HARVARD | BUSINESS | SCHOOL

9-620-066
REV: FEBRUARY 13, 2023

PRITHWIRAJ CHOUDHURY
EMMA SALOMON

GitLab and the Future of All-Remote Work (A)

GitLab CEO Sid Sijbrandij grabbed his coffee and headed over to his workspace in his home office, ready to start his workday. He powered up his laptop and logged in, switching his background from a cluttered home office to a photograph from GitLab's last company offsite in New Orleans. Finally, Sid turned to say hello to Beamy, his office robot. Beamy slowly circled the office, transmitting video to the company and the world to prove that Sid was indeed sitting at home alone, not in a boardroom surrounded by other company leaders. It was time to start the day.

After lunch, Sid hopped on a call with GitLab investor Khosla Ventures. The firm had initially been wary of the all-remote model, where every employee from junior staff to CXOs worked outside a traditional colocated office space—indeed GitLab had no physical office space at all. However, over time, Khosla Ventures grew convinced that the model could work, at least in some cases. It remained less convinced about the model's ability to scale beyond the start-up phase though. Discussing GitLab's rapid growth, Sid reiterated his belief that, more than being simply an alternative to the traditional model, all-remote had an advantage, saying:

> I think all-remote scales even better that the traditional model. I see that a team in one room works really well. A team on one floor probably works OK. A team on multiple floors? You start having problems. A time in multiple offices? Getting harder. Multiple cities? Getting harder. Multiple continents? Super hard. That model breaks down, and there's very little benefit to colocation if you're such a big company. Meanwhile, the benefits of all-remote—writing down your processes, stimulating cross-company informal communication—they get much more pronounced at scale. So, it's not just that this model scales, it scales way better.[1]

He also highlighted how effective GitLab's executive team was, despite being geographically dispersed. Though Sid lived in the San Francisco Bay Area, his VP of Product Strategy Mark Pundsack lived in Illinois, his EVP of Product Scott Williamson lived in Boulder, CO, and his Chief Marketing Officer Todd Barr lived in Raleigh, NC. Even so, they coordinated regularly via Zoom and Slack, meeting in person only during their quarterly strategic planning meetings and during the whole-company offsite.

Professor Prithwiraj Choudhury and Research Associate Emma Salomon prepared this case. It was reviewed and approved before publication by a company designate. Funding for the development of this case was provided by Harvard Business School and not by the company. HBS cases are developed solely as the basis for class discussion. Cases are not intended to serve as endorsements, sources of primary data, or illustrations of effective or ineffective management.

Further, Sid argued, the all-remote model allowed GitLab to tap into talent pools that would be inaccessible with a single colocated office, or even with several spread across the globe. People with family obligations or dual-career couples—especially those with spouses whose jobs required regular moves, like the military or diplomats—could continue to work for GitLab regardless of their physical location. Said Sid, "I'm absolutely certain that the caliber of people we hire would be much harder to obtain if we had to relocate them."[2]

Hanging up with Khosla Ventures, Sid checked his calendar, and saw that he had three tough calls coming up. The first was with People Operations Manager Nadia Vatalidis. Nadia's team faced a unique scheduling challenge. With team members spread worldwide, coordination was a major challenge, even when they did their best to work asynchronously with one another. Something needed to change.

Next, Sid needed to speak with Senior Director of Corporate Marketing Melissa Smolensky. An unauthorized media appearance by a GitLab team member had spurred a flurry of negative online feedback, after the engineer revealed an internal debate regarding worldwide hiring practices. It seemed likely that a communication failure was at the root of the problem, but identifying what had gone wrong and why would be crucial to avoiding making a similar mistake in the future.

Finally, Sid had a call planned with GitLab's Head of Remote Darren Murph. For years, GitLab had held a Monday morning company check-in call, but as hiring ramped up, it had grown unwieldy. Team updates were down to a couple of sentences and personal updates were eliminated completely. This had made the call more efficient, but it was clearly no longer an effective way to engage employees and build a cohesive company community.

It was all well and good to evangelize the all-remote model to investors like Khosla Ventures, Sid thought. As far as he was concerned, the model had clear upsides. The real estate savings alone from having an all-remote workforce were at least $2,000 per worker annually,[3] with total cost savings to the business (from real estate savings, productivity increases, and reduced attrition) topping $18,000 per worker per year, according to some calculations.[4] Given that though, Sid wondered, why weren't all companies pursuing remote work? Was there something that major companies like IBM (which recalled its remote workers in 2017)[5] understood, but that he had missed?

Remote Work Models in the Broader Economy

In recent years, a number of high-profile organizations had begun offering their employees greater geographic and temporal flexibility. That flexibility has ranged from one work-from-home day per week, to full-time work from home, to total geographic flexibility within the continental United States, or even globally. Some firms had limited the types of positions that were allowed to work remotely. One such case was the U.S. Patent and Trademark Office, where only patent examiners had the option of remote work.[6] Another was American Express, which divided its workers into four tiers based on job function, and allowed each tier a different level of flexibility, from being fully colocated to being fully out-of-office.[7] Most established companies maintained a physical office space, even as they began offering their workers greater geographic flexibility. Akamai, for example, allowed many team members to work remotely even as it continued to expand to new physical locations.[8] In contrast, some newer firms—many of them in the technology sector, like GitLab—opted for no physical offices at all.[9]

Some major corporations that experimented with geographic flexibility ultimately rolled those programs back. Yahoo! recalled its remote workers in 2013, with CEO Marissa Mayer citing lost opportunities for collaboration among colocated coworkers.[10] IBM eliminated its work-from-anywhere

policies in 2017, citing similar concerns.[11] The federal government, which expanded its telework opportunities across a variety of departments under the Obama administration, halted several of those programs under the Trump administration, citing slow customer response rates and reduced productivity.[12] However, a recent academic study contradicted that claim, showing that workers granted geographic flexibility at the U.S. Patent and Trademark Office improved their productivity by 4% over workers participating in limited work-from-home programs.[13]

Companies that managed a hybrid model, with both colocated and remote staff, faced several challenges. Remote workers reported feeling left out of decision-making processes. Said one engineer, "I, as a remote worker, was basically excluded. Even when they tried their best to include me by doing a conference call… while they were waiting for me to connect, they would already have some of their discussion happening, so it always felt like I was going into a conversation that was already ongoing. You could never start from zero."[14] Remote workers also said that they were often not informed when priorities shifted or ideas changed, meaning they wasted time working on irrelevant tasks.

Remote workers also highlighted interunit friction as a challenge, particularly in settings where only certain teams were given the option of working remotely. Divisions between units also led to asymmetric buy-in to remote-friendly processes. A senior product manager explained, "Some departments of [my last] organization never bought in [to documenting everything]… and without that discipline [across the whole organization], you can't work effectively."[15] Asymmetric buy-in could also be a challenge among company leadership. As one remote worker noted, "In situations where there is one single headquarters that higher-profile executives commuted to, you, as a remote worker, definitely wonder if that lack of face time would impact you in some way."

GitLab's Head of Remote Darren Murph argued that many of these challenges were mitigated by having an all-remote, rather than a hybrid-remote, team. He explained, "We couldn't get the people in one room [to reach consensus], even if we wanted to, so by nature we had to do things differently. And it turns out that doing things differently works really well… which is fortunate, because we couldn't—practically—do it any other way."[16]

Becoming All-Remote

GitLab had remote workers throughout its history, but it also had three periods of time when the company attempted colocation, with team members working in a traditional, physical, office. The first such instance was in 2014, when CEO Sid Sijbrandij made his first two GitLab hires in the Netherlands.

Before those hires, GitLab's team was totally distributed. Dmitriy (DZ) Zaporozhets and Valery Sizov—who created GitLab in 2011—both lived in Ukraine. Sid came onboard in 2012 and began to develop GitLab.com, GitLab's first SaaS (software as a service) offering, but continued to live in his home in the Netherlands. An engineer based in Serbia also joined the team in this initial period.

The early GitLab team sought to develop a more effective collaboration tool for programmers seeking to write, troubleshoot, deploy, revise, and store code (for a detailed overview of the GitLab product, see **Exhibit 4**). The initial version of GitLab ran as an open-source tool, with users contributing to the underlying code, constantly iterating and improving on the product. In 2013, GitLab introduced a new product—GitLab Enterprise Edition—for larger businesses, and in 2014, the company incorporated and shifted its business model, leaving a no-cost core tool available, while marketing the Enterprise Edition as a paid service (the open-core development model).

In the lead-up to incorporation, Sid hired more staff in the Netherlands, bringing the total GitLab team to six employees. At that point, he decided that it made sense to carve out office space in his home for them to colocate. The initial three worked in that space for several days, but it quickly became clear that it was more efficient for the other team members to work from home, skipping their hours-long commute. Thus ended GitLab's first experiment with traditional colocated offices.

The second period of colocation was in 2015, when the entire GitLab team—at that point, nine people—traveled to California to participate in Y Combinator. During those several months, the team lived and worked in a shared space. However, once their time at Y Combinator concluded, the team members returned home, ending the company's second experience colocating.

After Y Combinator, GitLab raised $1.5 million in seed funding,[17] and began to expand its hiring. Advisors and investors applauded GitLab's remote model, but cautioned that while there was proven success for engineering teams working remotely, there was not yet evidence that other company functions—particularly marketing, sales, and support teams—would work effectively in a remote setting. With this advice in mind, Sid rented office space in the Bay Area, with the intention of colocating most of the team—the third and final time GitLab would attempt a traditional office model. However, once again Sid noted a similar pattern. New hires came into the office for their first few days, and then stopped, preferring to work from home or otherwise skip their commute.

As a portion of the GitLab workforce always worked remotely, even when there was a physical office space available, the company was already developing an operational model conducive to remote work. With this remote-friendly operational model in place, and after a few months of renting a largely empty office space, Sid explained, "We were so used to using our systems correctly, using Slack, using video calls, using Google Docs, using the GitLab tool [that] we decided, remote works. Almost nobody in the company actually wants to come to an office." In 2016, GitLab declined to renew its office lease, ending its third attempt at traditional colocation. The company became all-remote, with no offices anywhere in the world.

GitLab continued to grow year-on-year, rapidly expanding its customer base, contributing community, and staff. By year-end 2019, after five additional funding rounds (for details, see **Table 1**), GitLab grew into a company with an estimated valuation of $2.7 billion and over 1,000 employees worldwide (for a breakdown of GitLab employees by gender, department, and geographic location, see **Exhibits 1**, **2**, and **3**, respectively).

Operating Asynchronously

One of the key tenets of GitLab's remote-friendly operational model was operating as asynchronously as possible. This was key for a globally distributed team, as it allowed each team member to work largely independently, regardless of their time zone.

Asynchronous operations depended on empowering individual workers to manage their own schedules. While the executive team set company-level objectives and key results (OKRs) each quarter, how to meet those OKRs was left to the relevant teams. Teams set key performance indicators and assigned action items to specific individuals, leaving each worker to determine how best to complete those tasks. In the interest of transparency and accountability, all of these goals and tasks—as well as the progress of ongoing projects—were visible to the public via the GitLab *issue tracker* (for more on the GitLab product and company terminology, see **Exhibit 4**). The value of transparency extended throughout GitLab's online presence, and the company also made most meetings, documents, and even their salary calculator (which adjusted based on position, experience, and geographic location)

available to the public. Proprietary information, including personnel and HR matters, as well as the company's strategic planning meetings and financial documents, remained private.[18]

On an average day, GitLab team members started work by firing up their laptops and checking if they had any time-sensitive or pressing notifications (see **Exhibits 5** and **6** for examples of the GitLab tool and company Slack environment). They resolved as many of those *issues* as possible before turning to their ongoing work, whether writing code, researching potential clients, developing marketing materials, or managing personnel. After completing a task, GitLab team members tagged the relevant colleague to review their completed work in the GitLab *issue* or *merge request*. While waiting for the response, the team member then turned to their next task.

The GitLab workflow encouraged workers to break their tasks down to the *minimum viable change*. This meant that workers regularly submitted still-in-progress tasks for feedback or more input from collaborators. Where some companies expected you to finish a full draft of a web page, or an entire section of code, GitLab encouraged team members to do multiple iterations of any task, allowing more opportunities for team members (and the broader community) to contribute and improve the final product. As Head of Remote Darren Murph noted, this was a real change from the traditional office model, where people tended to seek out input from various coworkers or departments before creating an initial draft. At GitLab, in contrast, Darren explained, "You're empowered to put together [that work] based on your expertise… And then we'll open it up [to the whole company] for discussion."[19] The team member responsible for the task received the feedback, and then used it to iterate on the initial draft. Darren tied this process to the GitLab value of "blameless problem-solving," which he felt was crucial to the ability of team members to feel comfortable sharing less-than-perfect work.[20]

However, since GitLab's workflow relied heavily on workers operating independently, these moments of collaboration could create challenges. Given the all-remote setting, all communication and instruction had to be explicit to avoid confusion. As Director of Technical Evangelism Priyanka Sharma put it, GitLab employees "need to be extra good at saying, 'Okay, I'm done with my piece,' so the next person knows it's their turn to take over."[21] Team members also had to think more holistically about how work happened, considering not just their own portion of the process, but also future stages. Internal Strategy Consultant Emilie Schario explained, "I think one of the things you learn when you work asynchronously is that to communicate effectively, you have to be thinking two or three steps ahead, anticipating what comes next and how to address potential blockers at that stage."[22]

Thinking about future steps and timing was particularly important for teams spread across multiple time zones. Content Manager Suri Patel sat in Los Angeles, while two of her key collaborators sat in Boston and London. This affected how Suri planned her week, she explained, because "if I need a blog post to go out right away, I know I can't get it to the Editorial team on my Friday night; I need to send it by Wednesday, to give them enough time to work on it."[23]

Handovers also had the potential to disrupt operations if done incorrectly. Director of Engineering Marin Jankovski argued that working in a distributed team could be a double-edged sword, saying, "We have the luxury of working 24 hours, basically… You can do a handover and a colleague will continue your work and problems get resolved much quicker… But, if a handover is not done correctly, your colleague will need to figure out where to start [and] maybe they start from the wrong place [or] maybe they have to wait for [you] to come online again [to clarify something], which means they lost eight, nine hours."[24]

Other team members also noted the potential for input requests to get lost in the shuffle. Benefits Manager Brittany Rohde said, "The number one thing that can go wrong [with all-remote] is that you can't knock on someone's door and ask for a status update. If somebody hasn't been getting back to

you, you can ping them on Slack, but they still might not respond."[25] Suri Patel agreed, saying, "If I tag someone in an issue, sometimes I wait a week to get a response, and that can slow projects down."[26] Suri also noted that these communication challenges were compounded if team members did not consistently operate within the GitLab tool. She explained, "Some people don't work within GitLab, and rely on Slack or email, and [that makes it more] challenging to communicate with them. In order for this all-remote environment to work, for all of us to be effective, we all need to work within GitLab. Otherwise processes break down."[27]

Working Handbook-First

With team members spread across distant locations and time zones, clear documentation became another central tenet of GitLab's organizational model. Documentation meant that no matter where someone was located, all GitLab team members would have access to the same set of information.

For example, if a team member was unable to attend a given meeting, they could watch a recording of that meeting on GitLab's YouTube channel at a convenient time. Meeting organizers also posted agendas for every meeting ahead of time, linked to the relevant GitLab *issue*, to allow team members—including those unable to attend a meeting due to time differences—to read background information and post questions before the session. Following every meeting, organizers updated the agenda and the relevant GitLab *issue* to reflect any decisions made during the session. As one team leader explained, this documentation was necessary to keeping a distributed team on the same page: "[Everything] is written into the relevant issue, so that becomes our one source of truth for my team… If it's written somewhere else, it's like it doesn't exist."[28]

Documentation also extended to GitLab's operational processes. GitLab needed all team members to have a consistent source of information on company processes, regardless of when a question arose. That source of information was the GitLab handbook,[29] which GitLab published online and strived to update continuously (see **Exhibit 7** for the GitLab handbook Table of Contents). If a team member found that their query was not addressed in the handbook, they were tasked with reaching out to relevant colleagues and documenting their findings in a new handbook section—a process CEO Sid Sijbrandij called "working handbook-first."[30]

In this way, the handbook grew to thousands of pages of information, covering everything from how to create a *merge request* to how to file an expense report. Said one team member, "The handbook is a really great resource. Because so much is documented, you have this safety zone, where you can go look and find [an answer to] something, even if no one else is online."[31] Operating handbook-first also affected intra-company communication at GitLab. People Operations Manager Nadia Vatalidis explained, "The way I usually respond to a query is by linking, rather than just answering the question, and what that does is to empower people to read… Since things keep evolving and changing [it's better to] be empowered to search for things when you need them [rather than trying to memorize the whole handbook]."[32]

However, the sheer volume of information available through the handbook became unmanageable. GitLab team members grew frustrated with its inscrutability, with one complaining that even if an answer to a question might be there, "Who the hell knows where to find it?!"[33] In an effort to address this discoverability problem, GitLab hired a designated engineer tasked with managing the handbook, and created a set of instructions on how to effectively search the handbook. The instructions were sent out to the full company, and added to the handbook itself.

Another challenge with the handbook-first model was that information was sometimes missing or not up-to-date. This issue concerned Sid immensely, as he viewed the handbook as foundational to all company operations. "We have to have one source of truth to work effectively," Sid explained, or the remote model "can start to crumble."[34]

GitLab team members conceded that the company had thus far fallen short on its goal of constantly updating the handbook, to ensure that it always reflected current company practices. Director of Technical Evangelism Priyanka Sharma said that in her experience with the handbook, "I have definitely seen examples where a process has changed, but the handbook section wasn't updated. I think it happens because we are such a high growth company, and there is so much going on all the time. But it is definitely a real issue of quality control."[35]

However, Head of Remote Darren Murph noted that his biggest concern was not when a process had changed, but when a team member looked for guidance on an issue and there was none. Sometimes, it was because the issue was a new one, but more often, Darren explained, "It means that instead of solving a problem and documenting, someone took a shortcut. If they had paused to document, it would have enabled the entire company to see their thought process and weigh in,"[36] ensuring that information would be available the next time the question came up.

Building an All-Remote Company Culture

GitLab team members joined the company from a range of traditional colocated settings and remote work experiences. Many team members shared a common concern about shifting to an all-remote setting, namely the potential to feel isolated and disconnected from the broader company. Head of Technical Evangelism Priyanka Sharma shared her apprehension, saying:

> I was very nervous when I first was thinking of joining GitLab, because I am one of those people who was very social in the office, always at every hangout, every happy hour. So I thought, I'm going to be so lonely at home and I won't have that community feel. The perception is that you're going to be alone in a room… and you're never going to see people again, and you might not feel connected to a company. None of that has been true, and it has actually been really the opposite.[37]

To overcome this sense of isolation, and build a cohesive team, GitLab made a conscious effort to create structured opportunities for team building and informal communication. Explained Sid, "I know at Pixar they place the restroom centrally so people would bump into each other… but why depend on randomness for that? Why not step it up a notch and actually organize the informal communication?"[38] In addition to weekly calls where team members shared personal updates, employees also scheduled regular "coffee chats" with individual colleagues. Team members noted that Zoom video calls were key to facilitating these informal interactions, because they felt there was little difference between an in-person interaction and a video chat.

GitLab offsites also played a central role in creating this sense of community. Offsites were held annually, with the full GitLab team flying to places like New Orleans or Prague for a week of themed workshops and activities, focused on strategizing and team building rather than day-to-day company operations. As Marin explained, meeting your colleagues in person allowed you to build relationships that facilitated work throughout the year, whether because you better understood their communication style or simply credited them with having good intentions during a tense online exchange.[39]

Other GitLab team members noted that the all-remote setting actually allowed them to be more active in their local communities outside of work. With flexible schedules, they said they were able to participate in family activities or volunteer with local organizations, which reduced the sense of social disconnection. Flexible leave schedules also played a role in combatting isolation and burnout, team members said. Brittany said she took days off to align with her husband's work schedule,[40] while Darren said he often took a week off after a stressful deadline,[41] and Suri said that she often ditched work and headed to the beach if the weather looked nice.[42] Taking such breaks was encouraged by senior leadership, and Sid noted that he, himself, took several weeks off during the summer, to spend time with friends and travel.[43]

However, GitLab team members still acknowledged the potential for isolation when working in an all-remote setting, particularly among less social team members. If someone chose not to participate in informal activities, or if they were showing signs of burnout, it could take longer for a manager or colleague to notice due to the physical distance between them. As Suri explained, in an all-remote setting, "There is nobody to tell you to stop working. There is always a tremendous amount of work to do, and we are the masters of our own schedule. There's no one to say, 'Hey, it's 5 o'clock, everyone go home.' My computer is right here. I always have access to it, so it's easy to just open it up and start working, or forget the time."[44]

Managing Human Capital in the All-Remote Model

Sid looked over the documents related to each upcoming call. He worried that any one of these concerns could undermine GitLab's success if handled incorrectly. Plus, the fact that all three were emerging at the same time gave him pause. How would the company actually manage these challenges? Were GitLab's existing processes flexible enough to accommodate the company's needs as it grew to 1,000—and hopefully, one day, 10,000—employees, or would the entire operational model need to change?

Call 1: Collaborating across Time Zones

Nadia Vatalidis headed one of the key teams within GitLab's People Operations (i.e., human resources division). A new hire in Manila made the team of five truly globally distributed, adding to existing team members in Nairobi, Johannesburg, Raleigh, and Boulder. Though GitLab generally strived to work as asynchronously as possible, the team had historically done a weekly check-in video call. The call allowed for efficient team coordination, and helped create a sense of team unity. However, the time zone of the new team member meant that the current weekly call would occur at 11 pm for her (see **Exhibit 8** for team members' work schedules and current call time). Nadia told Sid that this was untenable, as having a flexible work schedule "certainly does not mean that people should always be awake and available outside their normal work hours."[45]

Sid agreed that working asynchronously had its limits, and that a video call was sometimes the most efficient path forward. Nadia was also right, Sid thought, that with such a wide range of time zones represented in this team, having a weekly check-in call might no longer be possible. Sid considered Nadia's scheduling concerns. Was an 11 pm call once per week truly an unreasonable ask? What about once per month? Could another time work better? Was it necessary to have the entire team on a single video call? Would it make more sense to move the new hire to a more Asia-Pacific-centric team?

Call 2: Working Handbook-First

Melissa Smolensky managed marketing outreach for GitLab. A podcast appearance by a GitLab engineer had set off a firestorm of negative online activity, after the engineer accidently disclosed an ongoing debate within GitLab about whether to hire Russian or Chinese engineers to work with the data of American customers.[46] GitLab had been shocked by the development, and Melissa had not even been aware that a team member was talking to the press.

After talking with the engineer in question, Melissa explained to Sid what had gone wrong. She said that the engineer received the invitation to appear on the podcast—which focused on technical matters, not corporate issues or hiring concerns—and turned to the GitLab handbook to determine who to clear it with. The engineer found extensive information about managing her social media presence, but no guidance on who in the marketing team was the right point person to clear an interview. Confused, the engineer reached out to the marketing team via GitLab issues and Slack, but received no response over several days. Assuming the lack of guidance left the decision up to her discretion, the engineer accepted the invitation. Melissa told Sid that the engineer apologized profusely for the lapse in judgment, as well as for the inadvertent disclosure. Even so, the marketing team now faced a flurry of additional work managing the public reaction.

Sid and Melissa were determined not to allow a similar series of events to occur again. They were working to develop a new handbook section on press engagement. However, Sid still had questions. Why wasn't there already a handbook section on this? Certainly, there had been instances of team members giving interviews in the past, yet when this team member had looked for guidance, there was none. The handbook-first mandate had clearly broken down in this instance. How could GitLab enforce its handbook-first model, both to ensure that key steps like this were added to company documentation and that subsequent actors followed the amended process?

Call 3: Facilitating Virtual Social Interactions

Darren Murph was the Head of Remote at GitLab, tasked with wrangling the challenges of operating a fully remote company at scale. GitLab's team more than doubled in size during the 2019 calendar year, from around 400 to over 1,000 employees, and Darren was concerned that some company processes needed to be adjusted to accommodate that growth. Darren worried that failing to facilitate social interactions at scale could negatively affect the company's sense of community. More than that, however, Darren worried that with the increased distance between frontline employees and the executive team, longtime employees could also be less effective is transferring norms to new hires. He explained:

> GitLab thrives on working very differently. When the company is small, it's pretty easy to get everybody on the same page, but as the team scales, you end up with more and more people that have spent their entire careers in typical, bureaucratic, colocated organizations, where, to get information, you schedule another meeting, etc. And no one loves [that way of doing things], but unless you're in the C-suite, you don't have the power to fundamentally change the culture and make it any different. So, what ends up happening is they show up to GitLab with this old, bad, habit—that they know is bad—but they don't really know how to operate any differently.[47]

In talking to team members, Darren noticed that many of them cited a direct interaction with Sid or another CXO that cemented their understanding of the GitLab way of doing things. As the company continued to grow, Darren emphasized, that direct interaction would likely become less common. Virtual face time, Darren argued, was a key part of the all-remote model, both in terms of fostering

community and in contributing to a sense of equity on the team. As another team member put it, "I don't think I would be approaching my CEO or CFO or whoever else by knocking on their office door, but the online space evens out the playing field for everyone and it is simple to send a message, even to an executive."[48]

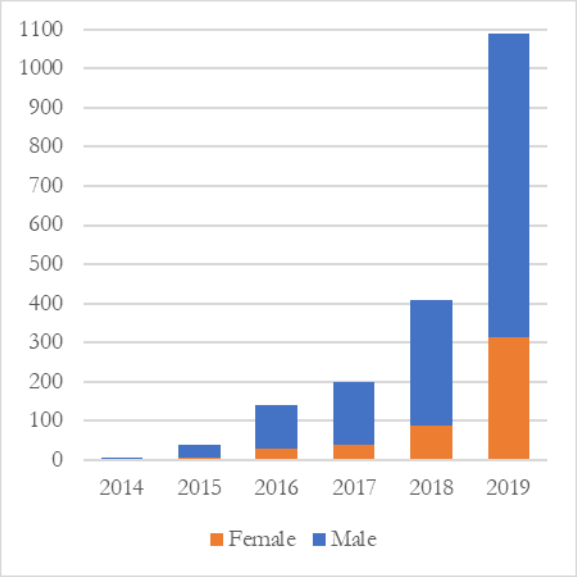
Darren highlighted the weekly company call as one vector for direct frontline-executive interactions in the early days. The informal portion of that call shifted from one unified call to a series of breakout groups, each with 20–50 team members. Even so, Darren worried that new team members were getting less and less interaction with company leadership. Sid agreed that now at 1,000 employees, the odds of him having a direct interaction with a new hire was far less than when GitLab had 50 or 100 team members (for examples of call distributions and time required at various employment levels, see **Exhibit 9**). Looking to the future, Sid realized something would have to change. But was it a question of how informal communications were scheduled? Could a different rotation, or scheduling scheme, increase frontline face time with CXOs? Or would informal communications need to undergo a more fundamental change, and if so, what would that look like?

Table 1 GitLab Funding Rounds

Series	Year	Value	Key Investors
Seed	2015	$1.5m	Khosla Ventures, 500 Startups, Crunchfund, Sound Ventures, Liquid 2 Ventures
A	2015	$4m	Khosla Ventures
B	2016	$20m	August Capital
C	2017	$20m	GV (formerly Google Ventures)
D	2018	$100m	Iconiq Capital, GV, Khosla Ventures
E	2019	$268m	Iconiq Capital, Goldman Sachs, Y Combinator Continuity Fund

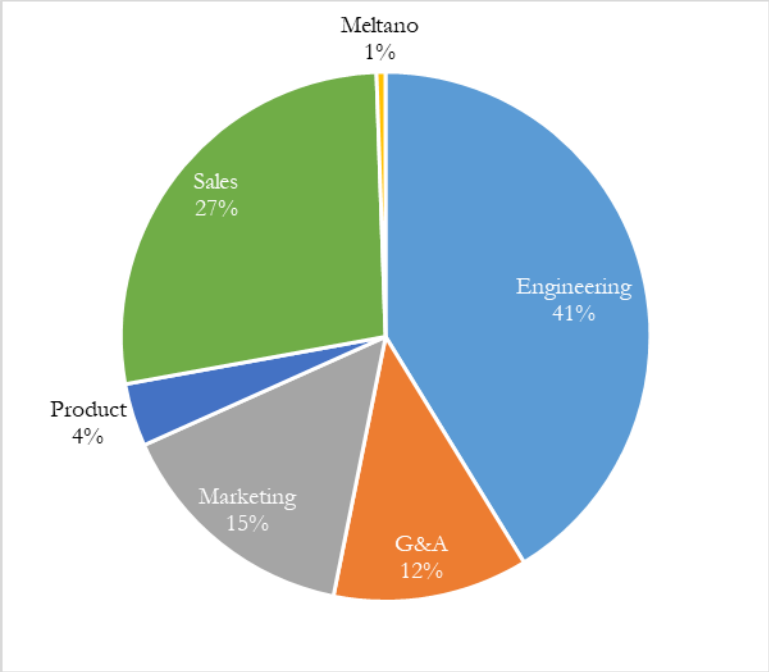
Source: Compiled from: Frederic Lardinois, "GitLab Raises $20M Series C Round Led by GV," TechCrunch, https://techcrunch.com/2017/10/09/gitlab-raises-20m-series-c-round-led-by-gv/, accessed November 2019; Frederic Lardinois, "GitLab Raises $4M Series A Round From Khosla Ventures For Its Open Source Collaboration Platform," TechCrunch, https://techcrunch.com/2015/09/17/gitlab-raises-4m-series-a-round-from-khosla-ventures-for-its-open-source-collaboration-platform/?_ga=2.15673590.1524112255.1574195054-809037987.1574195054, accessed November 2019; GitLab, "1.5M Raised in Seed Funding for GitLab to Accelerate Growth and Expand Operations," https://about.gitlab.com/blog/2015/07/09/1-5m-raised-in-seed-funding-for-gitlab-to-accelerate-growth-and-expand-operations/, accessed November 2019; Paul Sawers, "GitLab Raises $100 Million from Iconiq, GV, and Khosla, at $1.1 Billion Valuation, VentureBeat, https://venturebeat.com/2018/09/19/gitlab-raises-100-million-from-iconiq-gv-and-khosla-at-1-1-billion-valuation/, accessed November 2019; Paul Sawers, "GitLab Raises $268 Million at a $2.7 Billion Valuation," VentureBeat, https://venturebeat.com/2019/09/17/gitlab-raises-268-million-at-a-2-7-billion-valuation/, accessed November 2019; Ron Miller, "GitLab Secures $20 Million Series B," TechCrunch, https://techcrunch.com/2016/09/13/gitlab-secures-20-million-series-b/, accessed November 2019.

Exhibit 1 GitLab Employees since Incorporation by Gender



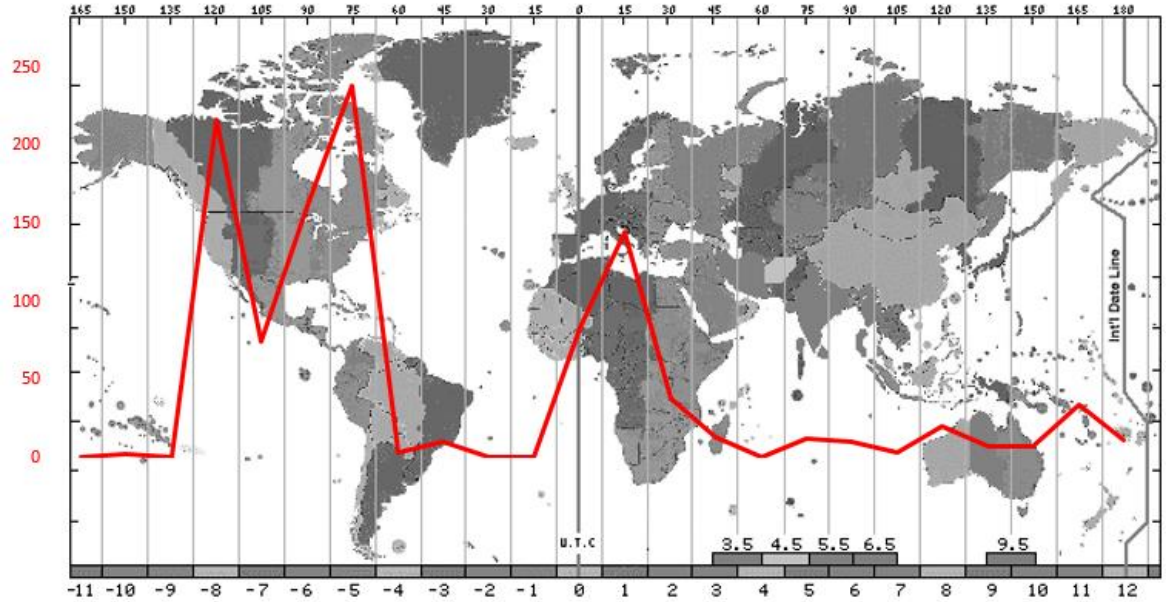
Source: Company documents; figures as of December 31, 2019.

Exhibit 2 GitLab Employees by Department



Source: Company documents; figures as of December 31, 2019.

Exhibit 3 GitLab Employees by Time Zone



Source: Employee information from company documents; Map source: NASA, "Time Zones in 2012," via Wikipedia Commons, https://commons.wikimedia.org/wiki/File:Time_Zones_(2012).gif, accessed January 2020.

Exhibit 4 GitLab Product and Terminology Overview

GitLab is a DevOps platform that allows users to collaborate on writing code, reviewing and troubleshooting it, and also to launch code and monitor its performance metrics. GitLab acts as a remote repository for Git code, and is a user-friendly way to engage with Git-based workflows. Notably, all the actions done within the GitLab product can also be done in a Git command line and then *pushed*, or transferred, to GitLab (a process that can work better for complex coding operations).

Within GitLab, users can create a project, or *milestone*, under which there are specific tasks, or *issues*, to be completed. Those *issues* can be assigned to different team members, and users can track the progress of work towards a given *milestone* and its due date.

An *issue* represents a task to be done, whereas a *merge request* represents a user's effort to complete that task. Within a *merge request*, one or more users can work to complete their task, editing code and reviewing each other's work. Once the content is finalized, users can *stage* and *commit* the changes within the *merge request*, moving the edits to the next stage of development.

At this stage, GitLab moves beyond the capacity of a generic Git repository, and creates a pipeline for preparing the proposed code change to be finalized and launched. GitLab runs the new code through a series of diagnostics, testing accuracy, security, and performance metrics before either approving the proposed change or identifying potential problems. The users working on the *merge request* can address any outstanding problems, and then move to merge the completed task into the master code file.

In order to prevent work from being merged prematurely, GitLab has a work-in-progress (WIP) label that locks the merge request in the collaboration phase. Once all users tagged in the task approve of the changes, the WIP label is removed. The new code can then be merged into the master code file and the *branch*, or parallel piece of code where edits were being made, can be resolved.

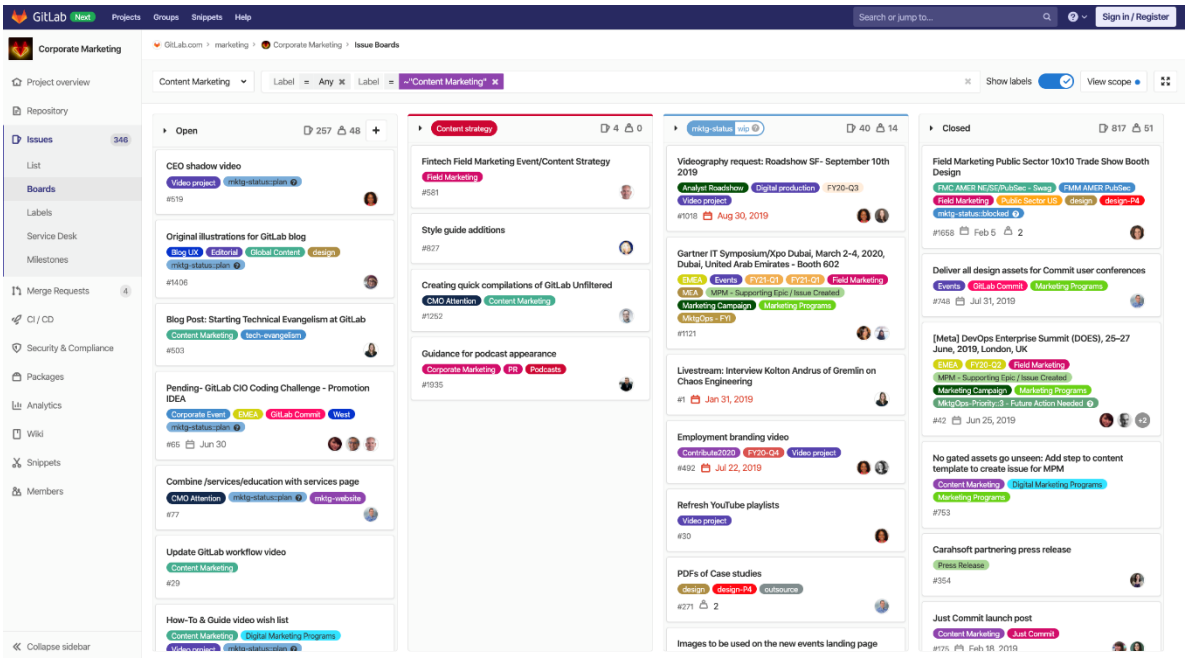
Once new code is launched, GitLab continues to monitor the performance and security of the end product. Meanwhile, the full history of the edits made, including all *commits* and resolved *issues*, remains accessible within the tool, allowing users to track changes over time and even to revert to previous versions if necessary.[a] For a short introduction to the GitLab product, see Video 1 below.

Video 1 GitLab Product Basics

https://www.youtube.com/watch?v=7q9Y1Cv-ib0

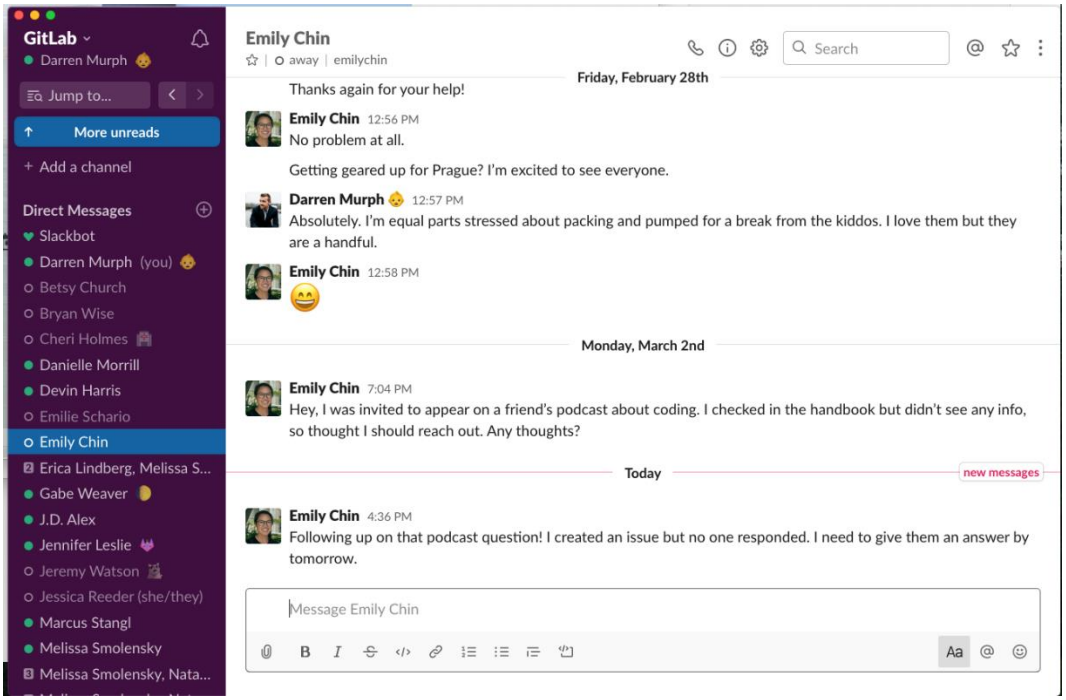
Source: Casewriter, based on "Introduction to GitLab Workflow," GitLab, https://www.youtube.com/watch?v=enMumwvLAug, accessed February 2023.

Exhibit 5 GitLab Environment



Source: Casewriters.

Exhibit 6 Slack Environment



Source: Casewriters.

Exhibit 7 GitLab Handbook: Stylized Table of Contents

1. **Company**

- Values
- Communication
- Culture
  - All-Remote
  - Life at GitLab
  - GitLab Contribute
  - Team
- Handbook
  - Usage
  - Edit This Website
  - Style Guide
  - Changelog
  - Roadmap
  - Searching the handbook like a pro
- CEO Readme
  - Shadow Program
  - Cadence
  - E-Group Offsite
  - KPIs
  - OKRs
  - Pricing Model
- Fellow of initial delight
  - Creator pairing
- About GitLab
  - History

2. **Product**

- Release posts
- Making Gifs
- Data analysis
- Markdown Guide
- Product Strategy
  - Corporate Development/ Acquisitions

3. **People Group**

- Code of Conduct
- Global Volunteering
- Hiring
  - Interviewing
  - Jobs FAQ
  - Vacancies
- Inclusion & Diversity
  - Ally Resources
  - FAQs
  - Unconscious Bias
- Labor and Employment Notices
- Leadership
- Learning & Dev
- On and Offboarding
- Spending company money
- Total Rewards
  - Benefits/PTO
  - Incentives
  - Compensation
  - Stock Options

4. **Engineering**

- Development
  - CI/CD
  - Defend Section
  - Enablement Section
  - Growth Section
  - Ops Section
  - Secure Section
- Infrastructure
- Quality
- Security
- Support
- UX

5. **Sales**

- Commercial
- Customer Success
- Reseller Channels
- Field Operations
- Reporting
- Technical Account Management
- Alliances

6. **Marketing**

- Website
- Blog
- Social Media Guidelines
- Revenue Marketing
  - Sales Development
  - Field Marketing
  - Digital Marketing
- Corporate Marketing
- Marketing Operations
- Community Relations
- Strategic marketing
  - Product marketing
  - Technical marketing
  - Partner marketing
  - Competitive intelligence
  - Market research
  - Analyst relations
  - Customer referrals
- Technical Evangelism
- Marketing Career Development

7. **Legal**

- Compliance

Source: "Handbook," GitLab, https://about.gitlab.com/handbook/, accessed January 2020.

Exhibit 8 Nadia's Team: Work Schedule and Weekly Call Time

Nadia GMT+2	Brittany GMT-5	Caroline GMT+3	Emily GMT-7	Verochka GMT+8
0:00	0:00	0:00	0:00	0:00
1:00	1:00	1:00	1:00	1:00
2:00	2:00	2:00	2:00	2:00
3:00	3:00	3:00	3:00	3:00
4:00	4:00	4:00	4:00	4:00
5:00	5:00	5:00	5:00	5:00
6:00	6:00	6:00	6:00	6:00
7:00	7:00	7:00	7:00	7:00
8:00	8:00	8:00	8:00	8:00
9:00	9:00	9:00	9:00	9:00
10:00	10:00	10:00	10:00	10:00
11:00	11:00	11:00	11:00	11:00
12:00	12:00	12:00	12:00	12:00
13:00	13:00	13:00	13:00	13:00
14:00	**14:00**	**14:00**	**14:00**	**14:00**
15:00	15:00	15:00	15:00	15:00
16:00	16:00	16:00	16:00	16:00
17:00	17:00	17:00	17:00	17:00
18:00	18:00	18:00	18:00	18:00
19:00	19:00	19:00	19:00	19:00
20:00	20:00	20:00	20:00	20:00
21:00	21:00	21:00	21:00	21:00
22:00	22:00	22:00	22:00	22:00
23:00	23:00	23:00	23:00	23:00

Source: Casewriters.

Note: Time shown is GMT. Cells with gray diagonal lines represent working hours (0900–1700); cells with a red grid pattern represent sleeping hours (2200–0600); the current weekly call time is 14:00 GMT and is shown in bold.

Exhibit 9 Virtual Socialization at Scale

Worker level	Total company employees			
	10	**100**	**1,000**	**10,000**
CXOs/Executive team	3	5	10	10
Mid-level management	-	15	90	990
Frontline employees	7	80	900	9,000

Source: Casewriters.

Company Size	Total call time for all workers to present (assumes 3 minutes per person)	Number of weekly calls		
		Randomly assigned group of 20 people	Randomly assigned group of 40 people	Randomly assigned group of 100 people
10	30	1	-	-
100	300	5	3	1
1,000	3,000	50	25	10
10,000	30,000	500	250	100

Source: Casewriters.

Endnotes

[1] Sid Sijbrandij, "Remote Work Benefits with Prof. Raj Choudhury & Sid Sijbrandij," interview by Raj Choudhury, September 26, 2019, https://www.youtube.com/watch?v=Rt57pb8KElI.

[2] Sid Sijbrandij in San Francisco, CA, interviewed by the casewriters in Boston, MA, via Zoom, November 21, 2019.

[3] Nicholas Bloom et al., "Does Working from Home Work? Evidence from a Chinese Experiment," *The Quarterly Journal of Economics* 130, no. 1 (2015): 170.

[4] Global Workplace Analytics, "Workplace Savings Calculator," https://globalworkplaceanalytics.com/roi, accessed February 2020.

[5] John Simons, "IBM, a Pioneer of Remote Work, Calls Workers Back to the Office," *Wall Street Journal*, May 18, 2017, https://www.wsj.com/articles/ibm-a-pioneer-of-remote-work-calls-workers-back-to-the-office-1495108802, accessed November 2019.

[6] Prithwiraj Choudhury, Cirrus Foroughi, and Barbara Larson, "Work from Anywhere or Co-Locate? Autonomy versus Learning Effects at the United States Patent Office," *Harvard Business School Working Paper* 19–054, 2019.

[7] Jeanne Meister, "Flexible Workspaces: Employee Perk Or Business Tool To Recruit Top Talent?" *Forbes*, April 4, 2013, https://www.forbes.com/sites/jeannemeister/2013/04/01/flexible-workspaces-another-workplace-perk-or-a-must-have-to-attract-top-talent/, accessed February 2020.

[8] Kathryn Mayer, "Benefits Professional of the Year Creates a Balance for Akamai Employees," *Employee Benefit News*, September 10, 2017, https://www.benefitnews.com/news/benefits-professional-of-the-year-creates-a-balance-for-akamai-employees, accessed November 2019.

[9] Emily Moore, "13 100% Remote Companies to Apply to Today," Glassdoor, June 22, 2018. https://www.glassdoor.com/blog/100-percent-remote-companies/, accessed February 2020.

[10] Kara Swisher, "Here's the Internal Yahoo No-Work-From-Home Memo," All Things D, February 22, 2013, http://allthingsd.com/20130222/physically-together-heres-the-internal-yahoo-no-work-from-home-memo-which-extends-beyond-remote-workers/, accessed November 2019.

[11] John Simons, "IBM, a Pioneer of Remote Work, Calls Workers Back to the Office," *Wall Street Journal*, May 18, 2017, https://www.wsj.com/articles/ibm-a-pioneer-of-remote-work-calls-workers-back-to-the-office-1495108802, accessed November 2019.

[12] Lisa Rein, "As Remote Work Rises at U.S. Companies, Trump Is Calling Federal Employees Back to the Office," *Washington Post*, January 12, 2020, https://www.washingtonpost.com/politics/as-remote-work-rises-at-us-companies-trump-is-calling-federal-employees-back-to-the-office/2020/01/12/37aad040-2d80-11ea-9b60-817cc18cf173_story.html, accessed January 2020.

[13] Choudhury, Foroughi, and Larson, "Work from Anywhere or Co-Locate?"

[14] Marin Jankovski in Amsterdam, the Netherlands, interviewed by the casewriters in Boston, MA, via Zoom, November 4, 2019.

[15] Gabe Weaver in Lookout Mountain, TN, interviewed by the casewriters in Boston, MA, via Zoom, November 21, 2019.

[16] Darren Murph, in Raleigh, NC, interviewed by the casewriters in Boston, MA, via Zoom, November 20, 2019.

[17] Jordan Novet, "Y Combinator-Backed GitHub Competitor GitLab Raises $1.5M," VentureBeat, July 9, 2015, https://venturebeat.com/2015/07/09/y-combinator-backed-github-competitor-gitlab-raises-1-5m/, accessed November 2019.

[18] "GitLab Values," GitLab, https://about.gitlab.com/handbook/values/#transparency, accessed February 2023; "The Compensation Calculator," GitLab, https://about.gitlab.com/handbook/total-rewards/compensation/compensation-calculator/, accessed February 2023.

[19] Darren Murph, in Raleigh, NC, interviewed by the casewriters in Boston, MA, via Zoom, November 20, 2019.

[20] For an overview of GitLab's corporate values, see GitLab, "GitLab Values," https://about.gitlab.com/handbook/values/, accessed November 2019.

[21] Priyanka Sharma in San Francisco, CA, interviewed by the casewriters in Boston, MA, via Zoom, November 22, 2019.

[22] Emilie Schario in Savannah, GA, interviewed by the casewriters in Boston, MA, via Zoom, November 26, 2019.

[23] Suri Patel in Orange County, CA, interviewed by the casewriters in Boston, MA, via Zoom, November 8, 2019.

[24] Marin Jankovski in Amsterdam, the Netherlands, interviewed by the casewriters in Boston, MA, via Zoom, November 4, 2019.

[25] Brittany Rohde in Clarksville, TN, interviewed by the casewriters in Boston, MA, via Zoom, November 6, 2019.

[26] Suri Patel in Orange County, CA, interviewed by the casewriters in Boston, MA, via Zoom, November 8, 2019.

[27] Suri Patel in Orange County, CA, interviewed by the casewriters in Boston, MA, via Zoom, November 8, 2019.

[28] Marin Jankovski in Amsterdam, the Netherlands, interviewed by the casewriters in Boston, MA, via Zoom, November 4, 2019.

[29] GitLab, "Handbook," https://about.gitlab.com/handbook/, accessed January 2020.

[30] Sid Sijbrandij in San Francisco, CA, interviewed by the casewriters in Boston, MA, via Zoom, November 21, 2019.

[31] Priyanka Sharma in San Francisco, CA, interviewed by the casewriters in Boston, MA, via Zoom, November 22, 2019.

[32] Nadia Vatalidis in Johannesburg, South Africa, interviewed by the casewriters in Boston, MA, via Zoom, November 18, 2019.

[33] Suri Patel in Orange County, CA, interviewed by the casewriters in Boston, MA, via Zoom, November 8, 2019.

[34] Sid Sijbrandij in San Francisco, CA, interviewed by the casewriters in Boston, MA, via Zoom, November 21, 2019.

[35] Priyanka Sharma in San Francisco, CA, interviewed by the casewriters in Boston, MA, via Zoom, November 22, 2019.

[36] Darren Murph, in Raleigh, NC, interviewed by the casewriters in Boston, MA, via Zoom, November 20, 2019.

[37] Priyanka Sharma in San Francisco, CA, interviewed by the casewriters in Boston, MA, via Zoom, November 22, 2019.

[38] Sid Sijbrandij, "Remote Work Benefits with Prof. Raj Choudhury & Sid Sijbrandij," interview by Raj Choudhury, September 26, 2019, https://www.youtube.com/watch?v=Rt57pb8KElI.

[39] Marin Jankovski in Amsterdam, the Netherlands, interviewed by the casewriters in Boston, MA, via Zoom, November 4, 2019.

[40] Brittany Rohde in Clarksville, TN, interviewed by the casewriters in Boston, MA, via Zoom, November 6, 2019.

[41] Darren Murph, "All-remote query for HBS case," email message to Emma Salomon, February 1, 2020.

[42] Suri Patel in Orange County, CA, interviewed by the casewriters in Boston, MA, via Zoom, November 8, 2019.

[43] Sid Sijbrandij in San Francisco, CA, interviewed by the casewriters in Boston, MA, via Zoom, November 21, 2019.

[44] Suri Patel in Orange County, CA, interviewed by the casewriters in Boston, MA, via Zoom, November 8, 2019.

[45] Nadia Vatalidis in Johannesburg, South Africa, interviewed by the casewriters in Boston, MA, via Zoom, November 18, 2019.

[46] Kristin Broughton, "Resignation at GitLab Highlights Concerns Over Corporate Espionage," *Wall Street Journal*, November 13, 2019, https://www.wsj.com/articles/resignation-at-gitlab-highlights-concerns-over-corporate-espionage-11573686134, accessed December 2019.

[47] Darren Murph, in Raleigh, NC, interviewed by the casewriters in Boston, MA, via Zoom, November 20, 2019.

[48] Marin Jankovski in Amsterdam, the Netherlands, interviewed by the casewriters in Boston, MA, via Zoom, November 4, 2019.