YYYY-MM: ${TEAM_NAME}

Purpose
All of a sudden, we have a big, distributed team. From the looks of our Team Schedule, it will be difficult to get synchronous time with everyone together at once.

In the earlier, Developer Experience-Platform iteration of this team, we operated with a high level of trust and collaboration. We should continue to do so. It can be difficult to trust/collaborate/communicate with someone that you don't know very well, and who you may not have many opportunities to connect with.

The goal of this document is to lay out a few communication standards for our new team, and to give folks' a place to define their personal preferences. I (Hannah) imagine that our processes and preferences will evolve over time as we get more comfortable working together. This document is a first step and a suggestion along the way.

When this document is complete, please be sure to read up on your teammates so that you can be a good per and partner in ~~crime~~ pairing.

People (please make sure you're familiar with everyone's write-up)

Firstname Lastname

Role
Engineer? PM?

Homebase
Where do you live?

Feedback preference
How do you like to receive feedback? This document has some suggestions for proper ways to give feedback further down in the file.

"Direct feedback" here means without sugar or without a compliment sandwich (“You’re REALLY good at X. You could use work with Y. But you’re still really good at XY!”).

Examples:

- Prefers immediate feedback; in a meeting if relevant. Prefers face-to-face; too easy to misinterpret text. Fine with public praise. Hearing the "why" is very important.
- Prefers private channels for feedback and praise. Slack is sometimes OK, but in-person is better in general.
- Prefers direct feedback, sooner than later. If it's something major or that needs a significant change, prefers async so they have some time to process, then with a follow-up sync conversation make sure we ended on the same page. Praise is fine in whatever way it’s packaged.
- Prefers private feedback if it's a personal issue but public feedback for more technical things; for the latter, our retro meetings are a great place to address that, but if tensions are high or if someone is taking things personally then a direct message is good.

Professional History Lite
A couple bullet points is fine here. Let folks know what you've done. It's interesting, and it gives us an idea of the things you might know.

If you've been at CircleCI for a long time, the saga of the various codebases you've touched is useful, too.

Areas of expertise

Services you know particularly well, things you could teach other teammates, areas you'd be delighted to help with and/or would decidedly not be delighted, but would be able to bail someone out.

OKRs
What are you trying to achieve this quarter? If you haven't made your OKRs yet, here are some ideas.

I hate having to explain to people that...
A pet peeve goes here, or a fact about pronouncing your name or about the city you grew up in.

Where does your name come from?
Tell the story.

Local work hours
Yes, we have a spreadsheet for specifics. This is for a gist, i.e. "I work 9-5 mon-fri."

My life outside of CircleCI... (optional)
Gasp! These exist???

They do. Please share as much or as little as you would like in this section.

A photo of where you work (optional)
${IM_A_PHOTO}

Team Process (you can skim this, it's good as a reference)

Where it comes from

Historically, as a team we've discussed process at a bi-weekly Friday morning meeting during Retro. Process is good until it doesn't work for us anymore, then we try something new. If something sounds beneficial, we'll try it out for the two weeks until the next retro.

Here are our retro notes.

Most recently, we've decided:

- To impose a WIP limit (number of folks on the team divided by two, rounded down)
  - JIRA turns red and screams when we're over 2 cards rn
  - If a card is blocked, right-click it and add a blocked flag. Move it back to Ready for Dev. It will show up in a separate swim lane.
- To skip daily stand-up in favor of channel updates
  - :status: what you're picking up
  - :spinning: some description of work you're stuck on, if you see this and are free, offer to help
  - :blocked: stuck. needs assistance.

Meetings

Historically, we've had two standing 1-hour meetings/week, and daily stand-up.

Planning Meeting

On **Mondays**, we meet first-thing to go over our Next Up column. We use Agile Poker to estimate cards before the meeting (an invite to update the estimates usually goes out before Friday), and discuss any disconnects.

Folks should leave this meeting with a clear picture of what they should be picking up throughout the week, and why.

DX-P Kanban Board, Agile Poker is visible in the upper right-hand corner:

Big Picture OR Retrospective

On **Fridays**, we alternate Big Picture and Retrospective meetings.

Big Picture -- Generally more Product-focused, where we're headed and why
Retrospective -- How's the team doing? What needs to change/stay the same/what other concerns do folks have?

Values

We've got an action item in the next retro to deliberately define these.

I would like for us to be able to collaborate and communicate in a high-trust environment where we're all striving to pragmatically ship the best solution we can think of given the constraints before us.

AKA we're not hecka fancy in our code, we build things we'd be comfortable maintaining, and we do so in an environment where we know that we can pass work to a teammate, and pick up work from a teammate, and that work will be carried to completion.

Assuming everyone's getting their work done, and surfacing when they have issues and need a hand, there's also always time for cheesy wordplay jokes and a bit of fun.

Many folks have small children so Things My Child Did are always an interesting source of amusement. I imagine Things My Roommate Did will add an excitingly diverse dimension to the team's shared humor.

Communication

Async First: Write it down

We have not perfected async communication. Whenever possible, write down what you're working on (e.g. take meeting notes) so that your teammates can review it.

Pay attention to your noise-to-signal ratio. While sometimes a painful overhead, this is a good skill to practice in working with remote teammates and in communicating with busy executives.

Examples:

- Yes, record meetings, but don't assume anyone will watch them
    - Take meeting notes
    - Indicate in the notes if a particular portion of the recording would be useful to view/at what time that portion begins
- Update your JIRA card at the end of every day
    - Assume that someone else will pick up your card
    - What information do they need in order to have a good hand-off experience?
        - Things you tried that were dead-ends
        - Things you think should be tried
        - Where your branch of work is
        - Anyone you spoke to that provided good context (and that context)

Slack @s

It is OK to write someone's name (even though it is a highlight word) or to @ them at any time in their day.

You can reasonably expect that they will respond within their working hours, and will ignore your pings outside of work hours.

Although some folks may have personal preferences here ("don't @ me in my free time"), that's a lot of cognitive load to keep track of on a regular basis. If you're not working, aren't on Pagerduty, and don't want to be pinged, set DND.

Giving Feedback

There are many different ways to give feedback (and praise!). In order to make sure the constructive criticism we deliver is actually constructive, keep these guidelines in mind:

1. Positive feedback can be offered in public or private. However, negative feedback can only be given in private.
2. Deliver feedback in a timely manner. It is best to give feedback as close as possible to the occurrence of the behavior or issue that requires correction.
3. Be clear, thoughtful, and respectful. Make sure the feedback you offer is accurate, specific, and is limited only to the behavior in question.

For a more extensive list of guidelines, check out ClearRock's list, here. For even more please let your manager know you'd like more guidance in delivering feedback.

Useful Links

1. Team Schedule
2. Big Picture Meeting Agenda
3. 2019 Retro Doc
4. Team folder
5. Team folder in Eng GitHub Repo

Q2 Goals and OKRs

If you're having trouble setting OKRs, here are some musings (originally shared in Slack) around areas to consider:

Technical

Musings: In terms of *goal-setting for this quarter,* many of us (including myself) are interested in getting a better understanding of system architecture. I personally am a huge fan of specific examples when learning, because I feel like a lot of the gist for architecture instruction in books roughly says: don't be dumb. I don't find that particularly helpful.

To focus on something more concrete, goals could be tied to:

1. There is a good chance we will pick up work in the VM service. A great deal of legwork has been done to scope that work already, but before we dive into it, we will need to make sure we agree with the approaches suggested. This is an opportunity to actively consider what architectural choices will work for us. We have a growing Platform Team now, how will their presence affect the design decisions that we make?

2. We have many microservices. You could spend time focusing on two or three services. Identify patterns that seem to work well, and ones that don't. Facilitate a conversation with our team about your findings (or in Eng Coffee Club) and see if suggestions for improvements or any concrete changes come out of them.

I would be happy to allocate some of our Tech Debt time to addressing compelling findings.

3. Pick an item that interests you from the Tiger Team "Key lessons" list. There is work getting scoped around these outcomes. Marc has more insight into it than I do, but certainly there will be room here to make opportunities for yourself to pick up pieces of the work and collaborate with folks on other teams. If there are more-senior folks you've been wanting to pair with that are already going to spend time on this work, reaching out and pairing seems like a good place to start. Make sure to ask questions about how they figured out these "Key Lessons," and to be thoughtful in designing and understanding more-sustainable solutions.

Communication

Other goals that will _always_ be useful for your career development and to the team are ones around improving communication and providing each other with constructive feedback. These are harder to generalize (we're all different and have different strengths and weaknesses), but could be like:

1. If someone does something that confuses you, or is difficult to follow in a technical conversation, commit to asking for clarification and telling them what manner of explanation worked for you.

2. Generate a metric around the number of follow-up questions folks ask you about cards you write, etc. Use it as a rough proxy for how well you're able to anticipate questions and clearly articulate problems in the first place. Ideally, your proxy would improve. Even if it doesn't, pay attention to the kinds of follow-up questions you get: do they follow a pattern?

3. Commit to maintaining these Slack & JIRA status updates, they're :ok_hand:

4. We have an intern joining. Flex goals: how well can you mentor asynchronously? That's like next-level challenging.