Training new Incident Commanders… with Pokemon!

Laura de Vesine
silverrose@datadoghq.com

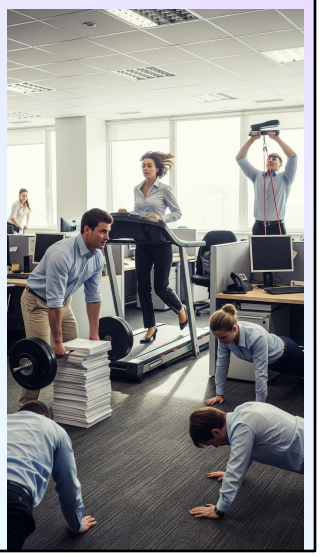
So before I start, nothing in this talk has been approved by Nintendo, and all references to Pokemon are to the extremely generic cultural understanding and not to be confused with the actual trademarked games in any way. That out of the way… Hi hello you're all awesome. In fact, you're so awesome (and you're SREs) that I'm going to assume you're really really good at incident command (as well as other skills, plus being good looking and exactly the right amount of weird). [click]

Training new Incident Commanders is Hard

How did you get good at incidents?

- Training
- Practice

SRE teams can devote regular time to incident management skills



But… how did you *get* good at incident commanding? A lot of it is by practice, of course – being involved in incidents with more senior folks who you can observe and guide you. And you've probably had some training.

It turns out, though, that training incident commanders is hard. We find (and actual research has also found, cf https://www.usenix.org/conference/srecon24emea/presentation/silatani) that engineers in a training (and in real incidents) want to focus on debugging the problem, not on incident command specifically. Sometimes, trainings literally just devolve into arguments about the accuracy of a possible failure scenario, or deep technical dives into particular services. While deep technical dives are valuable, it means that trainees are learning about a specific service, and not "how to coordinate an incident". As SREs, incidents are part of our core job, so we can devote regular time as a team and as individuals to practice and training, which is at least how I learned to get better at this skill.

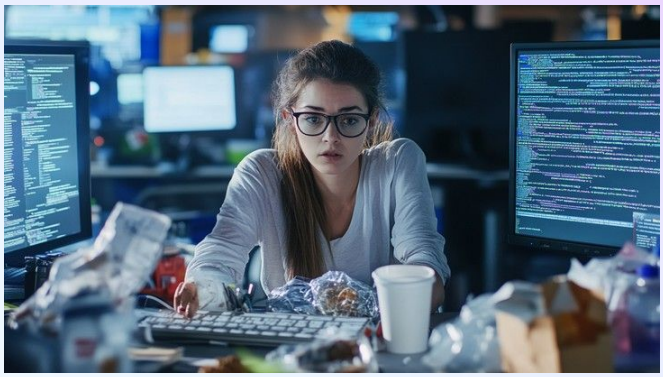
[image AI generated]

Especially non-SREs

Primary job is features, not incidents

Not always clear on incident command as a distinct skill

Limited training resources



DATADOG 3

At Datadog (and many other places), we run a more “devops” style [click]. Most of the people who respond to and even command our incidents are not “SRE”s. They’re software engineers whose primary job is writing product features, not operating software (including incident response). So asking teams for something like a weekly or even monthly practice at incident response is often a non-starter. [click] They’re often not clear on incident command and coordination as its own distinct skill, vs. “live debugging”. All the pieces of an incident that involve coordinating responders, communicating to stakeholders, making sure that decisions are getting made in a clear way tend to be afterthoughts at best. [click] And any training that we offer to help our developers be better incident commanders needs to scale, both practically and technically (in the sense of not being specific to individual teams’ services), because our small SRE team has ~350 teams to potentially train and coordinate.

We want to build a training that has both theoretical and practical elements. Attendees get about 45 minutes of “how to incident” lecture, and then about 45 minutes of “incident handling practice”. But how can we manage and scale that practical scenario so that we’re not learning everyone’s services and mostly doing live debugging exercises?

[images AI generated]

What if…

We build a training that's entirely non-technical
Relies only on "common sense" knowledge

And that looks like an incident
- Urgent
- Many reasonable solutions
- Challenging and dynamic environment
- Uncertainty
- Coordination between teams

DATADOG 4

So, why don't we address this "technical derailment" and scaling challenge by building a training that's actually entirely non-technical? Ideally it should rely only on common sense knowledge that is as culturally agnostic as possible (in an internationally distributed company, we want to be able to include and support everyone). We also want to make sure that the scenario we're building on isn't going to hit any obvious traumatic points for anyone – so no natural disasters, fire, or human violence, some of which are "obvious" sources of incident scenarios.
But we need it to quack like an incident [click, then for each] – it should be urgent to solve, with multiple actions and solutions that might be reasonable to try. The environment and situation to navigate should include meaningful uncertainty about what's happening, and be challenging and changing over time. Ideally, we should ask engineers to also practice the skills of coordinating between different teams that rarely work together, as that's a common place where incidents fall apart. So… what's a good emergency that's more silly than scary?

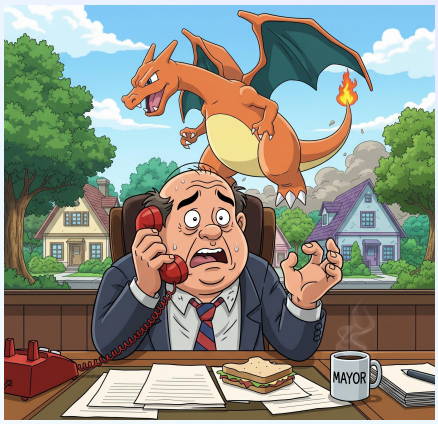
[images AI generated]

Enter Pokemon!

Summarize how Pokemon work (wrong)

Assign responder roles

“The mayor called…”

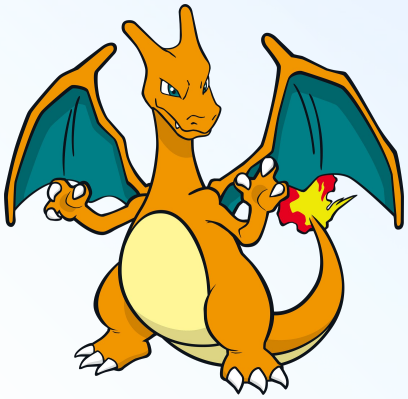


DATADOG 5

Well. At Datadog we name our k8s clusters after Pokemon, which turns out to be a great inspiration! While most folks have heard of Pokemon and already know how they work, we kick off the training with an (incorrect) summary of the world. As a level set, we make sure that trainees know to work from the exercise, not their real knowledge. We tell them “Pokemon are elemental monsters that have different types, one of which is fire. Pokemon can be captured by an experienced operator of a device called a pokeball. Pokemon exist in the context of the real world and can do physical damage to items in our world." Then we assign response roles as part of the “Consolidated Pokemon Rescue and Management Center”. We kick off the scenario with an alert to the oncall responder: "the mayor of a nearby town has called. They are worried about a stray Pokemon in the area. What would you like to do?"

Pikachu on bicycles originally posted to Flickr by jillccarlson at
https://flickr.com/photos/59229934@N04/27736300683
Mayor image AI generated
Pokeball public domain

The actual scenario



Dangerous Charizard
    Described by appearance, not name

Need a “fire” pokeball to catch
    None in inventory

Manager will suggest a wrong solution

Everyone wants updates

START 10-15 m 25m 40m END

DATADOG 6

So, here’s what’s happening and what our facilitator knows: [click]

- There is a particular pokemon called a charizard near the town. In the context of the training, “charizard” is basically a nonsense word – responders need to identify the right pokemon, but all the information they need is on their handouts.
- While the pokemon is not immediately hostile, disturbing it could cause it to become destructive – this lets the team consider user impact and mitigation
- [click] The pokemon is not described by name, only by appearance (a “big red dragon”) – this gives us uncertainty typical of an incident
- [click twice] The team can catch the charizard with a “fire” pokeball, but there’s not one currently in inventory, so it will have to be made at some delay – this allows cross-team coordination and ETA updates, as well as giving us a “wrong” solution the team can try
- [click] Midway through the scenario, your manager will join and suggest that it’s a different pokemon (a sandslash, which is just another nonsense word in context), which needs to be caught with a different “type” of pokeball
- [click] Throughout the scenario, various stakeholders will ask for updates

Image via https://www.deviantart.com/monstermmorpg/art/Charizard-649591769

Training roles

- Oncall
- Team expert
- Team manager
- Inventory team oncall
- Home team member

START 10-15 m 25m 40m END

DATADOG 7

To respond to the prompt, we separate out participants into groups of 4-6 with one facilitator. Each responder gets a handout explaining the basic facts they know and skills they have. The trainee group's roles are:
* oncall person (a newer team member who has some general knowledge but isn't confident in some of their skills)
* team expert (a long-time team member who is a deep expert in some areas, but rarely does "hands on" work)
* team manager (the manager has some business-related concerns, needs regular communication, and has some "technical knowledge" that turns out to be wrong)
* member of another team (needs to be paged in to provide essential resources for resolution)
* generic team member (no special knowledge, but is comfortable performing operations and can do any regular team task; good for delegating to).

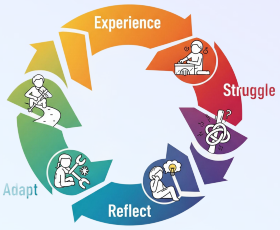
Images AI-generated

Facilitating

Facilitators learn too

Participants learn by experience and guided self-reflection

Facilitation is about driving that self-reflection



START 10-15 m 25m 40m END

DATADOG 8

The facilitator also has a handout guiding them through the exercise, with the understanding that they will respond with some flexibility to the choices made by the training group. [click] Facilitators also get a chance to learn and reflect on incident response; we find that guiding other groups through the exercise helps reinforce existing skills. [click] The job of the facilitator is to let the group learn by doing – allow them to “run with” the scenario, answering questions and providing answers where asked, but taking a passive role. [click]Then, every 10-15 minutes (so about 3 times during the 45 minute exercise), the facilitator stops the group, asks them to pause role-playing, and to reflect on how their incident response is going, drawing on some classic ways that many incident responders struggle.
Let’s look at those prompts

Images AI generated

Prompts at 10-15 minutes

Who is in what incident response roles at this time?
- Is it clear to everyone who the IC is?
- Who is handling communications?

How do we feel about the response so far?
- Did we triage the incident before starting to respond?
- Are we sure we have the right priorities?

START 10-15 m 25m 40m END

DATADOG 9

At our first checkin with the group, we have two basic goals, aimed around a chance to practice using the role assignment techniques they've learned – first, we literally ask "whose in charge" and "who currently has what roles". Second, we give a gentle reminder that the goal of the incident is to triage (figure out the impact, goals, and priorities), then mitigate (stop customer impact), and *finally* resolve. We check in with the group if they've followed this process… or if they've jumped straight to mitigation or (especially) resolution, which is a common way incidents can go off the rails early in real life. The facilitator is encouraged to give feedback on these answers if the group seems misaligned, but in general we find that just the opportunity for self-reflection is plenty to catch problems.

Images AI generated

25 minutes

Responders may focus on capture and forget other dangers
- The townsfolk are in danger. Should we do something?
- Is someone watching the rogue Pokemon?
- How are we avoiding angering the Pokemon?

ICs are usually feeling overwhelmed
- IC: do you feel in control of the scenario?
- How can we help the IC from being overwhelmed?

START 10-15 m 25m 40m END

DATADOG 10

After the group's first chance to check in, reflect, and correct we restart the scenario and let participants adjust and continue their response. We also ask The Boss to step in during this time and try to derail the response by suggesting some wrong solutions about which Pokemon the team is dealing with – if the team over-indexes on this suggestion, it will significantly delay the response. After another 10-15 minutes, we pause for another reflection check in. At this point we want to focus on the actual quality of response with our reflections:

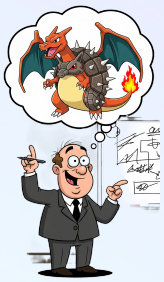
- have we remembered that there are elements of the incident around containing damage, not just solving the "cause" with capture?
- is the IC successfully delegating? how are other responders supporting the overall cadence and awareness of the response? Could we delegate more?

Images AI generated

25 minutes

Are we communicating successfully?
Would an executive understand the state of the incident?

How do we feel about The Boss helping?
How would we handle that during a real incident?



Is anything missing from our response?
And, what about our response is extra great?

START 10-15 m 25m 40m END

DATADOG 11

This is also the point where we ask about organizational communication, especially stakeholder management. We want the team to reflect on whether executives could understand what's happening, or if they'd have to interrupt the response to figure that out. We also ask them to reflect on how to handle a manger helping with the response, who might be taking up a lot of resources or leading in the wrong direction. And finally, we look for general feedback – things that are going well, and things that could be going better.

Images AI generated

Wrap up

What roles is everyone holding now?
Invitation to reflect on delegation

Are you happy with your response?
What can improve?

What did you learn today?

START 10-15 m 25m 40m END

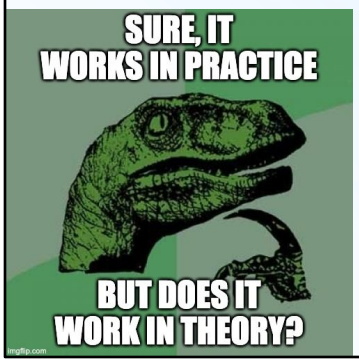
DATADOG 12

We restart the group one more time, and the facilitator is encouraged to either give help or add roadblocks (like unexpected system problems, or important responders needing to leave for personal reasons) to try and ensure the group just barely finishes their response. At the end, about 40 minutes into the exercise, we pause one more time to help everyone cement their learning. The main goal here is to help learners take away 1-2 major reflections to try and bring forward into future practice. The most important skill we want to emphasize is delegation, so we ask specifically about that. We also simply invite the group to share what they've learned from the exercise, and what they want to improve going forward, in an open-ended way.

Images AI generated

Does it Work?

90% of trainees report increased confidence
20-30% of trainees agree to be future facilitators
We're asked for more trainings several times a quarter



Ad hoc we see improved incident handling
- More active communication and delegation
- More user awareness
- Better "presence"

So, it's hard to directly measure the abilities of an incident commander – every incident is different and simple metrics like "incident time" don't give a good picture of what happened in a response.
What we *can* say is that our trainees clearly find the training valuable. They consistently report increased confidence in their own skills, ask for additional trainings of this type (we don't let them re-take the same one because we want to avoid training the scenario), and even agree to give their own time by volunteering as future facilitators. [click]
It's both hard and labor intensive to measure "incident commander quality", so I can't give you numbers here. Anecdotally, we see that engineers who have taken this training are more consistent about giving incident updates and trying to delegate response elements sooner. We also see that they are often more able to "take charge" of an incident room, but that is hard to separate from people who started with more of that ability and were interested in the training because of it.

Challenges

Only 1 training

Voluntary 90 minute training

4-6 person groups means we need many facilitators

DATADOG 14

So, some challenges we haven't solved yet
Right now we only have one training, so we can't give people repeat versions. We'd like to have more, but team priorities make it a challenge. We also have relatively limited uptake – the training itself is voluntary and 90 minutes long, since we cover both the basics of how an incident should go, then do the practice scenario, as one session. So while we have run around 100 engineers through since we started offering this training, that's only around 5% of our total engineering population. Using our graduates as facilitators helps us scale, but we still need a lot of facilitators because the training is run at a small group size, and the coordination and recruiting is an ongoing time commitment for the team. It's a worthwhile investment for having a pool of particularly excellent responders who can also share with their own teams, but it's important that we also pursue other less time-intensive training and documentation.

You can do it too

You can teach incident command without debugging
- Feel free to use this one
- Lean on fun

Self-reflection works
- Triage first
- Delegate more
- Don’t let your manager derail

Please: take this training back to your own teams! And if you invent new scenarios, I will absolutely steal them for our use if you let me. For us, the two biggest things we’ve learned from creating this training are that you absolutely can teach incident command (and response) without talking about computer systems at all. [click] we’ve also been reminded that our colleagues are smart people who love learning – if you have patterns where your responders struggle to succeed, try giving them a safe chance to self-reflect. We wanted to teach our responders to be more mindful about triaging first, to delegate more and sooner, and to resist management attempts to derail. Simply inviting them to assess their own incident response in this exercise lets them improve.

Questions