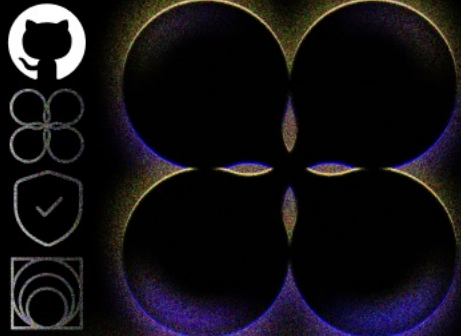


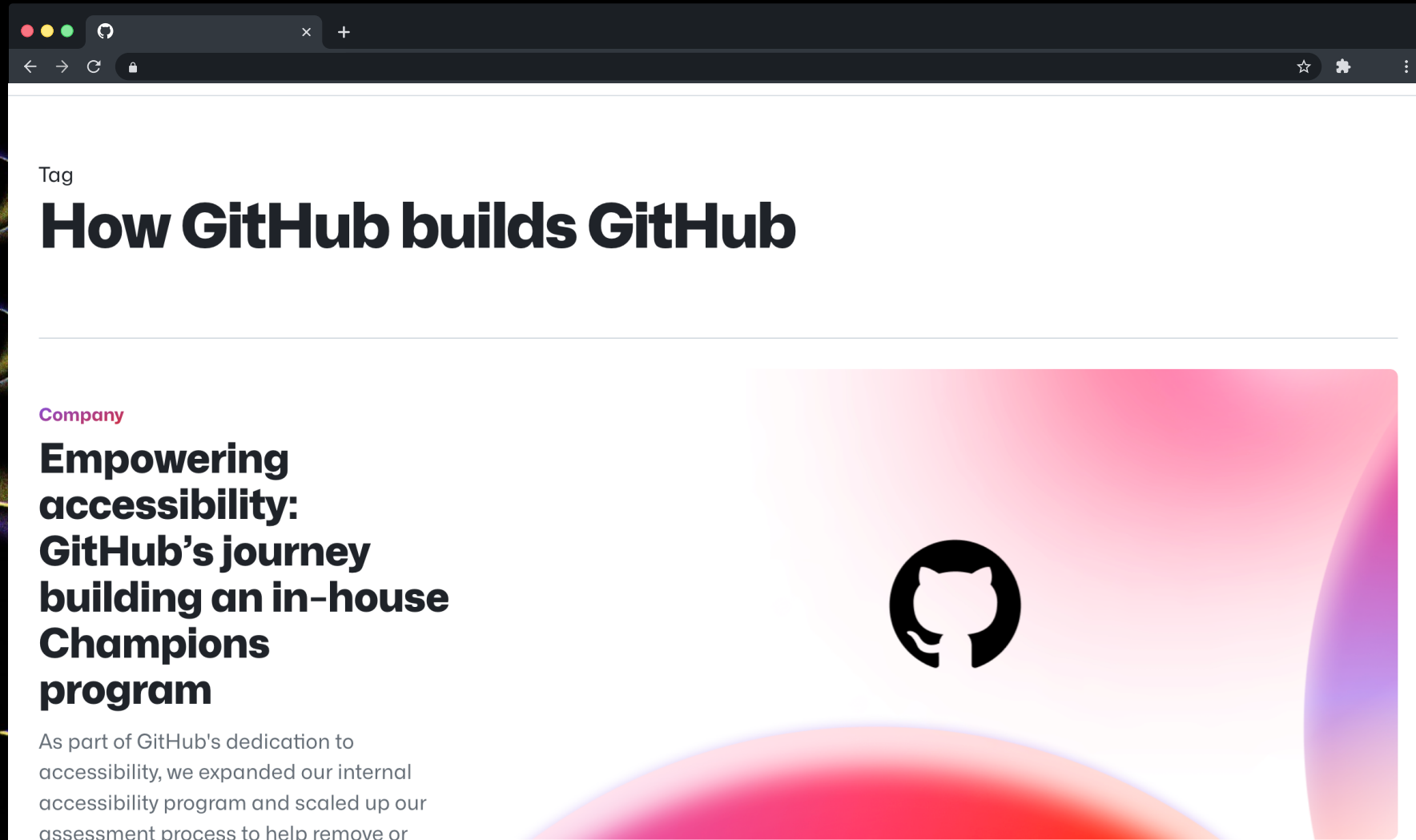
# How GitHub is using GitHub



Kenta Nakase / @parroty  
Senior Solutions Engineer, Corporate

# How GitHub builds GitHub

(<https://github.blog/tag/how-github-builds-github/>)




Tag

## How GitHub builds GitHub

**Company**

### Empowering accessibility: GitHub's journey building an in-house Champions program

As part of GitHub's dedication to accessibility, we expanded our internal accessibility program and scaled up our assessment process to help remove or



# Topics



## Collaboration

- コミュニケーション
- プロジェクト管理



## Productivity

- 開発者体験
- 自動化 (インフラ・ビルド)



## Security

- セキュリティ  
オペレーション



# Collaboration



# コミュニケーション



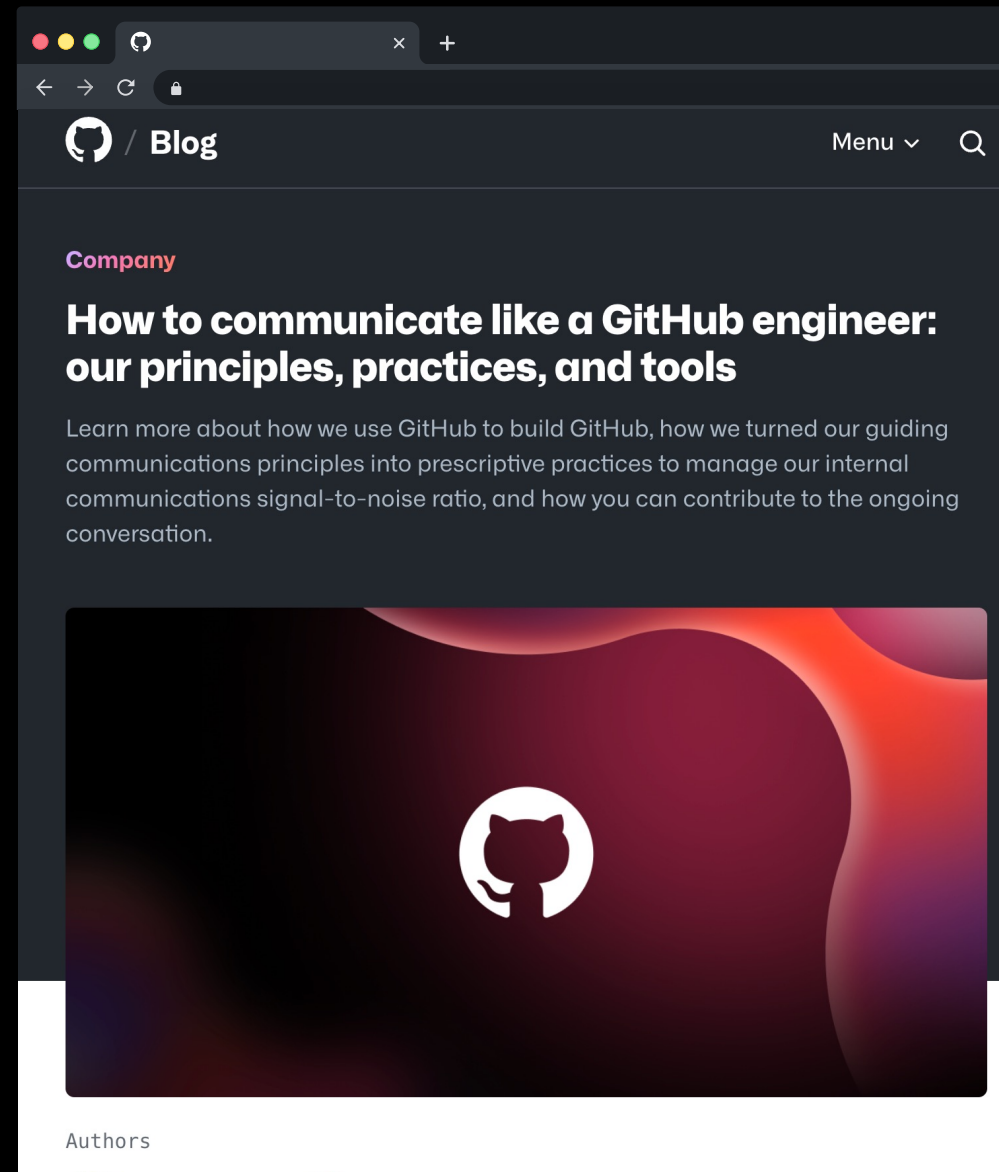
## 文書化・非同期

- あらゆる情報を書き起こす  
(各チームのリポジトリ)
- ソースコードのレビュー  
だけではなく、計画・議論  
でもGitHubを積極活用

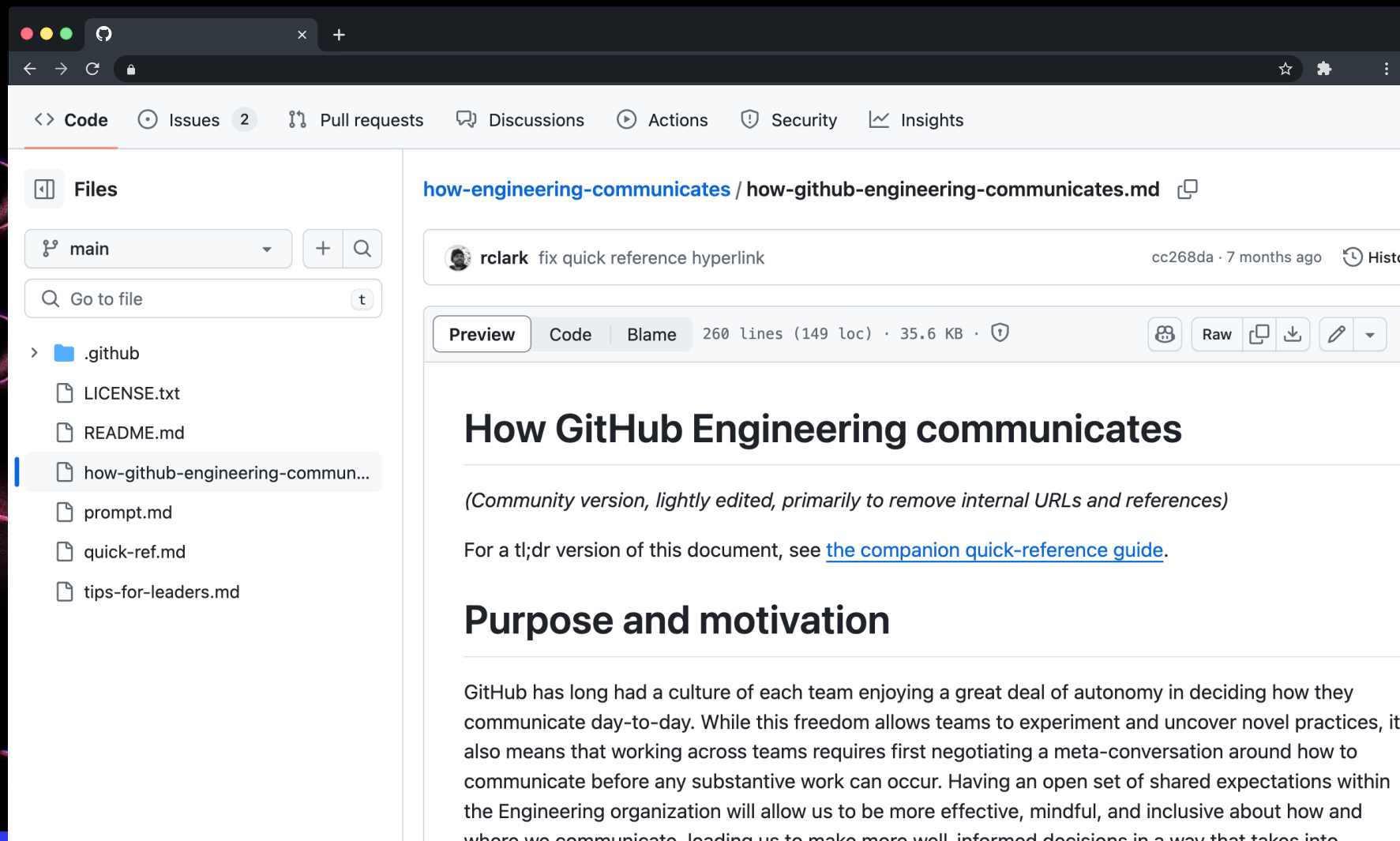


## 情報の集約・展開

- URLでアクセス可能にして  
共有 (@Mention機能)
- Chatへの自動展開、GitHub  
Actionsによる情報集約



# ガイドラインドキュメント (<https://github.com/github/how-engineering-communicates>)



The screenshot shows a GitHub repository page for the repository `how-engineering-communicates`. The file `how-github-engineering-communicates.md` is selected, showing a commit by `rclark` with the message "fix quick reference hyperlink" from 7 months ago. The document content is displayed in a preview view, featuring a title "How GitHub Engineering communicates" and a subtitle "(Community version, lightly edited, primarily to remove internal URLs and references)". The main text begins with "For a tl;dr version of this document, see [the companion quick-reference guide](#)." and a section titled "Purpose and motivation" which discusses GitHub's culture of team autonomy and the need for a meta-conversation on communication.

Code Issues 2 Pull requests Discussions Actions Security Insights

Files

main

Go to file

- .github
  - LICENSE.txt
  - README.md
  - how-github-engineering-commun...
  - prompt.md
  - quick-ref.md
  - tips-for-leaders.md

how-engineering-communicates / how-github-engineering-communicates.md

rclark fix quick reference hyperlink cc268da · 7 months ago

Preview Code Blame 260 lines (149 loc) · 35.6 KB

## How GitHub Engineering communicates

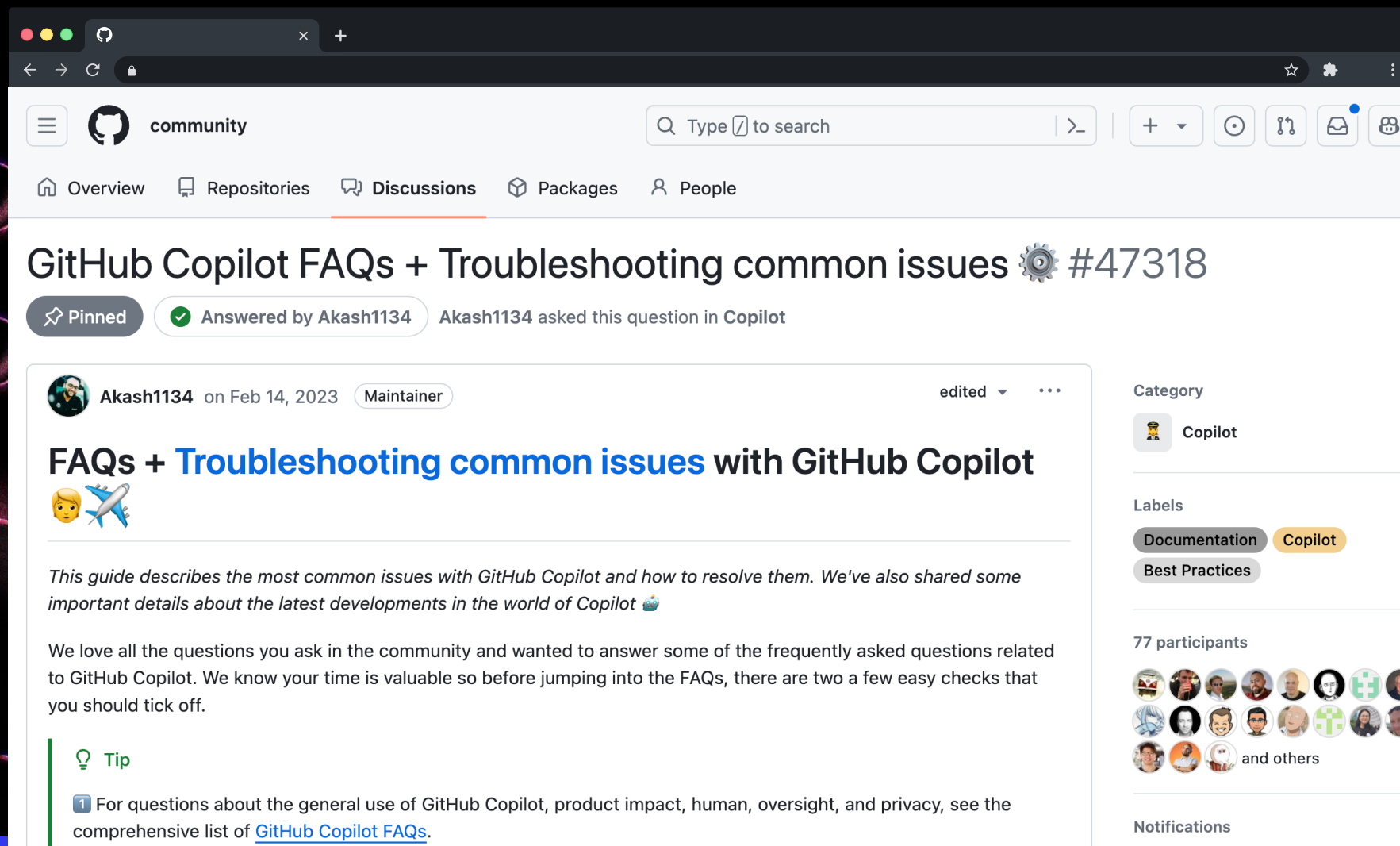
*(Community version, lightly edited, primarily to remove internal URLs and references)*

For a tl;dr version of this document, see [the companion quick-reference guide](#).

### Purpose and motivation

GitHub has long had a culture of each team enjoying a great deal of autonomy in deciding how they communicate day-to-day. While this freedom allows teams to experiment and uncover novel practices, it also means that working across teams requires first negotiating a meta-conversation around how to communicate before any substantive work can occur. Having an open set of shared expectations within the Engineering organization will allow us to be more effective, mindful, and inclusive about how and where we communicate, leading us to make more well-informed decisions in a way that takes into

# GitHub上での情報共有・議論（例：Community Forum） (<https://github.com/orgs/community/discussions>)



The screenshot shows a web browser window displaying a GitHub Community Forum discussion. The browser's address bar shows the URL <https://github.com/orgs/community/discussions>. The page header includes the GitHub logo, the word "community", a search bar, and navigation links for Overview, Repositories, Discussions, Packages, and People. The main content area features a discussion titled "GitHub Copilot FAQs + Troubleshooting common issues" with a gear icon and the ID #47318. The discussion is pinned and has been answered by user Akash1134, who is also the maintainer. The discussion content includes a title "FAQs + Troubleshooting common issues with GitHub Copilot" with a pilot emoji, a paragraph describing the guide's purpose, and a tip section. The right sidebar shows the category "Copilot", labels "Documentation", "Copilot", and "Best Practices", and a list of 77 participants.

community

Overview Repositories Discussions Packages People

## GitHub Copilot FAQs + Troubleshooting common issues #47318

Pinned Answered by Akash1134 Akash1134 asked this question in Copilot

Akash1134 on Feb 14, 2023 Maintainer

### FAQs + Troubleshooting common issues with GitHub Copilot

This guide describes the most common issues with GitHub Copilot and how to resolve them. We've also shared some important details about the latest developments in the world of Copilot

We love all the questions you ask in the community and wanted to answer some of the frequently asked questions related to GitHub Copilot. We know your time is valuable so before jumping into the FAQs, there are two a few easy checks that you should tick off.

**Tip**

For questions about the general use of GitHub Copilot, product impact, human, oversight, and privacy, see the comprehensive list of [GitHub Copilot FAQs](#).

Category: Copilot

Labels: Documentation, Copilot, Best Practices

77 participants

Notifications



# プロジェクト管理



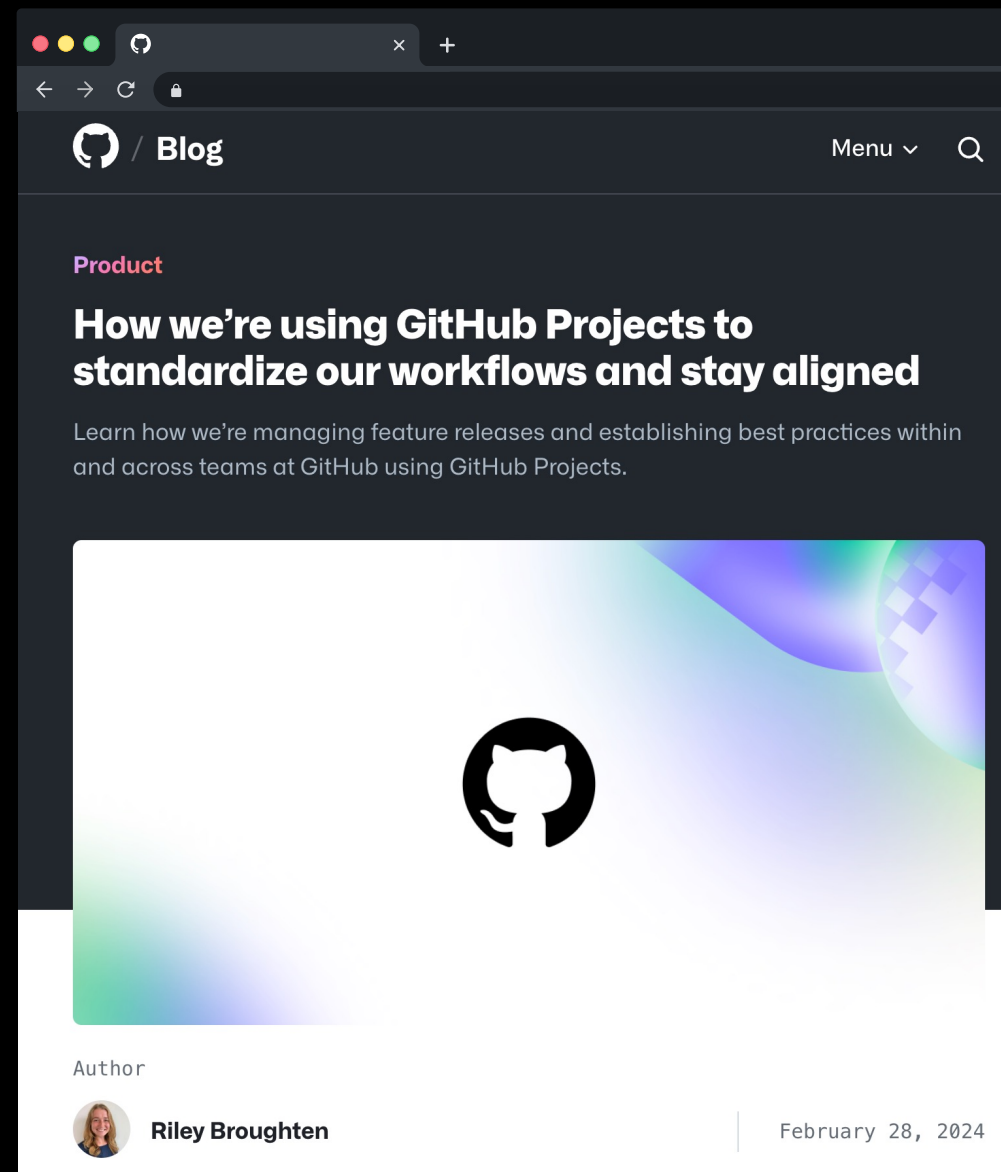
## 共通化とカスタマイズ

- 用途ごとのテンプレート  
(機能リリース, ロードマップ,  
オンボーディング, etc.)
- ロールに応じた異なる  
ビュー (マネージャ向け,  
担当者向け, etc.)



## 継続的な更新・状態管理

- 情報の定期的な更新・可視化  
(Status Update機能)
- Built-in WorkflowやGitHub  
Actionsでの更新自動化



# Product Feature Release Board

(カテゴリ分け、複数の異なるビュー)

Product Feature Release (Private template)

Use this template when planning out and releasing a new feature

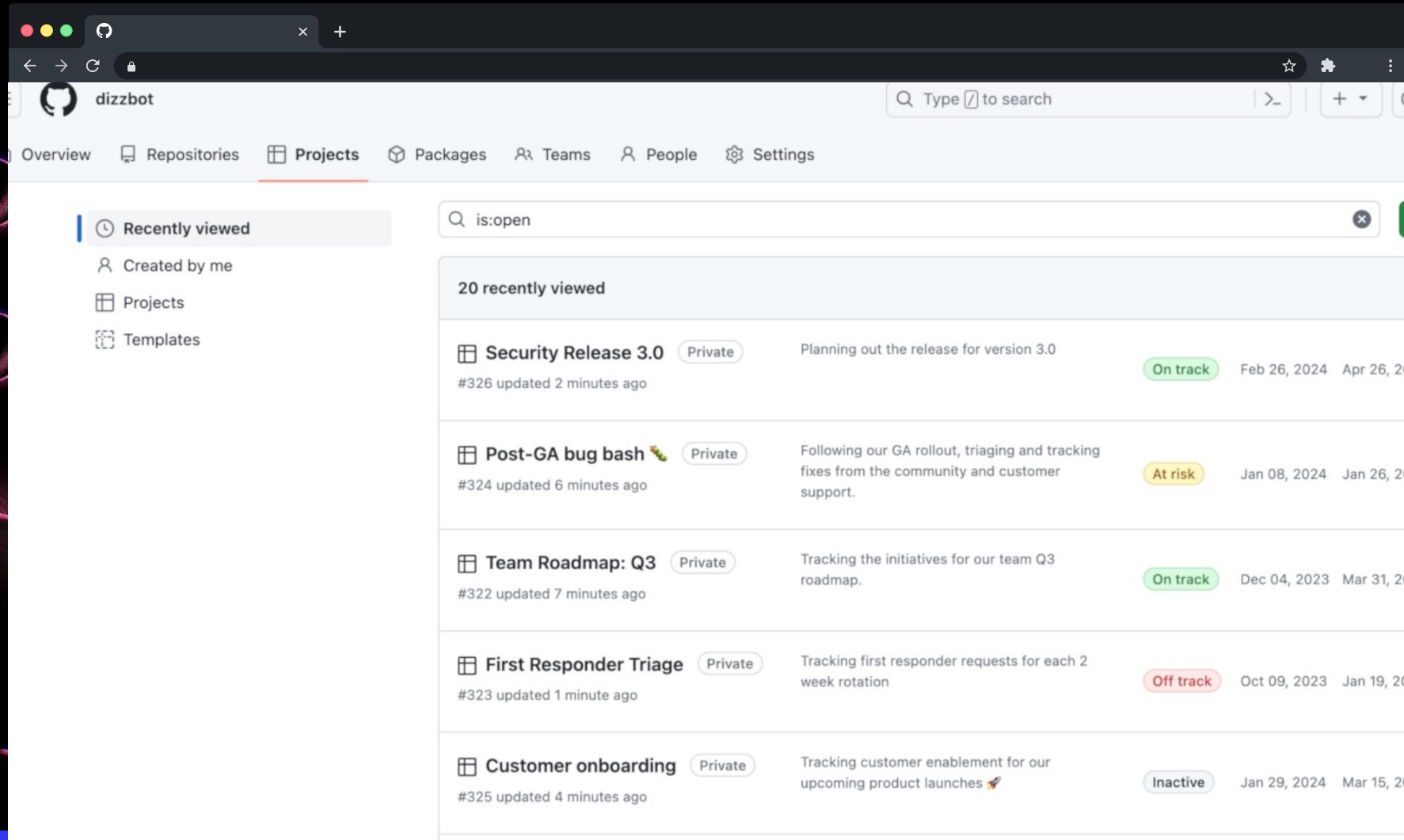
Prioritized backlog | Status board | Roadmap | Bugs | In review | My items | + New view

Filter by keyword or by field

Title	Status	Size	Assignees	Area
P0 (Estimate: 7)				
1 Initial user research	To do	L		Product
2 Document requirements and scope	To do	M		Product
3 Schedule kickoff call with engineering team	To do	XS		Engineering
4 Create release issue to inform stakeholders	To do	XS		Marketing
5 Break down scope into issues and batches of work	To do	L		Engineering
+ Add item				
P1 (Estimate: 10)				
6 Create plan for public documentation	To do	S		Docs
7 Determine timing for each release phase (Alpha, Beta, GA)	To do	M		Product
8 Plan out marketing strategy and content	To do	XL		Marketing
9 Gather early user feedback on designs and functionality	On hold	M		Design

# Product Feature Release Board

(継続的なStatus Update)



The screenshot shows the dizzbot Product Feature Release Board interface. The top navigation bar includes Overview, Repositories, Projects (selected), Packages, Teams, People, and Settings. A search bar is present with the text "is:open". The main content area displays a list of 20 recently viewed projects, each with a title, a "Private" label, a description, a status indicator, and dates.

Project Name	Status	Start Date	End Date
Security Release 3.0	On track	Feb 26, 2024	Apr 26, 2024
Post-GA bug bash	At risk	Jan 08, 2024	Jan 26, 2024
Team Roadmap: Q3	On track	Dec 04, 2023	Mar 31, 2024
First Responder Triage	Off track	Oct 09, 2023	Jan 19, 2024
Customer onboarding	Inactive	Jan 29, 2024	Mar 15, 2024



# GitHub Public Roadmap

The screenshot displays the GitHub public roadmap interface. At the top, there's a navigation bar with tabs for 'Board', 'Table', 'Generally available', and 'Cloud (github.com)'. A search bar contains the query 'is:open -status:"Q4 2022 - Oct-Dec","Q2 2023 - Apr-Jun","Q3 20' with 94 results. The main content is divided into two columns for 'Q1 2024 - Jan-Mar' (7 items) and 'Q2 2024 - Apr-Jun' (31 items). A right-hand sidebar shows a detailed view of 'Repository Properties #799' for 'github-product-roadmap', which was opened on Jul 11, 2023. The repository is marked as 'Open' and includes a description of 'Custom Properties' and a list of features.

**Q1 2024 - Jan-Mar 7**

- roadmap #799  
Repository Properties  
ga GHE 3.13 github enterprise  
repositories shipped
- roadmap #830  
Projects: Project status updates  
all cloud ga GHE 3.13 projects  
shipped
- roadmap #854  
Code scanning default setup can be enabled on repositories before code in supported languages are added to the repositories

**Q2 2024 - Apr-Jun 31**

- roadmap #237  
Git LFS moves to metered billing  
billing cloud ga github enterprise  
github team other
- roadmap #178  
GitHub Enterprise budgets for organization and services  
billing enterprise administration github enterprise
- roadmap #578  
Packages: maven - granular permissions, organization sharing

**Repository Properties #799**  
Open  
github-product-roadmap opened on Jul 11, 2023

**github-product-roadmap**  
on Jul 11, 2023

We are introducing a new feature for repositories called Custom Properties. Think of it as a key-value property store for each repository in which you can store arbitrary data about the repository.

- Organization admins can assign key-value metadata to repositories called custom properties
- Properties can be viewed and edited using the UI or API



# Productivity



# 開発者体験



## 環境構築の効率化

- Codespacesによる迅速な開発環境構築
- Docker ContainerやCLIツールによる構築操作の自動化



## 運用作業の効率化

- サービスデプロイの自動化
- ChatOps (Hubot) や GitHub Appsによるシンプルな運用操作の提供

The screenshot shows a browser window displaying a GitHub blog post. The page title is "How GitHub's Developer Experience team improved innerloop development". Below the title, there is a sub-header "Shipping code quickly and safely in our own distributed environment". The author's name "Author" is visible at the bottom of the article content area.



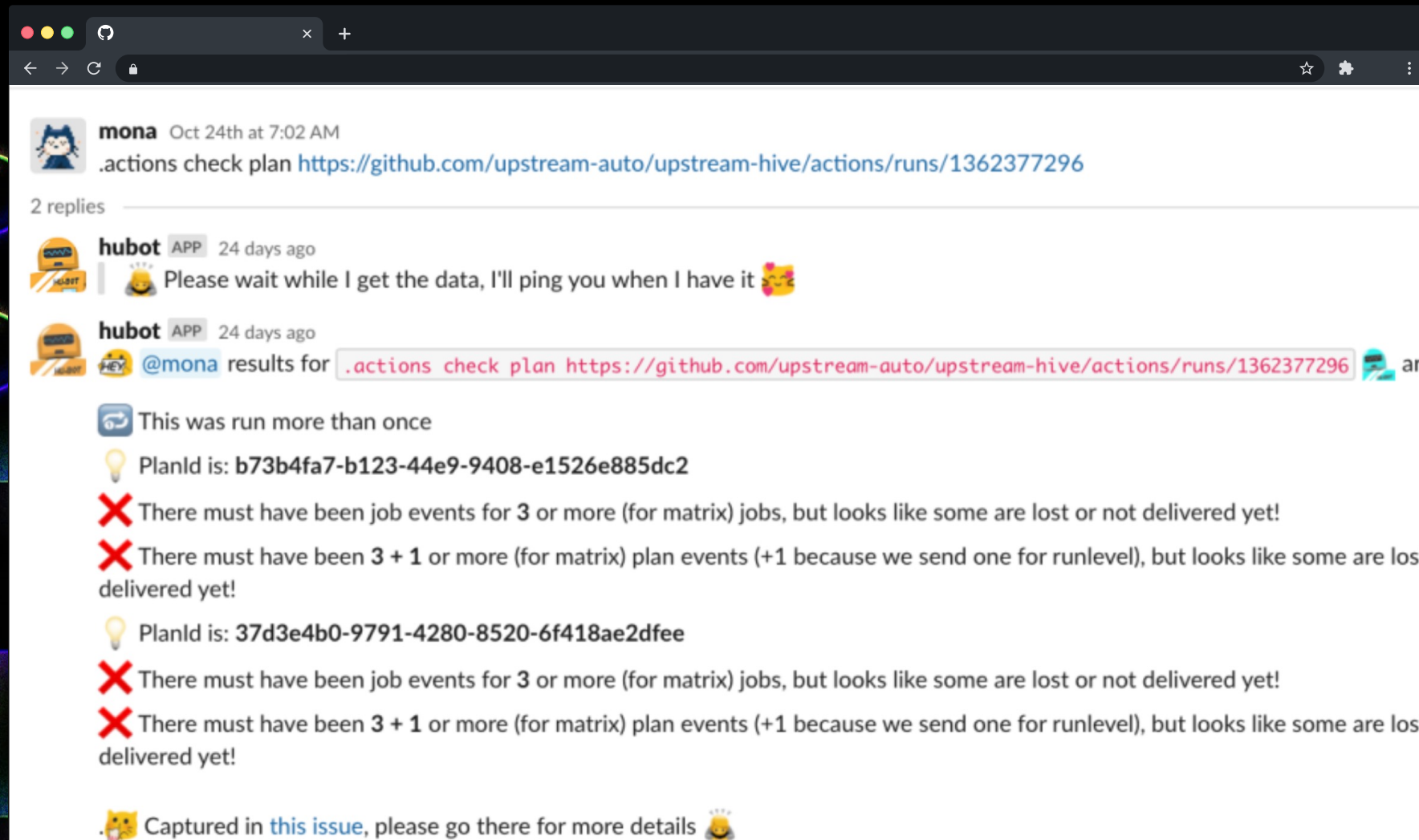
# Codespaces 設定ファイル on “github/github”

(from “How GitHub Builds GitHub with GitHub” video on YouTube)

```
1 {
2   // TODO: Ensure all logs are correctly captured by supervisord.
3   // TODO: Integration of VSCode with developer tools like Rubocop.
4   "name": "Base Dotcom Development",
5   // Runs automatically after Prebuild Template environment is created.
6   "onCreateCommand": ".devcontainer/on-create-command.sh",
7   // Runs automatically once the dev container has been assigned to a user for the first time
8   "postCreateCommand": ".devcontainer/post-create-command.sh",
9   // Runs automatically every time the dev container has been (re-)started
10  "postStartCommand": ".devcontainer/post-start-command.sh",
11  // Runs automatically every time the user has attached to the running container
12  "postAttachCommand": ".devcontainer/post-attach-command.sh",
13  // Nightly bootstrapped image built from https://github.com/github/codespaces-base
14  "image": "ghcr.io/github/github/bootstrap-nightly:latest",
15  "runArgs": ["--privileged", "--security-opt", "seccomp=unconfined"], // Needed to run Docker in Docker.
16  "overrideCommand": false, // We need to call supervisord as the entrypoint as it's used for running system services
17  "hostRequirements": {
18    // Specs for 16 cores, 32 RAM, 128 Gb storage
19    "storage": "128gb",
20    "memory": "32gb",
21    "cpus": 16
22  },
23  "mounts": [
24    "type=tmpfs,destination=/dev/shm,tmpfs-size=536870912"
25  ],
26  "forwardPorts": [
27    80, 2222, 3003, 3033, 3034, 3035, 3306, 3333, 4000, 5800, 42421, 8025, 10042, 9092, 9293, 15000, 15001, 15306,
28    9400, 9091, 43334, 8000, 8001, 9000, 6006
29  ],
30  "appPort": [
31    80, 2222, 3003, 3033, 3034, 3035, 3306, 3333, 4000, 8025, 10042, 9092, 9293, 15000, 15001, 15306, 18081, 8091,
32    9091, 43334, 8000, 8001, 9000, 6006
33  ],
34  "portsAttributes": {
35    "80": {
```

# ChatOps with Hubot at GitHub

(from “Using ChatOps to help Actions on-call engineers” Blog Post)



A screenshot of a GitHub chat interface. At the top, a user named 'mona' posts a message: ".actions check plan <https://github.com/upstream-auto/upstream-hive/actions/runs/1362377296>". Below this, two replies from 'hubot' are shown. The first reply says: "Please wait while I get the data, I'll ping you when I have it". The second reply shows the results for the check plan, including a list of plan IDs and error messages. The error messages state: "There must have been job events for 3 or more (for matrix) jobs, but looks like some are lost or not delivered yet!" and "There must have been 3 + 1 or more (for matrix) plan events (+1 because we send one for runlevel), but looks like some are lost delivered yet!". The chat interface includes a browser window header with navigation icons and a user profile picture.

mona Oct 24th at 7:02 AM  
.actions check plan <https://github.com/upstream-auto/upstream-hive/actions/runs/1362377296>

2 replies

hubot APP 24 days ago  
Please wait while I get the data, I'll ping you when I have it 🐱

hubot APP 24 days ago  
@mona results for `.actions check plan https://github.com/upstream-auto/upstream-hive/actions/runs/1362377296` are

🔄 This was run more than once

💡 PlanId is: `b73b4fa7-b123-44e9-9408-e1526e885dc2`

❌ There must have been job events for 3 or more (for matrix) jobs, but looks like some are lost or not delivered yet!

❌ There must have been 3 + 1 or more (for matrix) plan events (+1 because we send one for runlevel), but looks like some are lost delivered yet!

💡 PlanId is: `37d3e4b0-9791-4280-8520-6f418ae2dfce`

❌ There must have been job events for 3 or more (for matrix) jobs, but looks like some are lost or not delivered yet!

❌ There must have been 3 + 1 or more (for matrix) plan events (+1 because we send one for runlevel), but looks like some are lost delivered yet!

🐱 Captured in [this issue](#), please go there for more details 🐱

# 自動化 - インフラ



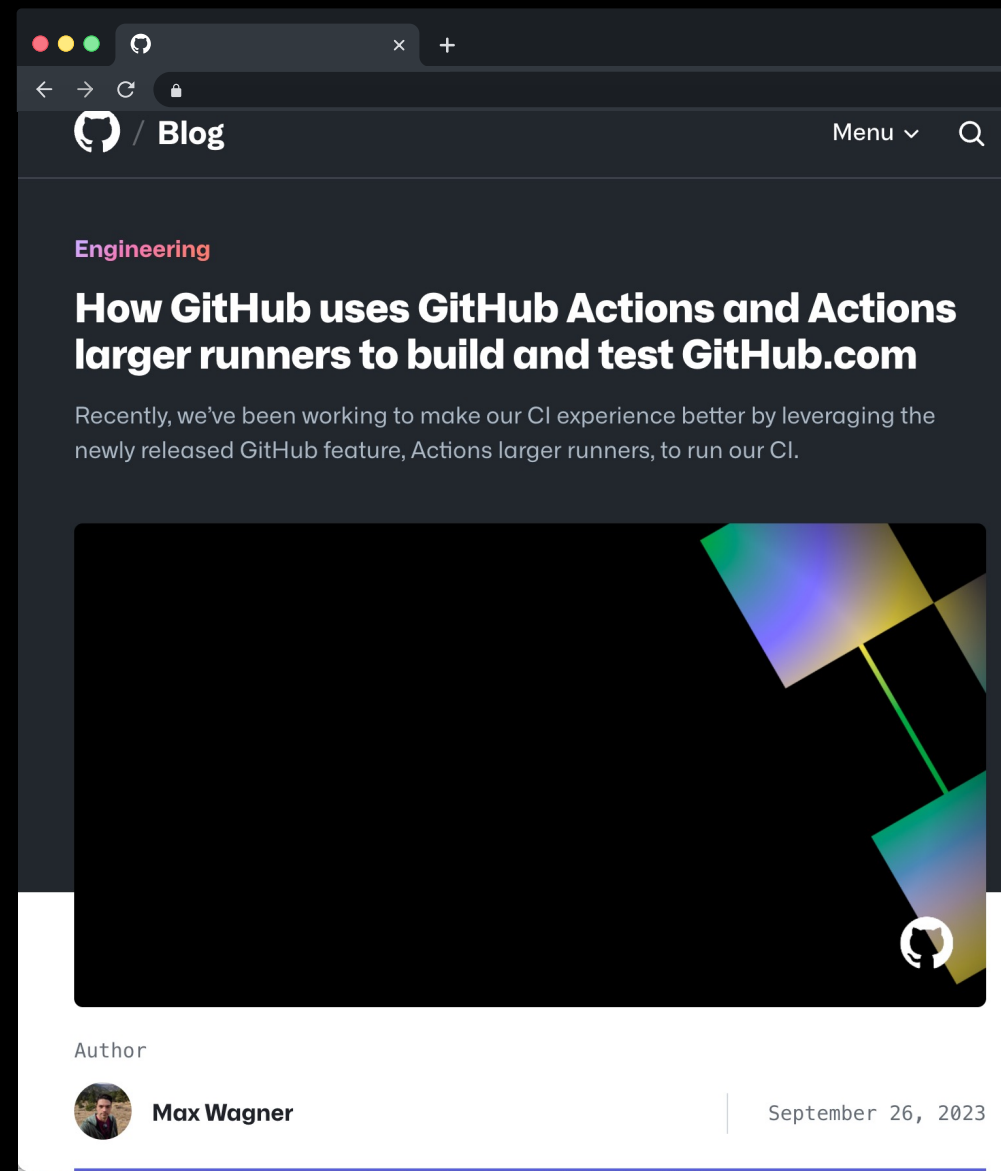
## ランナーの高速化

- Larger Runnerによる大規模な展開（ピークで32-core Runnerの4,500並列, 1時間に15,000ジョブ実行, etc.)



## ワークフローの共通化

- Reusable Workflowによる実装共通化
- 成果物の再利用（キャッシュ）による高速化





# GitHub Actions - 公開リポジトリでの一例 (github/docs - 85 workflows & 300,000 workflow runs)

The screenshot shows the GitHub Actions page for the `github/docs` repository. The left sidebar displays the file structure, with the `workflows` folder expanded. The main content area shows a list of workflows, with the most recent one selected.

Name	Last commit message	Last commit date
..		
add-review-template.yml	Upgrade actions/checkout to v4.1.1 (#49390)	3 months ago
alert-changed-branch-protections.yml	Alert branch protection settings change part 2 (#49178)	3 months ago
auto-close-dependencies.yml	Bump actions/github-script from 6.4.0 to 7.0.0 (#46199)	6 months ago
azure-preview-env-deploy.yml	Disallow any code from public builds into Preview enviro...	5 days ago
azure-preview-env-destroy.yml	bump azure/login to v2 which resolves node16 warnings...	2 months ago
azure-prod-build-deploy.yml	Port start-server to TS (#50078)	2 weeks ago
azure-staging-build-deploy.yml	Port start-server to TS (#50078)	2 weeks ago
check-broken-links-github-github.yml	TypeScript to start server (#49800)	2 months ago

# 自動化 - ビルド



## ビルドの高速化

- ボトルネックの解消
- フィードバックサイクルの短縮

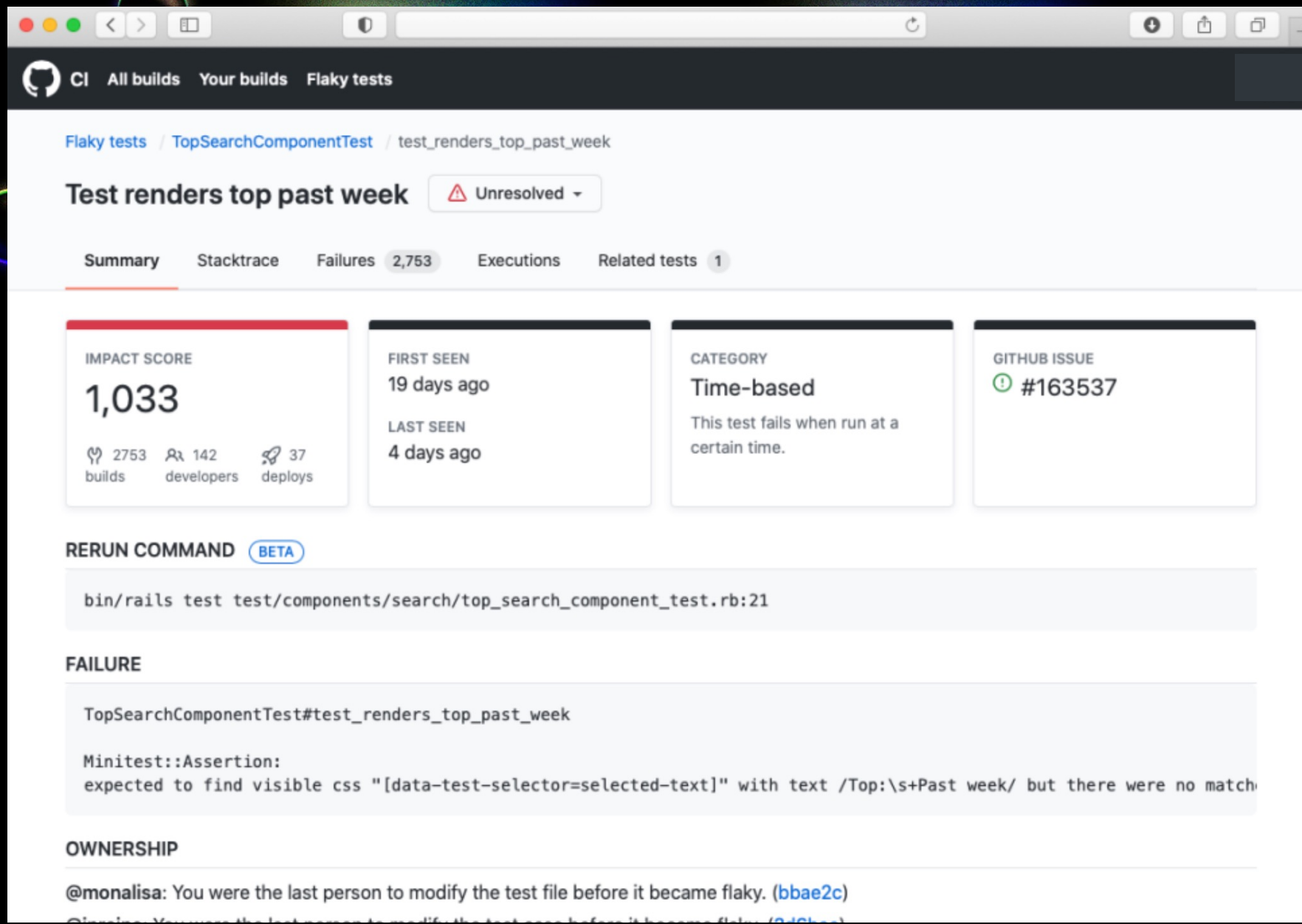


## ビルドの安定化

- Flaky Test (成功・失敗が安定しないテスト) を検出・修正する仕組み

The screenshot shows a browser window displaying a GitHub blog post. The page title is "Reducing flaky builds by 18x" under the "Engineering" category. The text below the title reads: "Part of the Building GitHub blog series. It's four o'clock in the afternoon as you push the last tweak to your branch. Your teammate already reviewed and approved your pull request...". Below the text is an illustration of a person in a yellow hard hat and blue overalls working on a large yellow lightbulb. Another person is sitting on the floor nearby. The author's name "Author" is visible at the bottom of the illustration area.

# Flakes – Flaky Testの検出 / 可視化 (GitHub Apps)



The screenshot shows the GitHub Flaky Tests interface for a specific test. The breadcrumb navigation is "Flaky tests / TopSearchComponentTest / test\_renders\_top\_past\_week". The test name is "Test renders top past week" with a status of "Unresolved". The interface includes tabs for "Summary", "Stacktrace", "Failures" (2,753), "Executions", and "Related tests" (1). A summary card displays an "IMPACT SCORE" of 1,033, with sub-metrics for 2753 builds, 142 developers, and 37 deploys. It also shows "FIRST SEEN" (19 days ago) and "LAST SEEN" (4 days ago). The "CATEGORY" is "Time-based" with a description: "This test fails when run at a certain time." A "GITHUB ISSUE" link points to "#163537". A "RERUN COMMAND" section shows the command: `bin/rails test test/components/search/top_search_component_test.rb:21`. The "FAILURE" section shows the error message: `TopSearchComponentTest#test_renders_top_past_week` and `Minitest::Assertion: expected to find visible css "[data-test-selector=selected-text]" with text /Top:\s+Past week/ but there were no match`. The "OWNERSHIP" section lists "@monalisa" as the last person to modify the test file before it became flaky.

CI All builds Your builds Flaky tests

Flaky tests / TopSearchComponentTest / test\_renders\_top\_past\_week

## Test renders top past week Unresolved

Summary Stacktrace Failures 2,753 Executions Related tests 1

**IMPACT SCORE**

# 1,033

👤 2753 builds    👤 142 developers    🚀 37 deploys

**FIRST SEEN**

19 days ago

**LAST SEEN**

4 days ago

**CATEGORY**

### Time-based

This test fails when run at a certain time.

**GITHUB ISSUE**

[#163537](#)

**RERUN COMMAND** BETA

```
bin/rails test test/components/search/top_search_component_test.rb:21
```

**FAILURE**

```
TopSearchComponentTest#test_renders_top_past_week
```

```
Minitest::Assertion: expected to find visible css "[data-test-selector=selected-text]" with text /Top:\s+Past week/ but there were no match
```

**OWNERSHIP**

@monalisa: You were the last person to modify the test file before it became flaky. ([bbae2c](#))

@gimnig: You were the last person to modify the test file before it became flaky. ([2161e1](#))



# Security

# セキュリティ オペレーション



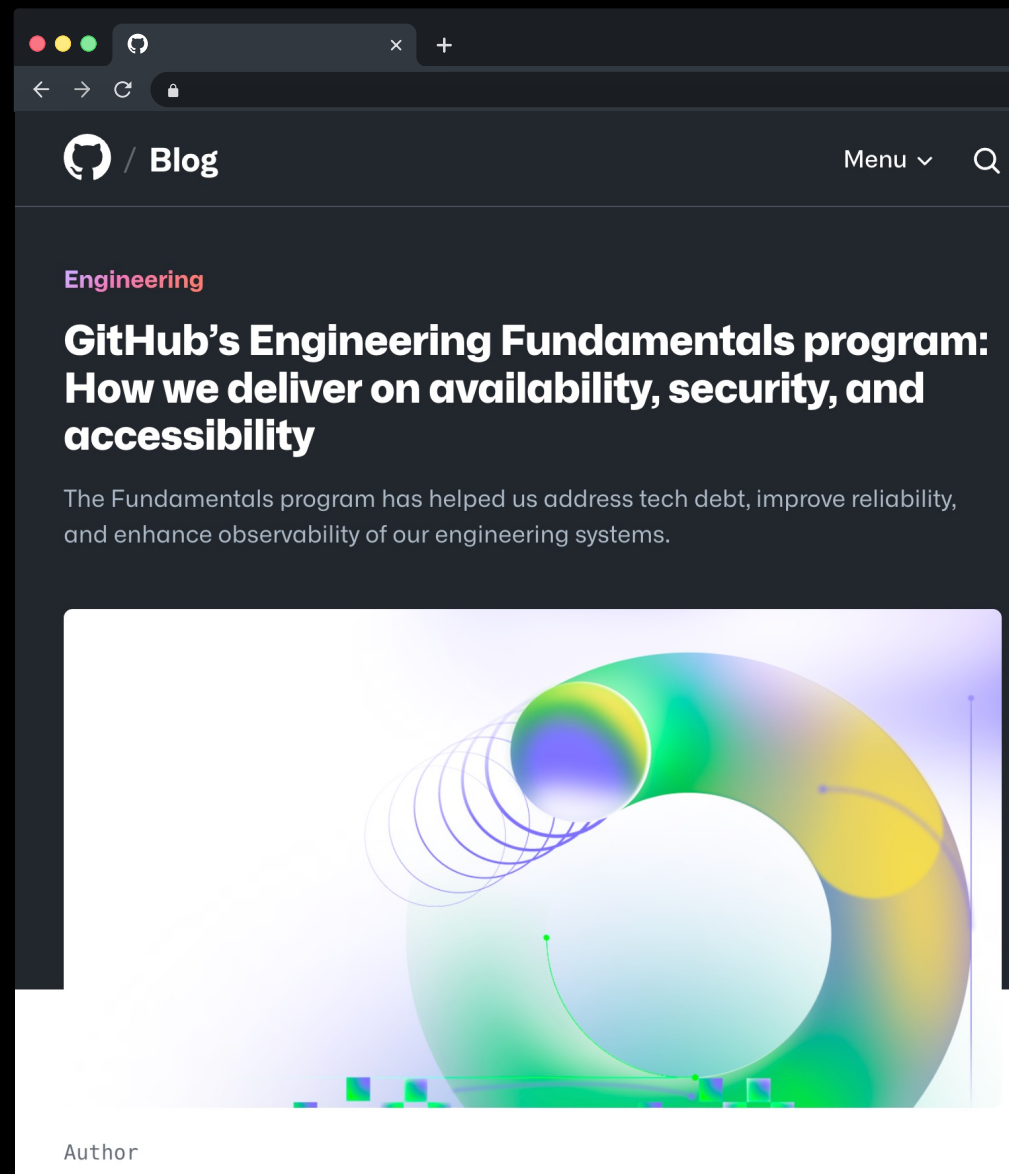
## 基礎となるプログラム

- 早期段階でのSecurity担保 (GitHub Advanced Security)
- GitHubを構成するService群のOwnership明確化 (SERVICEOWNERSファイル)



## 可視化・ダッシュボード

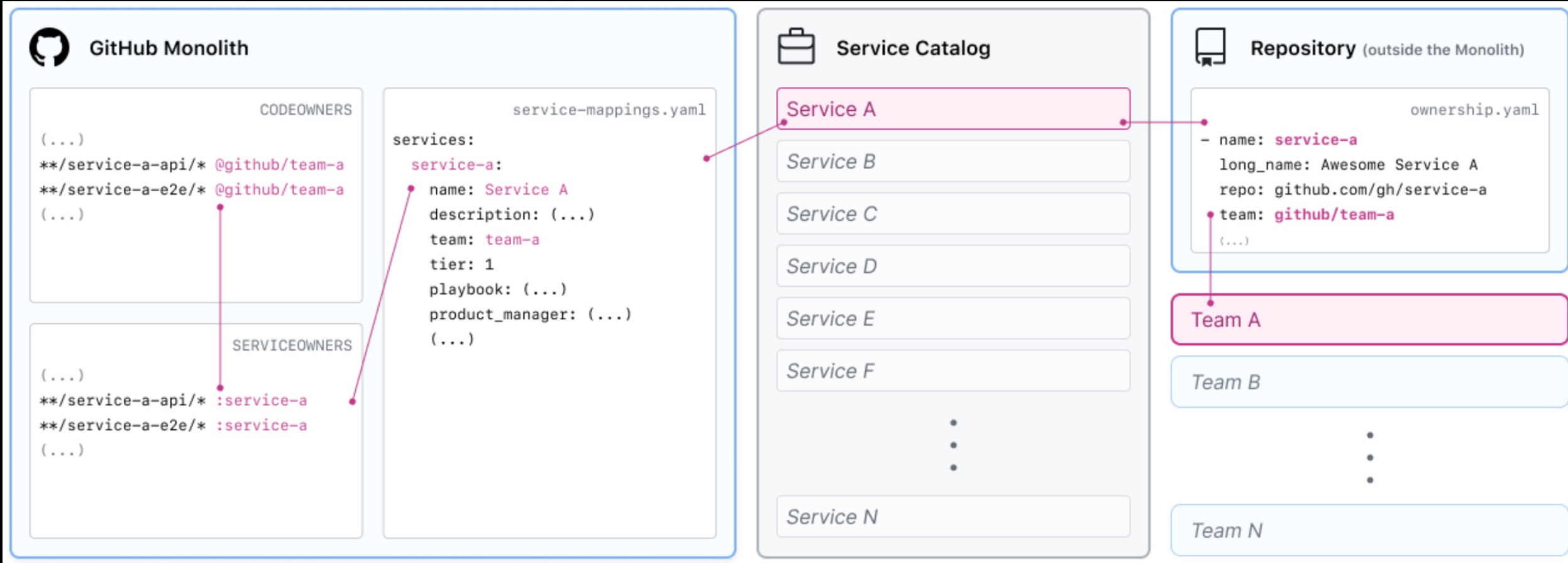
- GitHub内での情報集約 (GitHub Issue)
- 必要に応じたカスタムレポート (GitHub Pages / GitHub Apps)





# GitHubにおけるService Ownershipの定義・管理

(from “How we organize and get things done with SERVICEOWNERS” blog post)





# GitHub AppsによるDashboard (Secret Scanning視点のScorecard)

**Scorecards / Secret Scanning** 🗄️ <>

Scorecard Owner: @github/product-security-engineering • Slack: #pse-eng • Fundamental: Yes

[🏠 Overview](#) [📈 Trends](#) [📊 Scores](#) [🕒 Unmet Requirements](#) [± Score Changes](#) [📉 Charts](#)

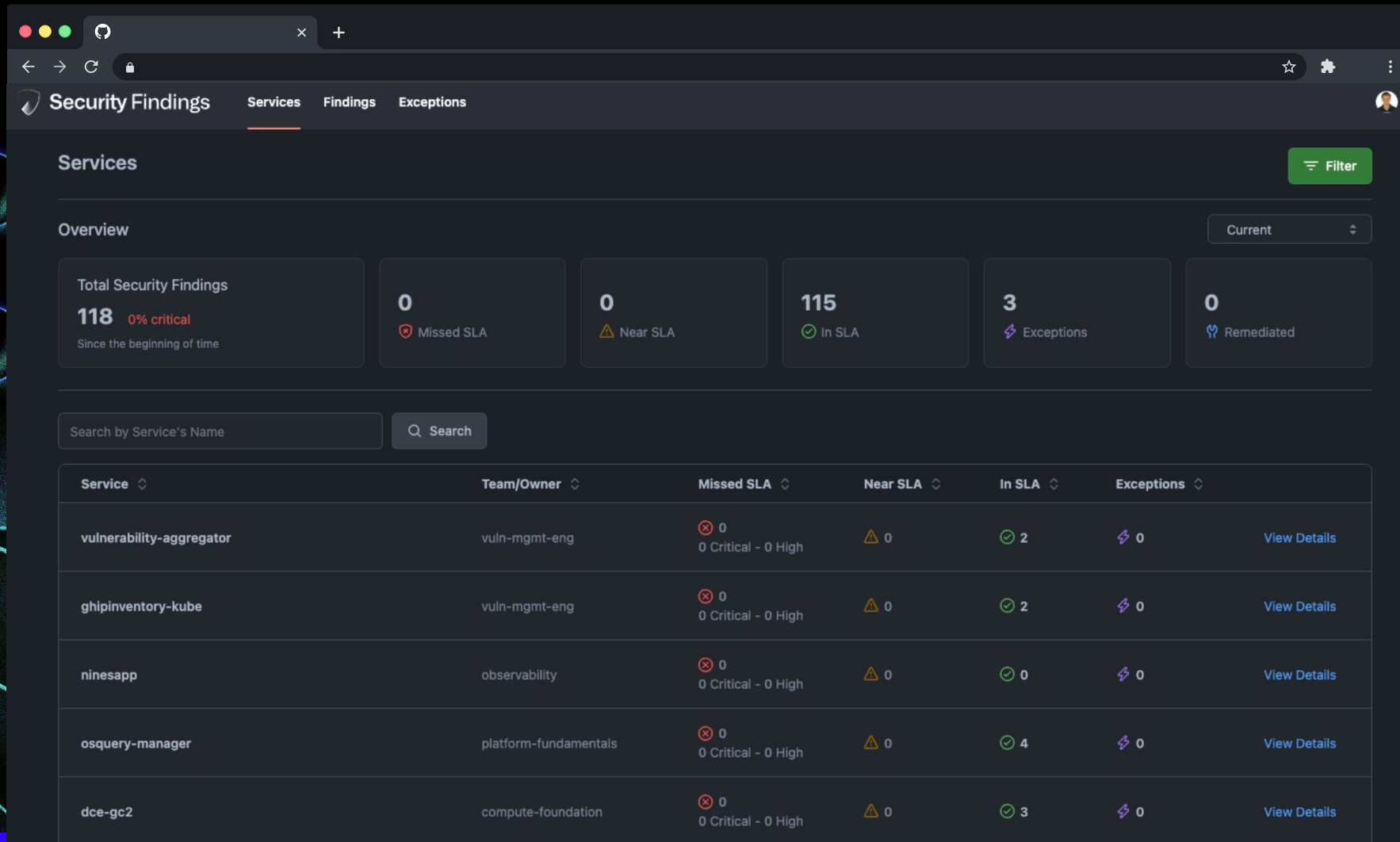
Ensures that services do not have Secret Scanning alerts for checked in secrets.  
For feedback or requests, please [open an issue in the github/product-security-engineering repository](#).

Select filter ▾ Filter services Search syntax tips 📄 ✕ [Save new filter](#)

### Objectives

<b>Risks</b> The service has fixed all outstanding risks or is in the target fix window.	
This service has no secrets checked in to the repository. 🗄️ 85 points	Requirement met by 1922 of 1922 services (100%)
This service has push protection turned on in the repository. 🗄️ 10 points	Requirement met by 1922 of 1922 services (100%)
This service has secret scanning turned on in the repository. 🗄️ 5 points	Requirement met by 1922 of 1922 services (100%)

# GitHub AppsによるDashboard (サービスごとの脆弱性)



# GitHub IssueによるSecurity Findingsのトラッキング

(from “Scaling vulnerability management across thousands of services and more than 150 million findings” blog post)

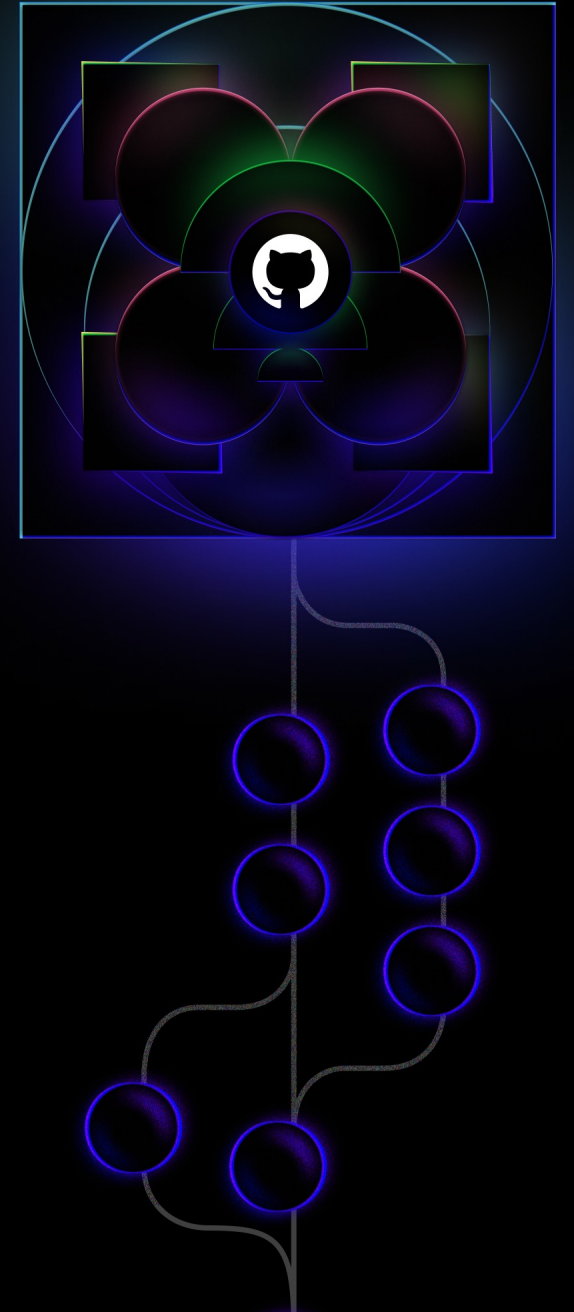
The screenshot shows a GitHub issue page for a repository named 'vuln-notifier-prod'. The issue is titled 'Security Findings Summary Issue' and was opened by a bot named 'vuln-notifier-prod' 2 weeks ago. The issue body contains a welcome message from the 'sentinel' bot, explaining that this is a tracking issue for security findings. It includes a note about automatic maintenance and links for help and feedback. Below the text is a table of security findings.

Criticality	Title	Due Date	Tracking Issue	Security Findings
high	Vulnerabilities for libc-bin (2.31-13+deb11u6)	2023-11-07	<a href="#">Link</a>	<a href="#">Link</a>

On the right side of the issue, there are several metadata sections: Assignees (No one), Labels (application: sentinel, Automated Gripe Vulnerability, Owner: code-analysis, summary), Projects, Milestone (No milestone), Development (Create a branch for this issue or link a pull request.), and Notifications (Unsubscribe button).



# Thank you



# Links

- [YouTube - How GitHub Builds GitHub with GitHub](#)
- [YouTube - How GitHub builds GitHub](#)
- [Blog - How to communicate like a GitHub engineer: our principles, practices, and tools](#)
- [Blog - How we're using GitHub Projects to standardize our workflows and stay aligned](#)
- [Blog - How GitHub's Developer Experience team improved innerloop development](#)
- [Blog - Using ChatOps to help Actions on-call engineers](#)
- [Blog - How GitHub uses GitHub Actions and Actions larger runners to build and test GitHub.com](#)
- [Blog - How GitHub uses merge queue to ship hundreds of changes every day](#)
- [Blog - Making GitHub CI workflow 3x faster](#)
- [Blog - Reducing flaky builds by 18x](#)
- [Blog - GitHub's Engineering Fundamentals program: How we deliver on availability, security, and accessibility](#)
- [Blog - How we organize and get things done with SERVICEOWNERS](#)
- [Blog - Scaling vulnerability management across thousands of services and more than 150 million findings](#)