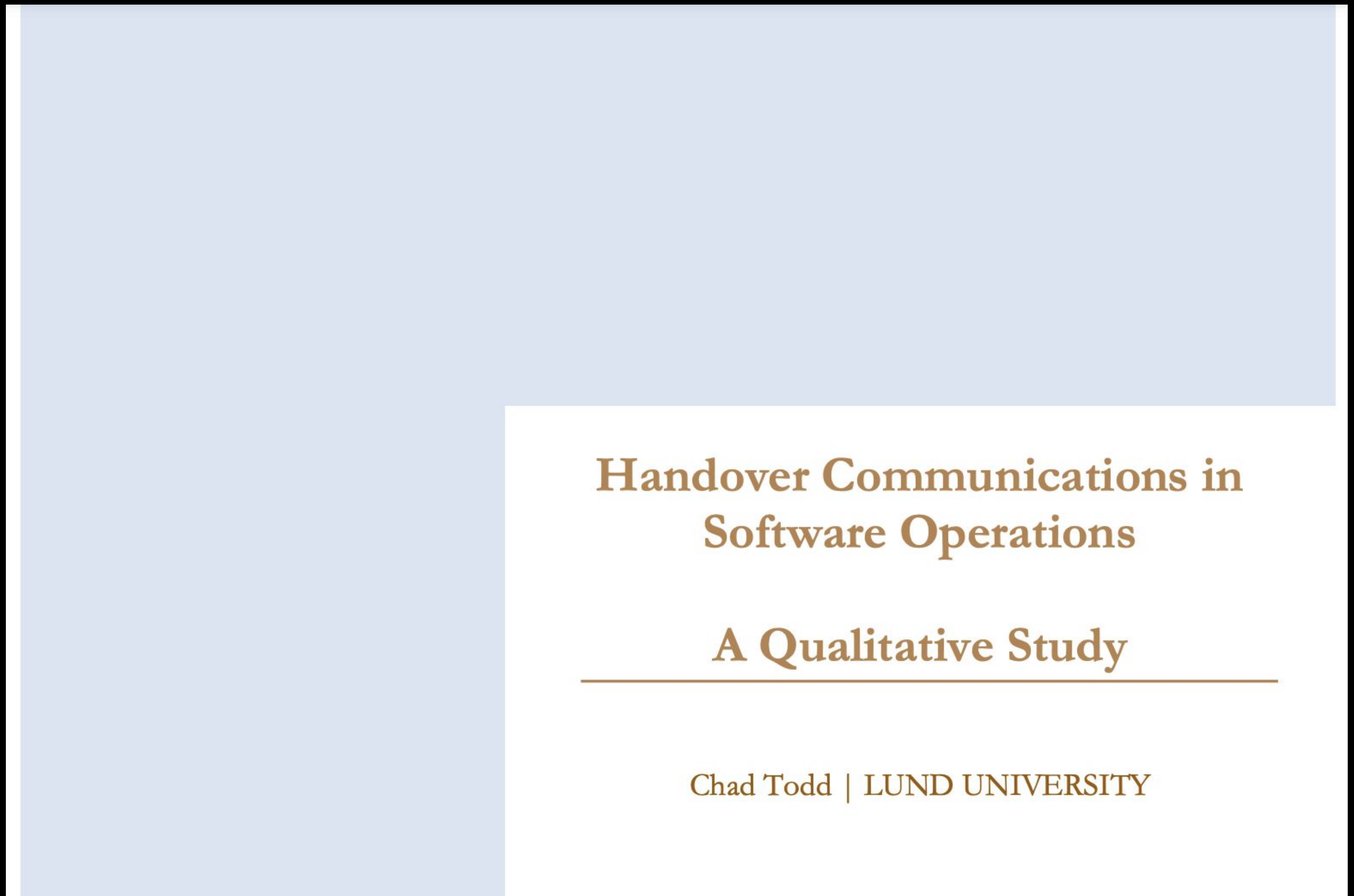
“sometimes” get handoff

😴

have a

GOOD NIGHT

“The complexity and uncertainty of information provided during [handoff] can effect an engineer’s confidence in understanding the current state of the system”



Handover Communications in Software Operations

A Qualitative Study

Chad Todd | LUND UNIVERSITY

Idealistic scheduling

2.85 of 4

Shift frequency

2.83 of 4

Schedule management

53% 😥

sometimes or always feel anxious about on-call

Difficulties changing schedules

24/7

rotations for 1 week seen as fair

1 2 3 4 5 6 7

Too many responsibilities

SLAs not sustainable

artificial
not actual
urgency

can’t get ahead of the workload

efficiency
thoroughness
trade offs

run the bases

run the bases

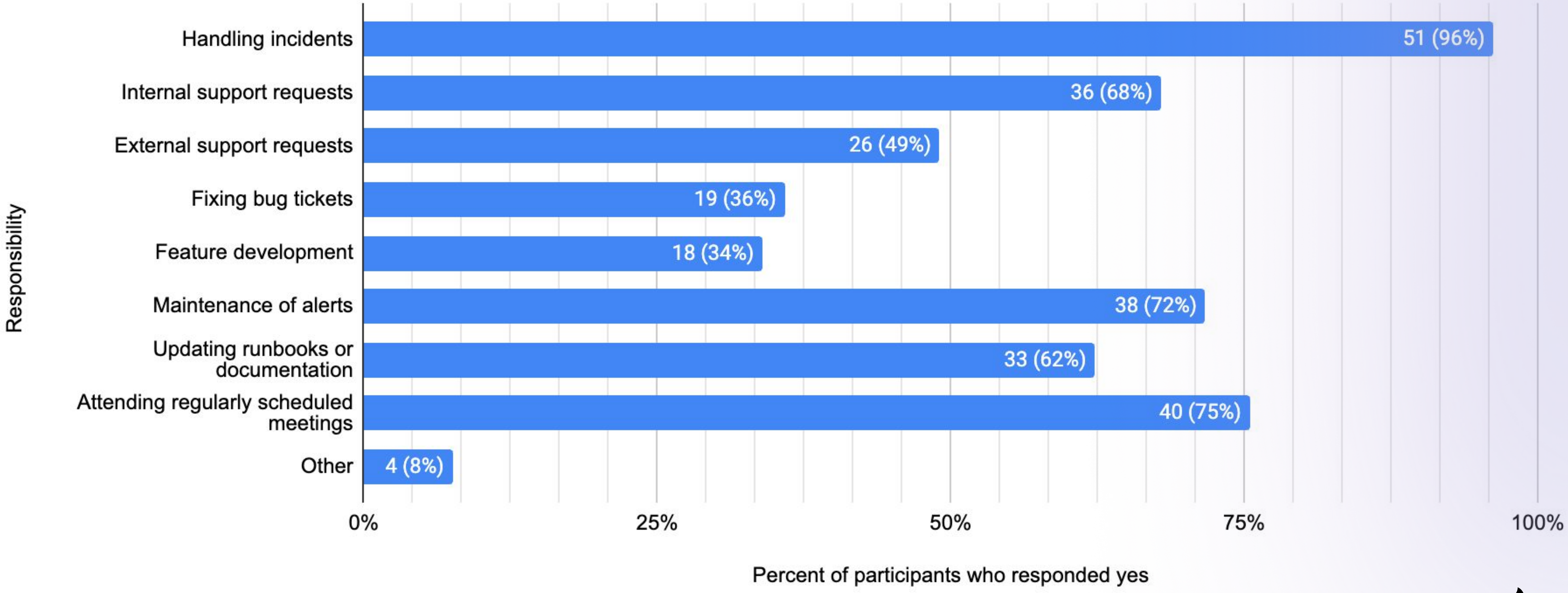
run the

engineers repeatedly dealing with the same problems

hotpot.works

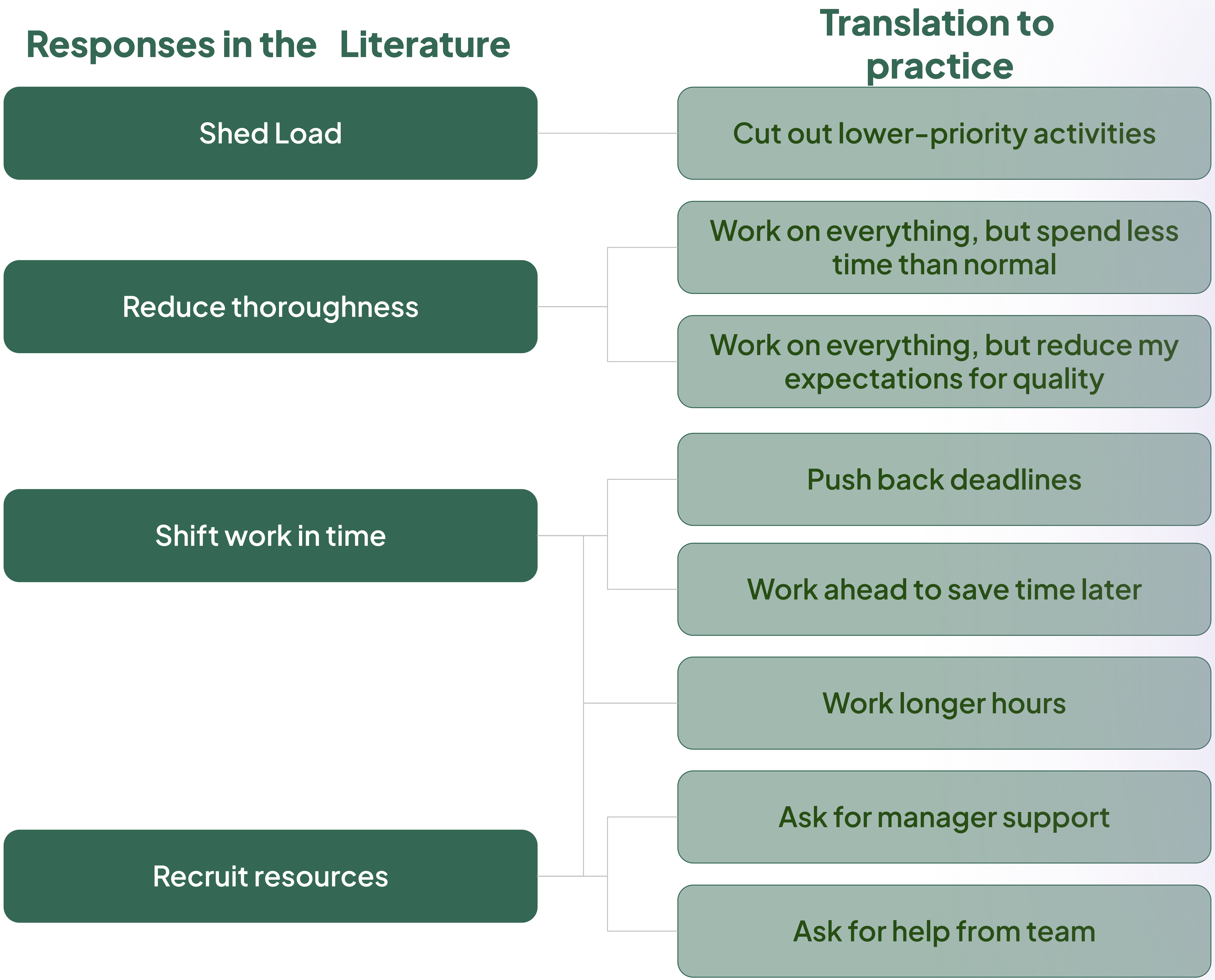
Too many responsibilities

Average of **4** responsibilities in **addition** to on-call



Responses to overload from too many responsibilities.

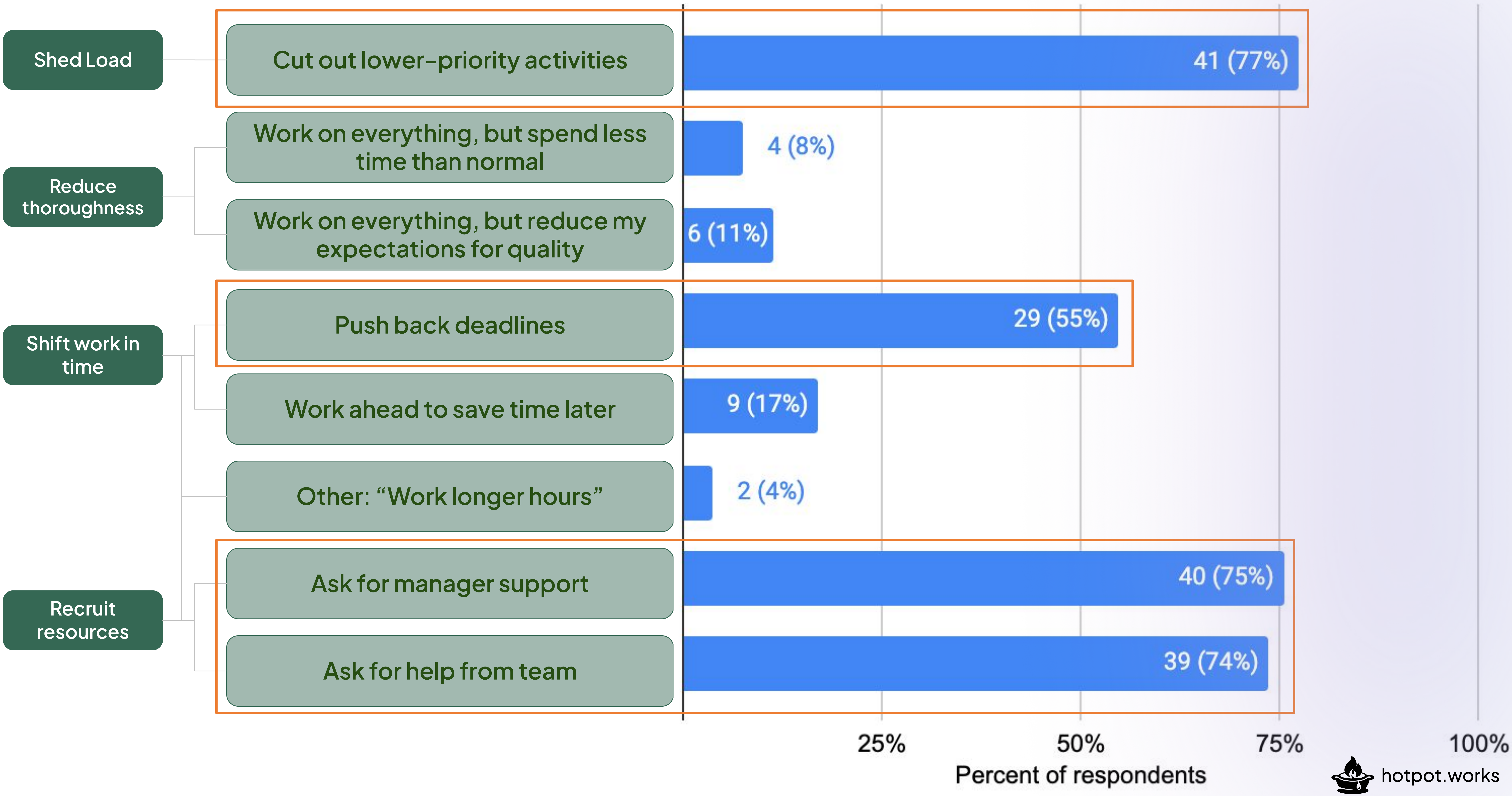
It’s not you.



Responses in the Literature

Translation to practice

- Shed Load
  - Cut out lower-priority activities
- Reduce thoroughness
  - Work on everything, but spend less time than normal
  - Work on everything, but reduce my expectations for quality
- Shift work in time
  - Push back deadlines
  - Work ahead to save time later
  - Work longer hours
- Recruit resources
  - Work longer hours
  - Ask for manager support
  - Ask for help from team



hotpot.works

“My biggest complaint about on-call is that ***it's high-risk, low-reward****.*

You have to work very tactically to avoid making your management chain or customers upset. ***It doesn't help your performance review and it won't get you promoted.****”*

The “no good” nine

- Clunky tools
- Noisy alerts
- Limited agency
- Onboarding & training
- Reactive improvements
- Complex or no process
- Unsupported handoffs
- Idealistic scheduling
- Too many responsibilities

On-call engineers and line managers **carry the costs** of poorly structured on-call programs and tooling.

74% of engineers we surveyed reported experiencing overload, burnout, or both in part due to on-call responsibility.

My Kafka story.

People **work together** to **overcome** the organization's shortcomings.

Teammate support

⭐ 3.33 out of 4 stars

~50% rated both their teammates and managers as 4 out of 4 stars.

Manager support

⭐ 3.15 out of 4 stars

83% can **confidently and consistently count on teammates** to help troubleshoot, respond to incidents, and cover

87% reported their manager will help them out when they are overloaded by on-call by:

- **adding someone else to help (most common),**
- taking the pager for the night, or
- taking them off call (least common)

~~Individual engineer~~

~~Manager~~

It’s the Industry

In our survey, being responsible for “real work” was the most common reason for being overloaded and overwhelmed.

most common reason for overload:

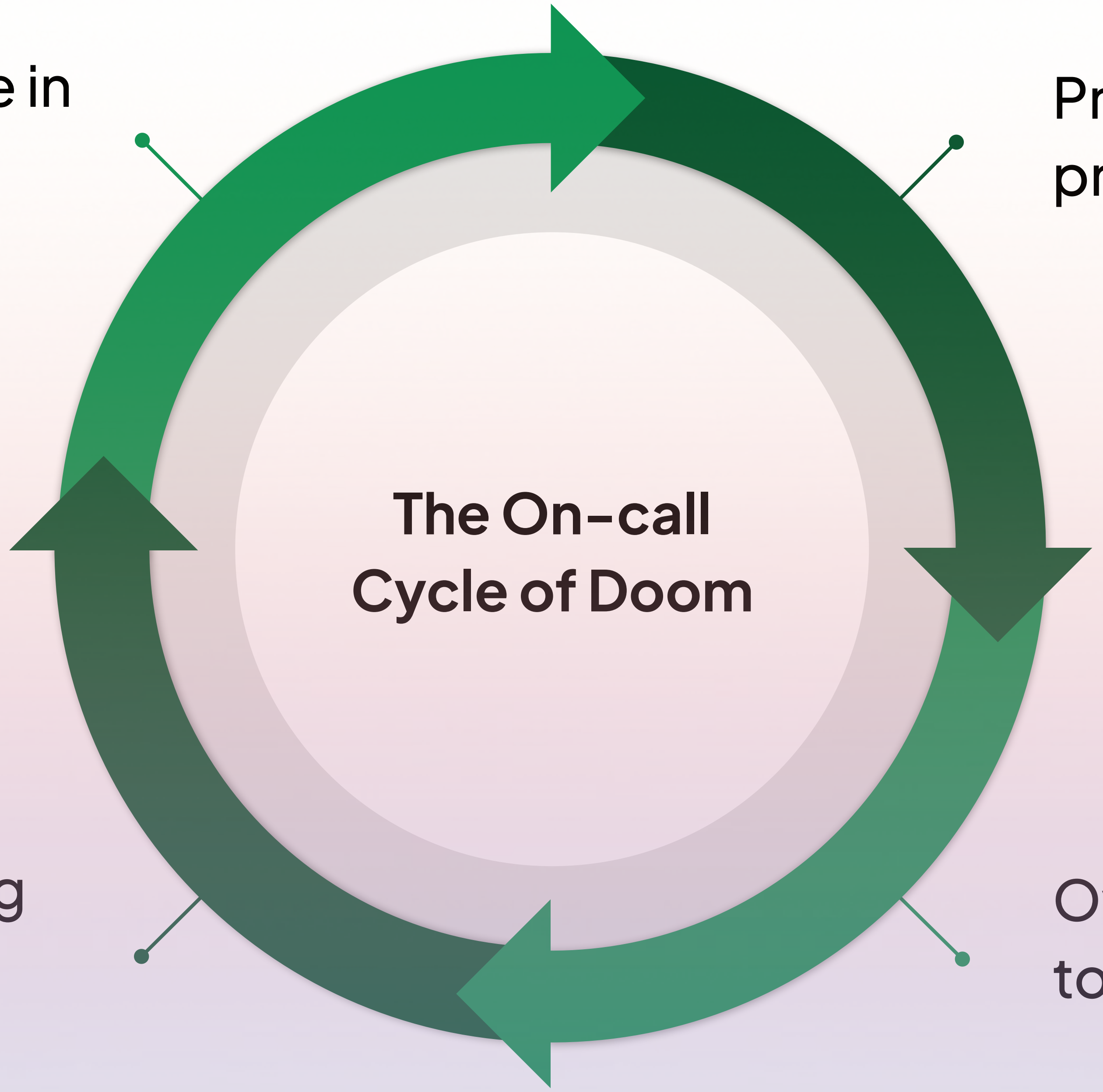
“real work”

- **34% are responsible for their regular feature development work** while juggling the responsibility of on-call.
- The other **66% are redirected toward on-call work**, halting their “real work” entirely.
- Typically is **not considered in yearly performance reviews**.
- “even though it takes **6-8 weeks out of [their] year.**”

The On-call Doom Cycle

Deadlines remain despite decrease in productivity, increasing the production pressure.

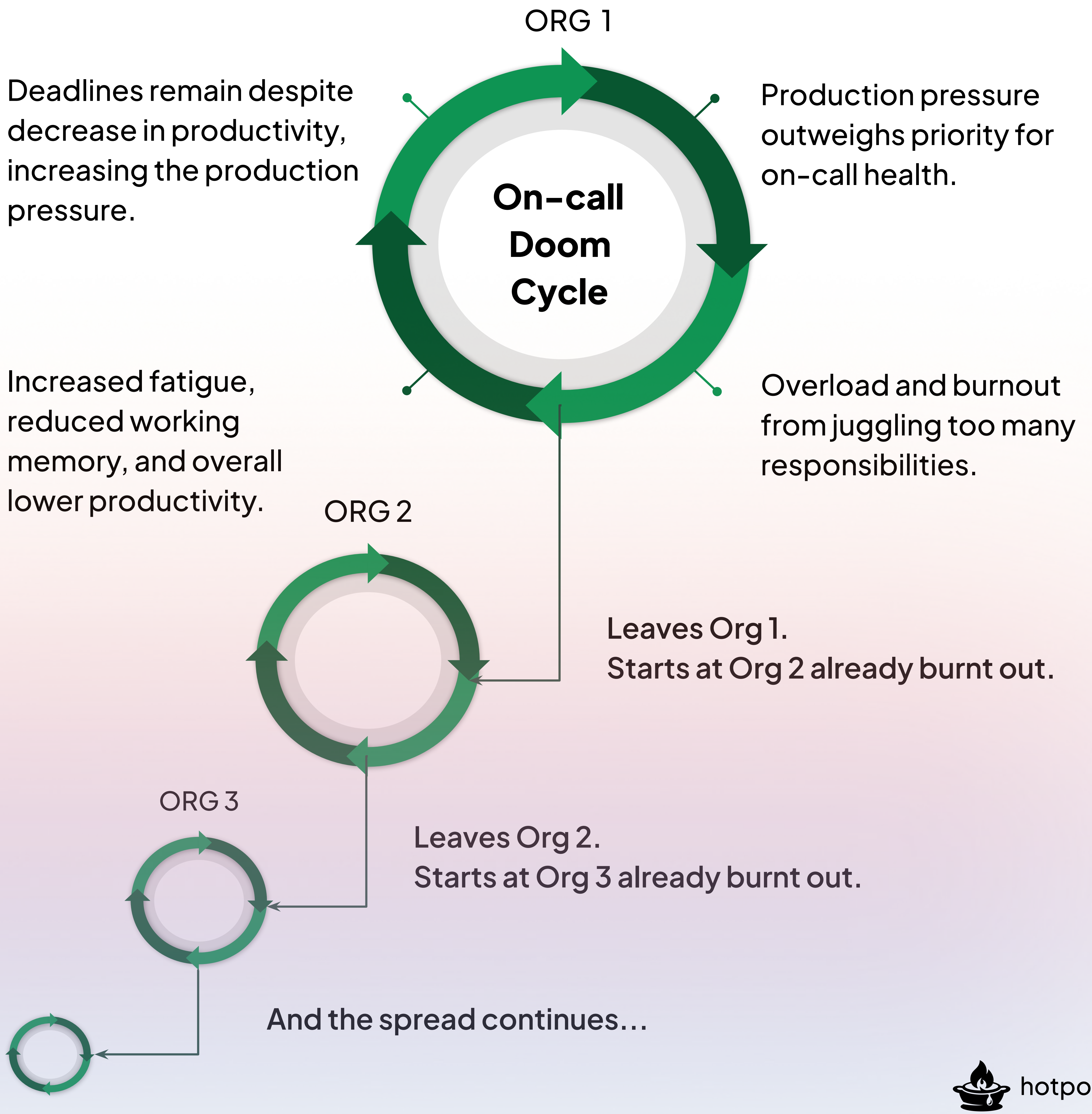
Production pressure outweighs priority for on-call health.



Increased fatigue, reduced working memory, and overall lower productivity.

Overload and burnout from juggling too many responsibilities.

We are literally trading burnt out engineers across companies.



ORG 1

On-call Doom Cycle

Production pressure outweighs priority for on-call health.

Overload and burnout from juggling too many responsibilities.

Increased fatigue, reduced working memory, and overall lower productivity.

Deadlines remain despite decrease in productivity, increasing the production pressure.

ORG 2

Leaves Org 1.
Starts at Org 2 already burnt out.

ORG 3

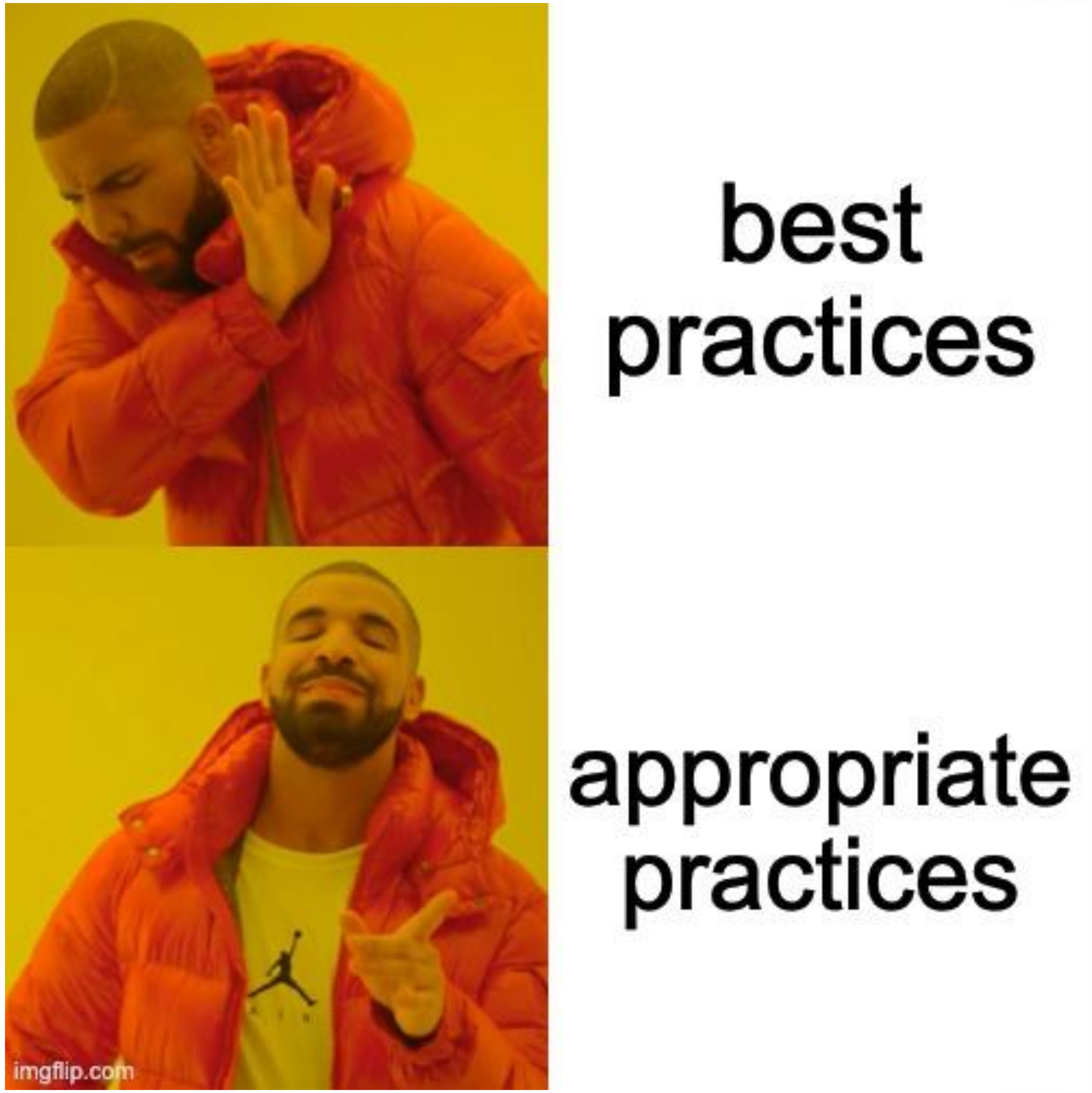
Leaves Org 2.
Starts at Org 3 already burnt out.

And the spread continues...

“If there is a [rotation] and it is well organised it is fun. It's good to make a difference in such a direct way.”

Recommendations

What you can do, today.


best practices
appropriate practices
imgflip.com

habits >wholesale

Processes, responsibilities & agency

“Remember, **bandaids are not real solutions**. If something catches on fire at 3 a.m., the on-call engineer might throw a quick hack into place to tide things over until morning (nobody’s likely to be doing their best work in the middle of the night) but **they need to have time to actually fix those hacks later.**

On-call will generate unplanned, interrupt-driven work by its very nature.”

- Ryn Daniels

How to create sustainable on-call rotations

Complex or non-existent processes

- Develop or assess:
  - Playbooks
  - Metrics
  - Release engineering practices
  - Software deployment tools
  - Escalations policies and tooling
  - Observability/monitoring
- Establish engineering risk management practices
- Build learning and refinement into your processes with post-mortems

Responsibilities & agency

Deference to the most relevant expertise

Distributed decision making

Investments in resilience & agility

Be proactive about processes and alerts

- Allocate time, even if it’s small
- Improve the documentation for on-call when a gap is found
- Regularly audit alerts, why do they exist?
- Schedule game days

Clunky tools

- Don’t settle for good enough
- Prioritize usability
- If you can’t yet replace, pilot

Training & onboarding

- Bring it into the light!
- Provide multiple opportunities for onboarding
- Formal training, reading incident reports, shadowing/being shadowed, observing incidents, reviewing playbooks and change logs
- Provide a mix of formal and informal onboarding/training activities
- Consider both technical and coordinative skills development

Scheduling

- Consider flexibility in your schedules
- Big deployments or migrations coming up? Holidays? Schedule for shorter shifts to spread the load out and reduce potential burnout on any one person or team.
- Proactively monitor for overload
- Design your schedules to account for “life” happening

Evidence based handoff strategies

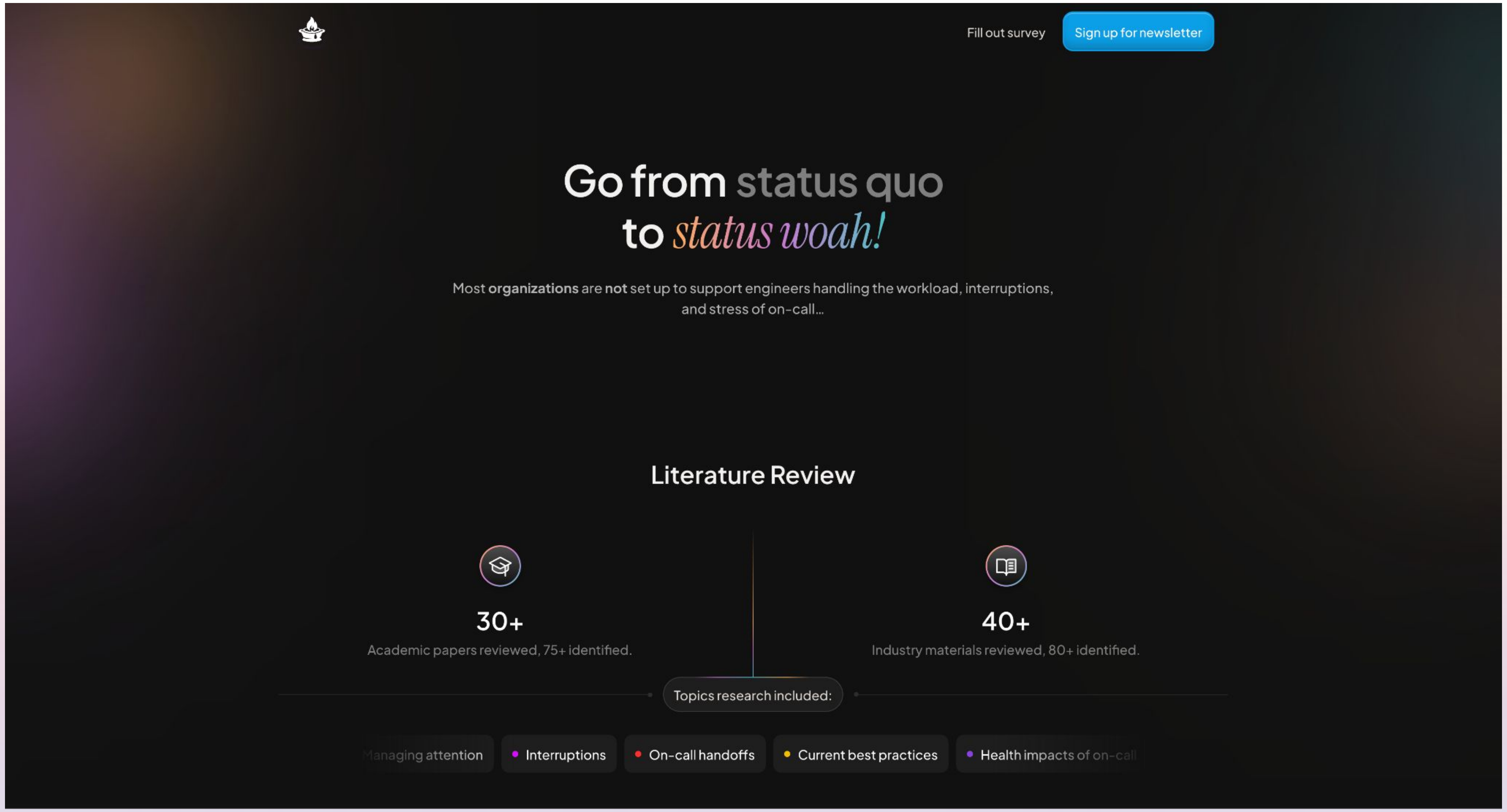
- Finding: Only **1.4 average activities** to prepare
- Verbal handoffs are more effective when they occur after a review of current state (checking logs, chat channels etc)
- 15–30 min overlaps between shifts required for context sharing
- Teach engineers how to more effectively handoff on-call
  - Successful handovers driven by the incoming responder actively questioning specific topics
  - Prioritize interjections rather than read-backs or explanations
  - Practice collaborative cross-checks especially when there is a power imbalance

Now what?

What’s next for our research?

- Trace the development of on-call programs over time
- Measure the impacts of on-call over time
- Industry discussion groups
- Sharing of *appropriate* practices

statuswoah.com



Fill out survey

Sign up for newsletter

Go from status quo to *status woah!*

Most **organizations** are **not** set up to support engineers handling the workload, interruptions, and stress of on-call...

Literature Review

30+

Academic papers reviewed, 75+ identified.

40+

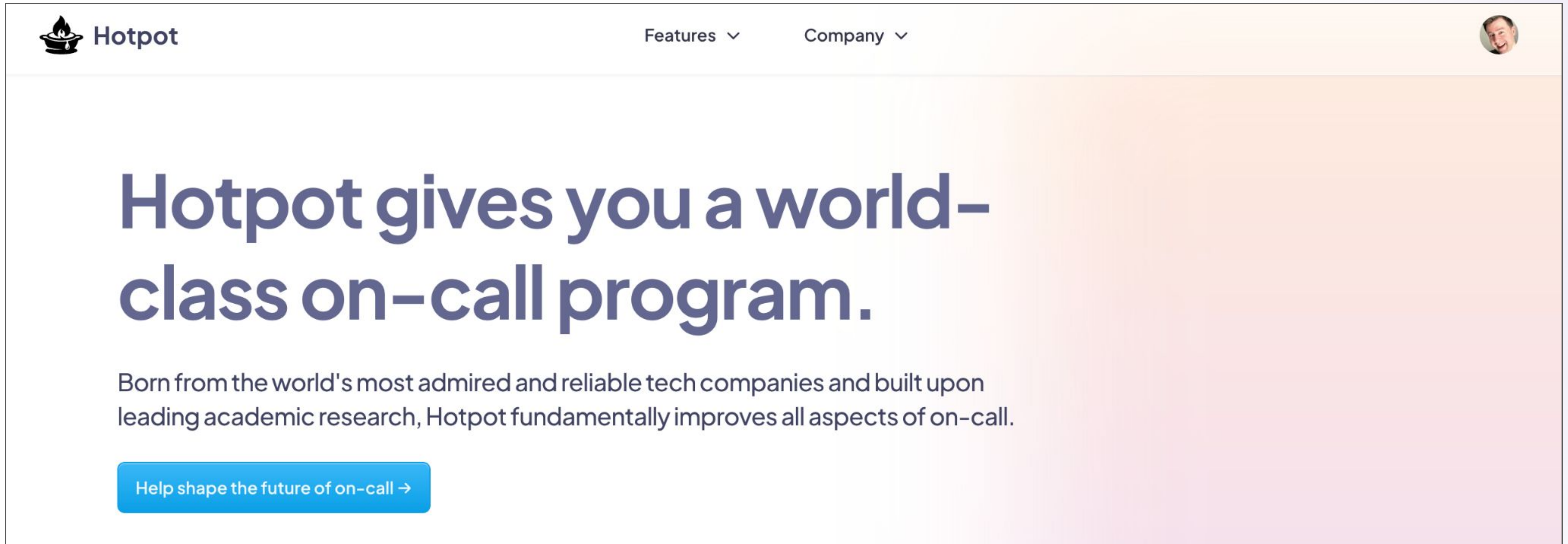
Industry materials reviewed, 80+ identified.

Topics research included:

- Managing attention
- Interruptions
- On-call handoffs
- Current best practices
- Health impacts of on-call

Thank you!

- Survey responders
- Signal boosters and reposters
- **Dr. Laura Maguire** and **Dr. Morgan Reynolds**
- Check out hotpot.works!



Hotpot gives you a world-class on-call program.

Born from the world's most admired and reliable tech companies and built upon leading academic research, Hotpot fundamentally improves all aspects of on-call.

Addendum

Literature Review

- Identified:
  - 75+ academic papers
  - 80+ industry materials
- Reviewed:
  - 30+ academic papers
  - 40+ industry materials
- Topics
  - On-call handoffs
  - Interruptions
  - Managing attention
  - Health impacts of on-call
  - Current best practices

Survey

- 65 questions
- 53 participants
- Topics:
  - Perception of current on-call
  - Current state of their on-call program
  - Training & expertise
  - Personal & professional impacts of being on-call
  - Improving on-call programs

We're in this together.

> *"We have a policy that we informally ask in a team channel for cover, and people are very good about offering to take shifts as needed."*

Engineers reported **high satisfaction with teammate and manager support**.

- Support from teammates averaged 3.33 out of 4 stars. Support from managers received 3.15 stars.
- **~50% rated both their teammates and managers as 4 out of 4 stars**.
- 45% think their leadership support is better than competitors

Most engineers feel supported by their manager; will support them during on-call shifts.

- **87% reported their manager will help them out when they are overloaded** by on-call. Forms of support:
  - **adding someone else to help (most common),**
  - taking the pager for the night, or
  - taking them off call (least common)
- **79% indicated that managers participate in on-call** (responding, incident commanding, monitoring, advocating for)

We’re in this together.

> *“If someone gets called a lot, we take them out of the rotation for a quarter. This is not an official policy and requires the Eng Manager to notice and action it.”*

Engineers count on teammates for handoffs, training, and support.

- 85% reported that **training includes learning from more experienced teammates** (e.g. shadowing)
- 33% surveyed rely entirely on a handoff from the outgoing engineer to prepare for their on-call shift
- 83% can **confidently and consistently count on teammates** to help troubleshoot, respond to incidents, and cover

Identified

75+

academic papers

80+

industry materials

poor affordances so novice or infrequent users struggle

2.65 of 4 ⭐

Clunky tools

clumsy, stubborn defaults

difficult to see key details

43% woken in last 3 months by self-resolving or inactionable

Noisy alerts

Interrupt workflow and degrade productivity

at least 15m interruption