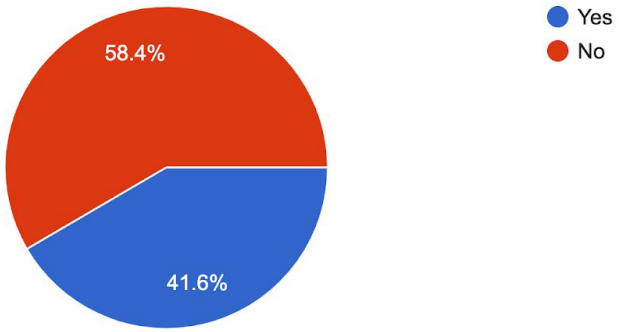
SRECon2019 On-call Onboarding Survey

Did you feel ready before going on-call for the first time?

(89 responses)



If not, why didn't you feel ready?

There were too many pieces to this complex system and I didn’t feel as though I was prepared to react to all scenarios
There were a lot of areas in our infra that I didn't know.
Felt there were too many things I didn't know or understand
Didn't know enough - including who knew what.
No onboarding
Was the only person
No training
Lack of training
Tribal knowledge
lack of information, lack of tooling and access
Not sure what to do. Not sure if I could help or would be able to get up late at night.
did not yet know the apps/environment well enough
Unclear documentation of where runbooks are, lots and lots of services that may have alerts
Never knew the system enough
nobody told me what anything meant

Too much environmental complexity, limited documentation, high stakes
Was a new Incident Management function and we were learning as we went.
No enough knowledge of system supporting.
So much to learn!
Didn't understand the service well enough. Felt like I would need to escalate often.
Felt like I had a lack of knowledge
Lack of systems knowledge
Started on-call my first week at the company
Did not know most operational prodecures
Complex system with too many dependencies
I felt that I had insufficient knowledge of the system to handle all issues that could arise.
need more system details and infrastructure understanding of applications
Undocumented alarms and non-actionable ones
Well, I need to understand the expectations, how to react, what to prioritise and escalation process
had no idea how to fix half of the things that might happen
It was at the start of my career and my manager's only instructions were don't wake me up.
so many unknowns
Sure things I didn’t know would overwhelm
Tools
I was on call before I had a good mental model of the system; I didn't know what the expectations were for the on call engineer
Because production incidents are so varied you never really know what's going to break or how it's going to break
Lots of systems and I was junior
I didn't know how to deal with certain types of outages
uncertainty. what happens if I don't know how to fix the problem
I don’t have context sometimes
Inexperience
I hadn't handled production errors before.

There was no-one around me who had ever been on call, it was the blind leading the blind.
Anxiety about failure to resolve in time
Too much unknown
I didn't know what to do, at all
No actual onboarding
Broadness of problems that could have to be solved.
I was shared. Wheel of misfortune helped get me more prepared and understand the tools

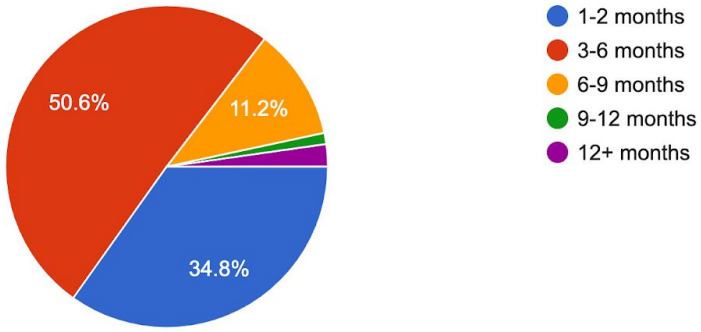
What do you wish you knew before going on-call?

Architecture of everything
Better familiarity with communication processes and tooling.
Better understanding of the systems that am oncall for
Clear agenda
Clients aren't looking for the most technical details
common failure states and how to recover from them gracefully, best people to reach out to when we need to hit the "panic" button
criteria, runbook
dont know what I dont know
Engagement model
Escalations are free/cheap.
Everything, structured training like Google
Greater depth of knowledge in production system internals.
have a good runbook, how to handle comms
How best to balance family life and oncall responsibilities
How it would affect my life.
How often I would be called? During the week? Weekends? Holidays?
how the fuck this system works

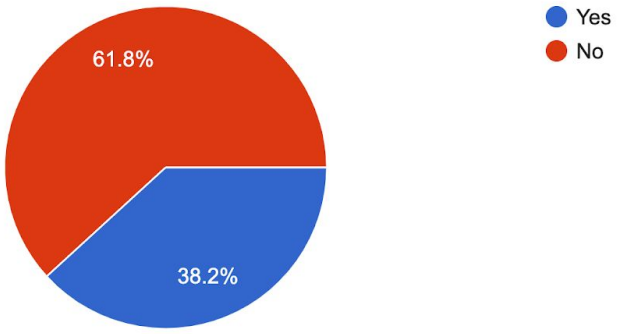
How the services I didn't know about worked
How to better engage teams I had dependencies on.
How to escalate / pass on if I was not capable to fix
How to find an employer that didn't have so many outages and had more structured on-call for the few outages they did have.
How to get help
How to get help from matrix teams.
How to prioritize stabilizing the system
How to respond to events
how to unfuck all the things
How to use our monitoring tools.
I was confident in the number of run-books we have ready to go.
I wish The alerts link to a run book that gives a general overview and link to metrics to see history
Inf full picture and applications
it doesn't have to be noisy af
It really isn't that bad.
It will be ok.
It would be 24/7/365... for the next 6 years...
It's ok to let the page escalate if you don't know how to handle it
More debugging strategies specific to our application, and context about the app's development
More exercises to prepare, more daytime only shifts.
more hands on exp with the apps' behavior
More in-depth training on App architecture/dependencies
More robust architecture documentation and known gotchas
No one is ready to go on-call the first time
Noise
Not all pages are emergencies.
People walked me through the process before. Also, many alerts are noise.
Potential Workload
That 50% of the pages could be ignored

That it's not the end of the world
That it's okay to not know the right answers and to vocalize that.
That nobody knows everything about a system!
That other people don't know everything
That the sense of ownership it instills sticks with you even when you've moved onto other roles that don't require the same level of service.
the most effective way to triager alerts
What can and can't cause data loss in our system
What I was doing
what the upward escalation path was
where all of the runbooks were located
Who owns the other services that also broke
You don't have to fix the problem, but you have to get it fixed. That is, you don't have to know everything.
You’re never alone
You’re not gonna be able to always do it alone
Your life is hell when on call

How long does it take for an engineer in your team to become part of the on-call rotation?



Do you have an on-call training process?



What's the one thing you would improve about your team's on-call training?

Anything
A structured learning curriculum.
Teach major incident conference call behaviors. How to manage troubleshooting, collaboration, and status updates. Knowing audience and being cognizant of good vs derailing comments
Adding shadowing or doing some kind of pair on-call
Have a structured training!
More active shadowing.
Create on-call training.
Dunno
Have a training
Make the Noise Stop
Preparation
just more information and better documentation
longer shadowing
Have training.
do more pair-style oncall response so the trainee sees hands on what is being done
Practice outage scenarios.
A structured curriculum with milestones and accountability would go a long way in improving the time-to-first-oncall, and the confidence level of new on-callers.

have a formalized document for new on-call engineers, on top of the existing runbook
Formalize when/how people are added to the rotation
More training in general - too much is implicit.
Training specific to being oncall, tools and processes to manage live site issues.
It's adhoc - could be better documented and presented
Formalize it.
Shadowing during the on call process
documentation
Better able to find information.
Commit to the IMS model and practice it regularly.
More product and systems knowledge - or - better runbooks
Formalize it
Community chatops
More "live" exercises
Like to see addition of Chaos exercises
Force all alarms to be actionable and documented w/ runbook
Reduce the calls
good question
Having one
more documentation, daytime rotation for new people.
Learn more about the environment they are responsible for
we'd establish a process; we've only been around for three months and all three founders are on-call
Clearer expectations; better process to ease people in; shadowing opportunities
Documentation
Have strong documentation about previous incidents
More practical wheel of misfortune exercises
Not to repeat status update in the all
We need more “wheel of misfortune” type of training.
Ability to put things in a less technical/high level context
Add shadowing and fake situations (or real ones from the past) to train with.
Accountability and expectation. Some folk don't come online, meaning imbalanced workload, and there is no tracking or hard guidelines around this .
Existing
Finishing the alert to run book linkage

Have some
Artificially creating incidents to test what you know (dirt, etc)
Have clear expectations about what you need to know to be on-call
There is training on the system used for on call, but no clear expectation of what the on call engineer should do outside of acknowledging the alert.
Better runbooks
Better define expected situations and train for them.