How GitLab Makes Agile Shine With Developers Spread Across 65+ Countries

Wayne Haber: Director of Engineering @ GitLab
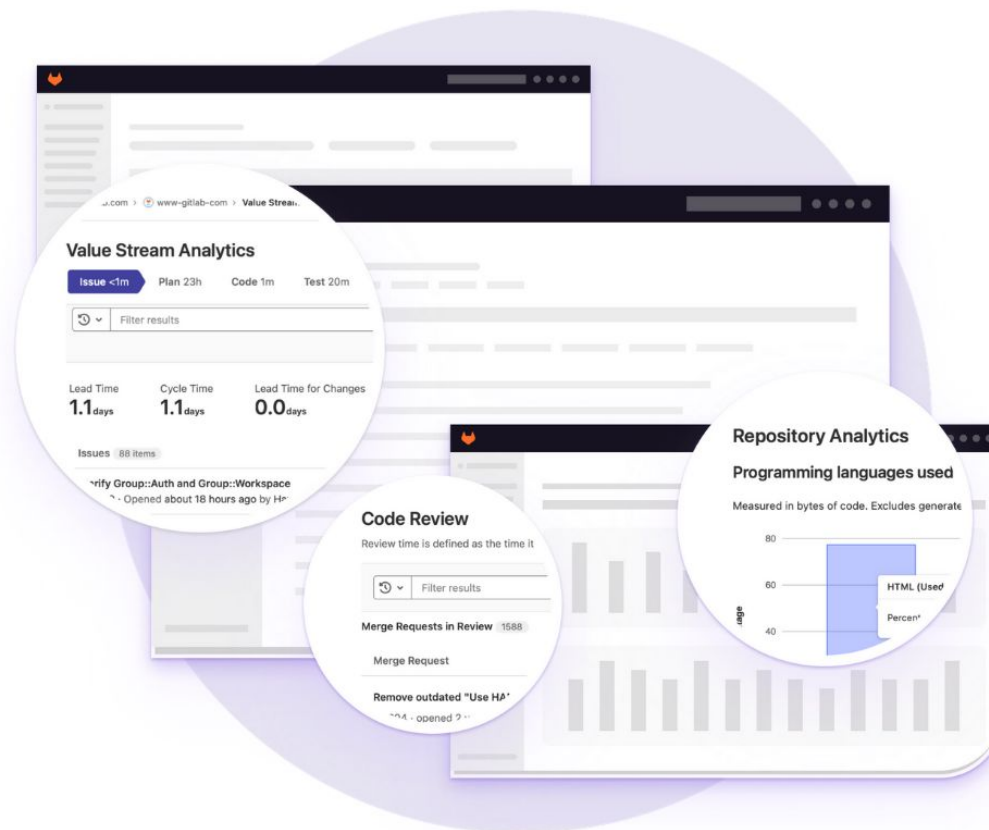
May 5, 2022

The One DevOps Platform

From planning to production, bring teams together in one application. Ship secure code faster, deploy to any cloud, and drive business results.

https://about.gitlab.com/

Who and Where is GitLab?

- 1,700+ employees in 65+ countries/regions.
  https://about.gitlab.com/company/team/

- All remote since 2015. Zero offices.

- Open source product that gets thousands of contributions every year from close to a thousand different people in the community.
  https://about.gitlab.com/community/contribute/

- Product is delivered via SaaS and as software users can run themselves

- Free versions and also paid versions are available for added features

- Not only is the product based on open-source, but our company handbook is also largely public: https://about.gitlab.com/handbook/

🤝 Collaboration

📈 Results

⏱️ Efficiency

🌐 Diversity, Inclusion & Belonging

👣 Iteration

👁️ Transparency

https://about.gitlab.com/handbook/values/

Product Development Flow

The product development workflow is used to manage the process from idea inception to production release.

https://about.gitlab.com/handbook/product-development-flow/

Lifecycle of a code change
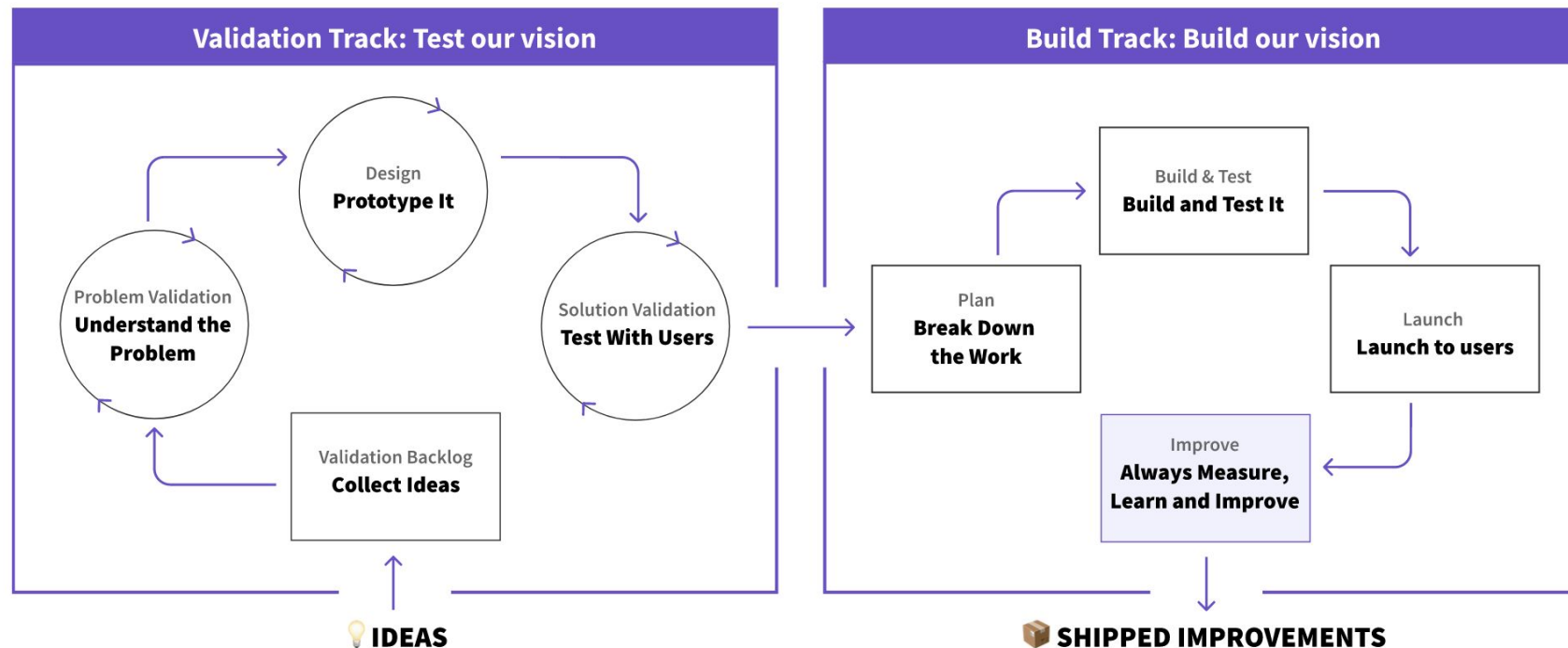
1. Developers work with their managers to choose items from the prioritized backlog
2. Development is often broken up into separate frontend and backend tasks which are tracked in issues. Some developers work on both types and some specialize in one type.
3. Developers create merge requests for their code changes
4. New functionality is often controlled by a feature flag (so that it can be quickly turned on / off based on success)
5. Once the merge requests are code complete and automated test cases are succeeding, the merge requests are submitted for the review process (https://docs.gitlab.com/ee/development/code_review.html)
6. The code is reviewed by reviewers, often on the same team as the author.
7. After the changes are reviewer approved, they are reviewed by the maintainers (with specific focus such as frontend, backend, database, and security based on type of change).
8. After the changes are approved by the maintainers, the code is deployed to staging and then to production.
9. All the changes for each release are compiled and released monthly for self-hosted customers.

Live demo using examples from
https://about.gitlab.com/handbook/engineering/development/sec/secure/threat-insights

Development processes

- Development teams have one manager and <=8 developers
- They have stable counterparts in:
  - Product Management
  - UX
  - Quality
- They are not located in the same time zone
- They primarily communicate asynchronously (in writing in issues, merge requests, and slack) vs in meetings
- They do daily standups in slack
- They do monthly planning before releases and monthly retrospectives after planning
- They do weekly meetings that have an agenda, are recorded, and alternate times so that everyone has an opportunity to attend no matter what time zone they are in
- Each engineering manager has a weekly 1:1 with each direct report
- Engineering teams have a consistent set of metrics include:
  - Team merge request rate
  - Work breakout by type (features, maintenance, bugs)
  - Past due infrastructure and security issues
  - Error budgets
  - 360 reviews
  - Culture surveys
  - OKRs

Live demo using examples from https://about.gitlab.com/handbook/engineering/development/sec/secure/threat-insights

Incentivizing changes from "others"

We incentivize changes across internal teams and the public via our concept of "everyone can contribute" :
https://about.gitlab.com/company/mission/#everyone-can-contribute

Why do they do this?
- **Teams don't need to wait on other teams** to change "the other team's code" for their team to succeed.
- **If someone see something they don't like, they can put in an MR** (merge request) to change it.
- **Free and paid users can improve the product** and the company handbook.
- **Celebration of open-source contributors** via the GitLab Heroes program:
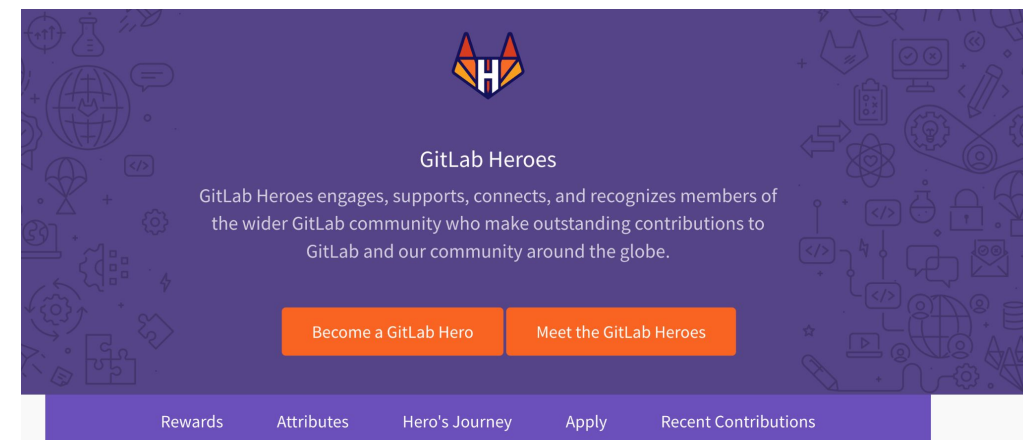https://about.gitlab.com/community/heroes/

Everyone can contribute

Everyone can contribute to digital products with GitLab, to GitLab itself, and to our organization. There are three ways you can Contribute,

1. Everyone can contribute with GitLab
2. Everyone can contribute to GitLab, the application
3. Everyone can contribute to GitLab, the company

Everyone can contribute with GitLab 🔗

To ensure that **everyone can contribute with GitLab** we allow anyone to create a proposal, at any time, without setup, and with confidence. Let's analyze that sentence a bit.

- Anyone: Every person in the world should be able to afford great DevOps software. GitLab.com has free private repos and CI runners and GitLab CE is free as in speech and as in beer. But open source is more than a license, that is why we are a good steward of GitLab CE and keep both GitLab CE and EE open to inspection, modifications, enhancements, and suggestions.

GitLab Heroes
GitLab Heroes engages, supports, connects, and recognizes members of the wider GitLab community who make outstanding contributions to GitLab and our community around the globe.

[Become a GitLab Hero] [Meet the GitLab Heroes]

Rewards   Attributes   Hero's Journey   Apply   Recent Contributions

Be a Hero

At GitLab, we believe everyone can contribute.
In that spirit, we believe everyone in our

Other interesting resources

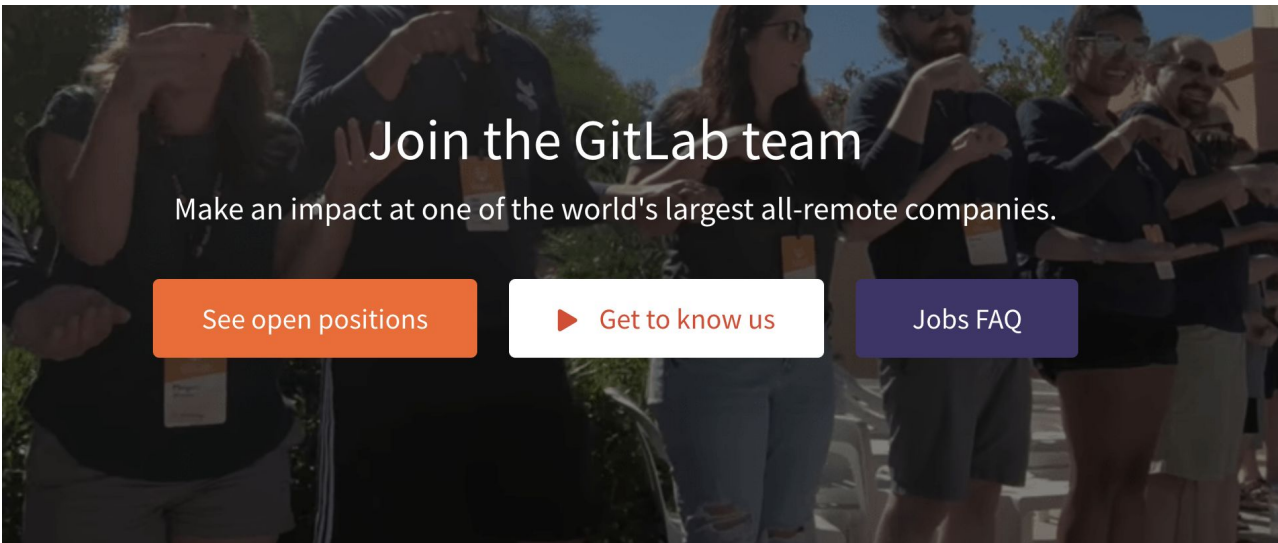## GitLab guide to all-remote:

https://about.gitlab.com/company/culture/all-remote/guide/

The Remote Manifesto

All-remote work promotes:

1. Hiring and working from all over the world (instead of from a central location).

2. Flexible working hours (over set working hours).

3. Writing down and recording knowledge (over verbal explanations).

4. Written processes (over on-the-job training).

5. Public sharing of information (over need-to-know access).

6. Opening up documents for editing by anyone (over top-down control of documents).

7. Asynchronous communication (over synchronous communication).

8. The results of work (over the hours put in).

9. Formal communication channels (over informal communication channels).

Join the GitLab team

Make an impact at one of the world's largest all-remote companies.

See open positions ▶ Get to know us   Jobs FAQ

We are hiring!

https://about.gitlab.com/jobs/

More about and how to contact Wayne:

https://about.gitlab.com/handbook/engineering/readmes/wayne-haber/

https://www.waynehaber.com

https://twitter.com/WayneHaber

https://www.linkedin.com/in/waynehaber/