

OSS Part 1 Presentation

Term Project

23102003 Kim Doyi

23102009 Park Shinhung

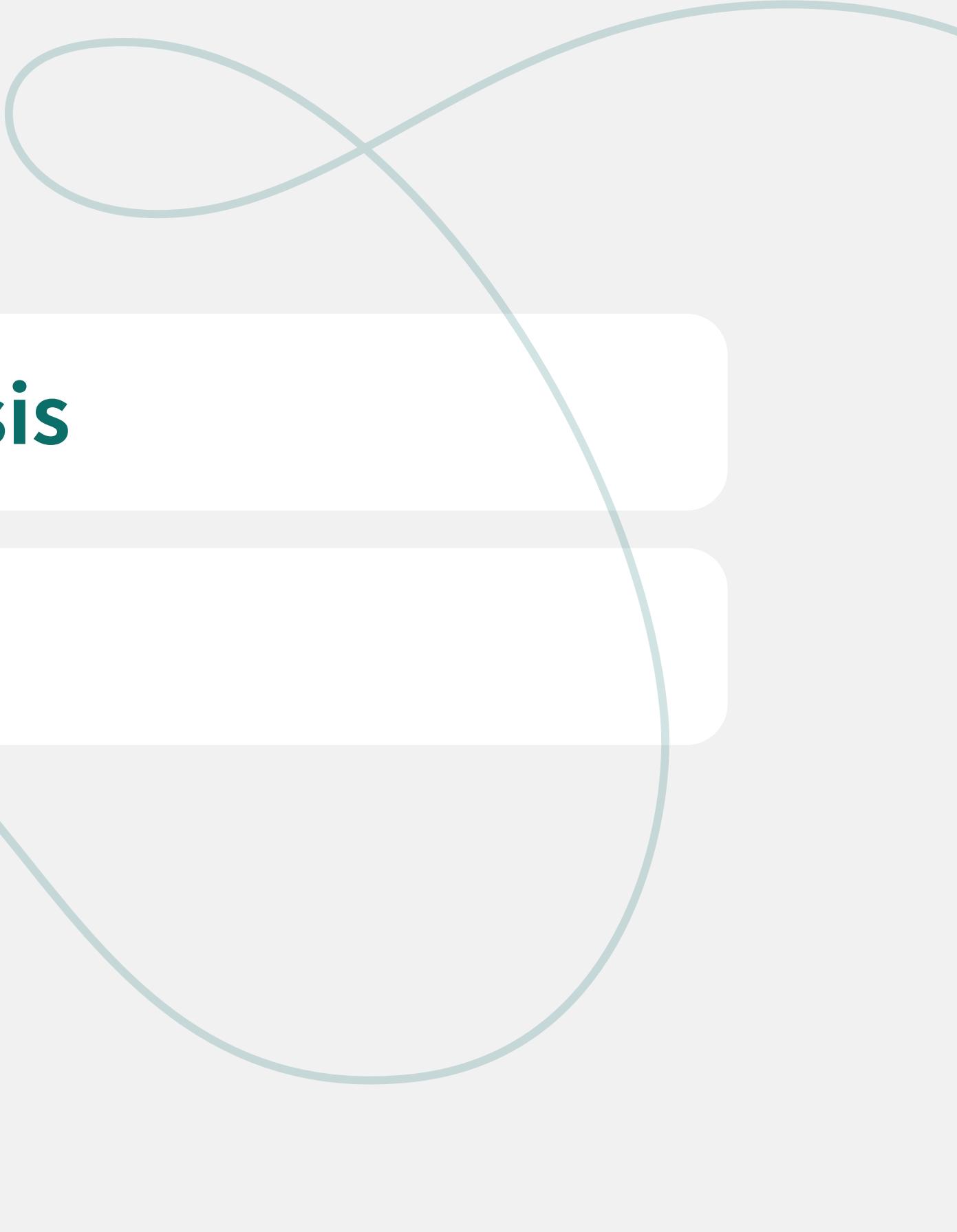
23102015 Oh Eunyoung

23102025 Lee Haneol

23102031 Hwang Yuyoung



CONTENT



01 | OSS Community Analysis

02 | Project Proposal

1-1. Open Source Project



Team collaboration platform



UI components platform

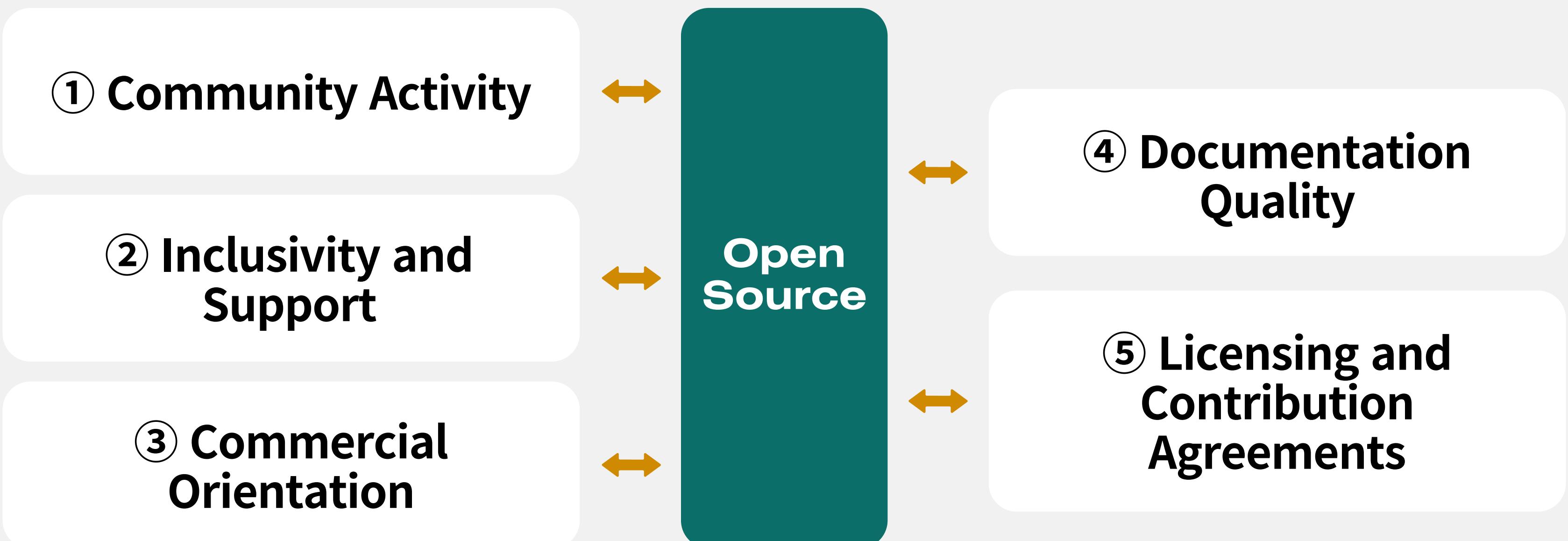


Project management and task tracking tool



Game engine platform

1-2. Evaluation Criteria



1-3. Analysis for Community Activity

Commit Frequency



ROCKET.CHAT

Total Commit

⌚ 26,610 Commits

```
상유영 @DESKTOP-3VCV5FF MINGW64 ~/Rocket.Chat (develop)
$ git log --all --since="2023-11-19" --until="2024-11-19" --pretty=format:"%ad" | wc -l
6733
```

Commit the last 1 year: 6733

- Monthly Average: 561.083
- Weekly Average: 129.480



Up until recently, commit is still active

TOAST UI

Total Commit

⌚ 361 Commits

```
shinhyung@bagsinhyeong-ui-MacBookAir ~/desktop/toast/tui.editor (master)
$ git log --all --since="2023-11-19" --until="2024-11-19" --pretty=oneline | wc -l
0
shinhyung@bagsinhyeong-ui-MacBookAir ~/desktop/toast/tui.editor (master)
[$ git log --since="2020-11-19" --until="2021-11-19" --pretty=oneline | wc -l
229
```

Commit the last 1 year: 0
Commit during a active year: 229

- Monthly Average: 23.583
- Weekly Average: 5.442



Not active commit

1-3. Analysis for Community Activity

Commit Frequency



Total Commit

⌚ 3,642 Commits

```
user@PC-16 MINGW64 ~/focalboard (main)
$ git log --all --since="2023-11-19" --until="2024-11-19" --pretty=format:"%ad"
| wc -l
20
user@PC-16 MINGW64 ~/focalboard (main)
$ git log --all --since="2022-01-01" --until="2023-01-01" --pretty=format:"%ad"
| wc -l
1932
```

Commit the last 1 year: 20

Commit during a active year: 1932

- Monthly Average: 161
- Weekly Average: 37.153

→ Recent Commit Not Active compared to the past



Total Commit

⌚ 69,207 Commits

```
user@PC-15 MINGW64 ~/godot (master)
$ git log --all --since="2023-11-19" --until="2024-11-19" --pretty=format:"%ad" | wc -l
11416
```

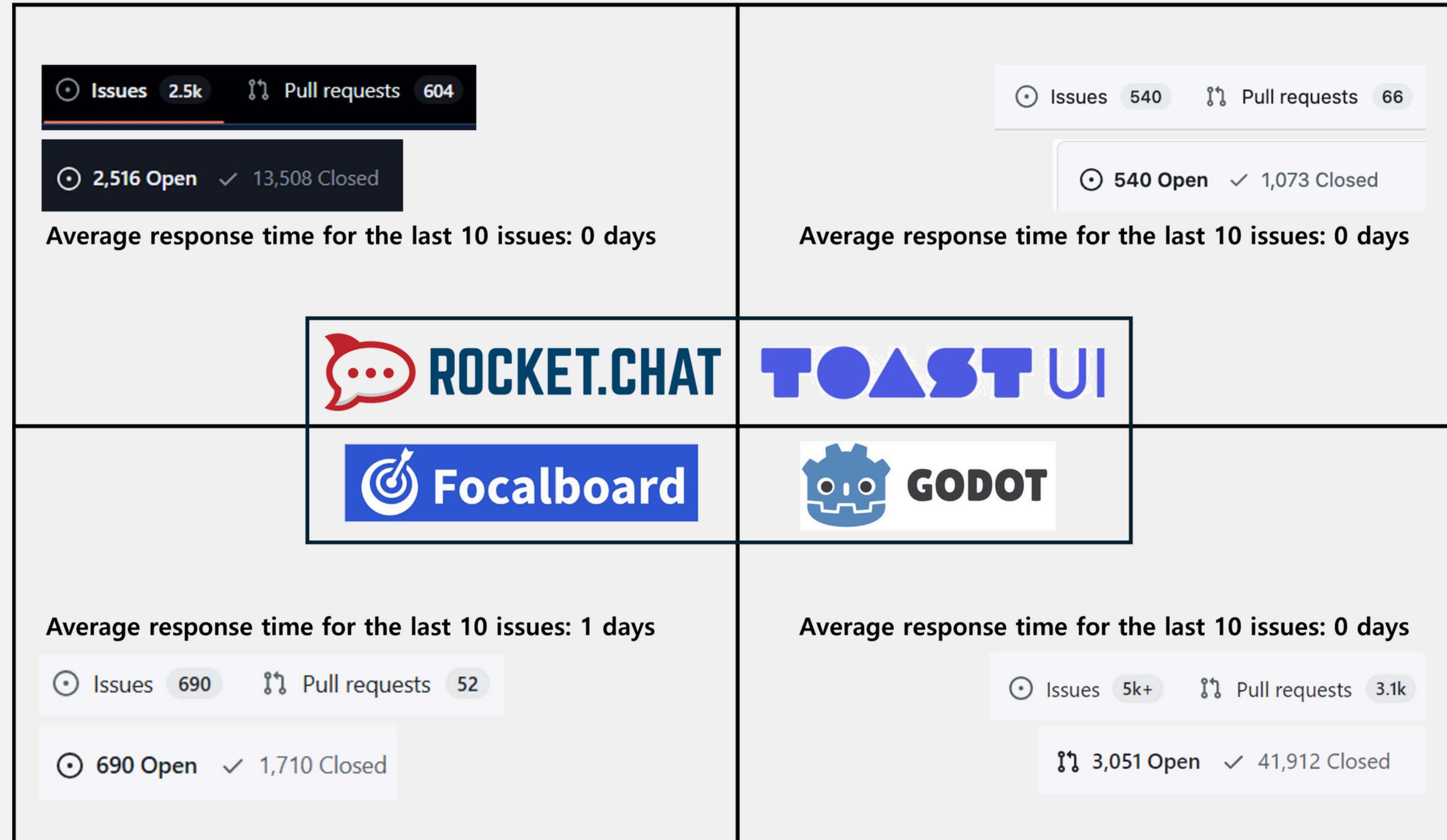
Commit the last 1 year: 11416

- Monthly Average: 951.333
- Weekly Average: 219.538

→ Up until recently, commit is still active

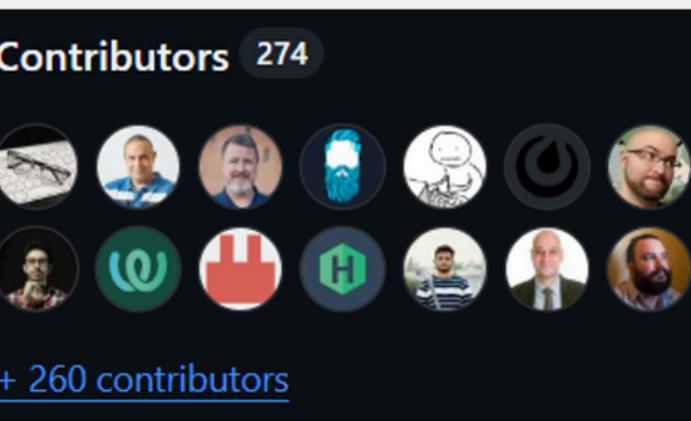
1-3. Analysis for Community Activity

Issue Response Time



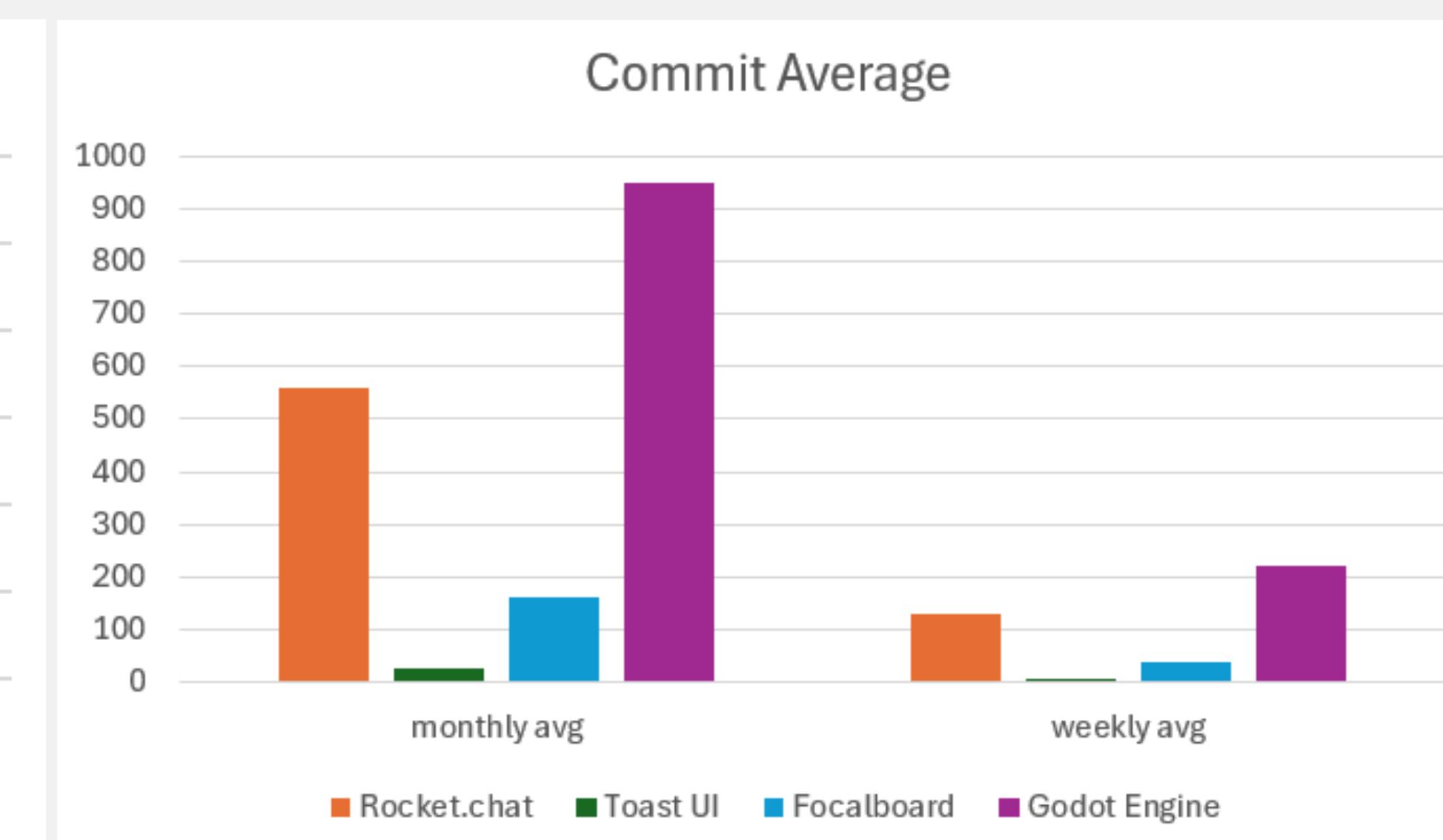
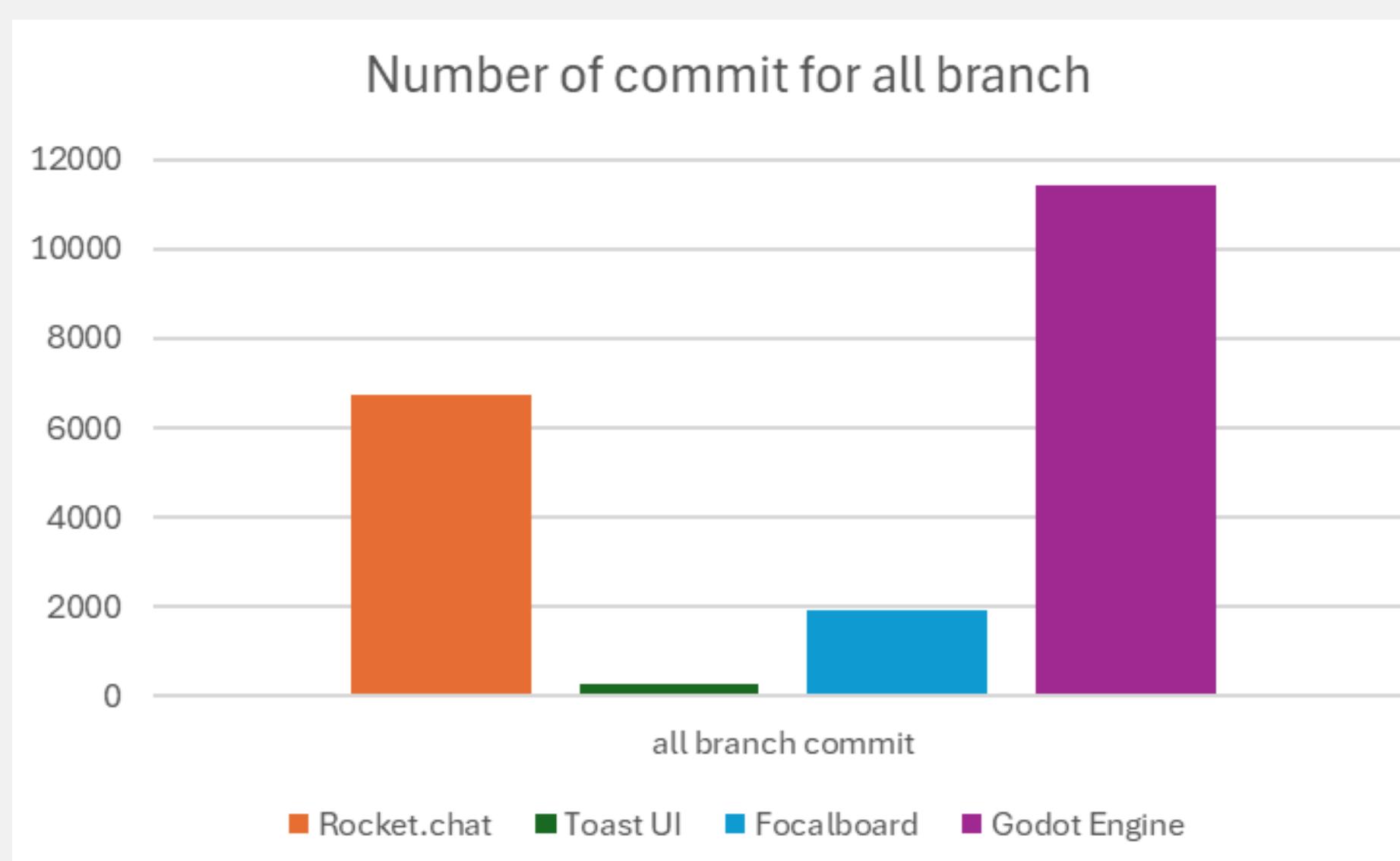
1-3. Analysis for Community Activity

Number of Contributors

 <p>Contributors 873 + 859 contributors</p>	 <p>Contributors 113 + 99 contributors</p>
 ROCKET.CHAT	
 Focalboard	 GODOT
 <p>Contributors 274 + 260 contributors</p>	 <p>Contributors 2,760 + 2,746 contributors</p>

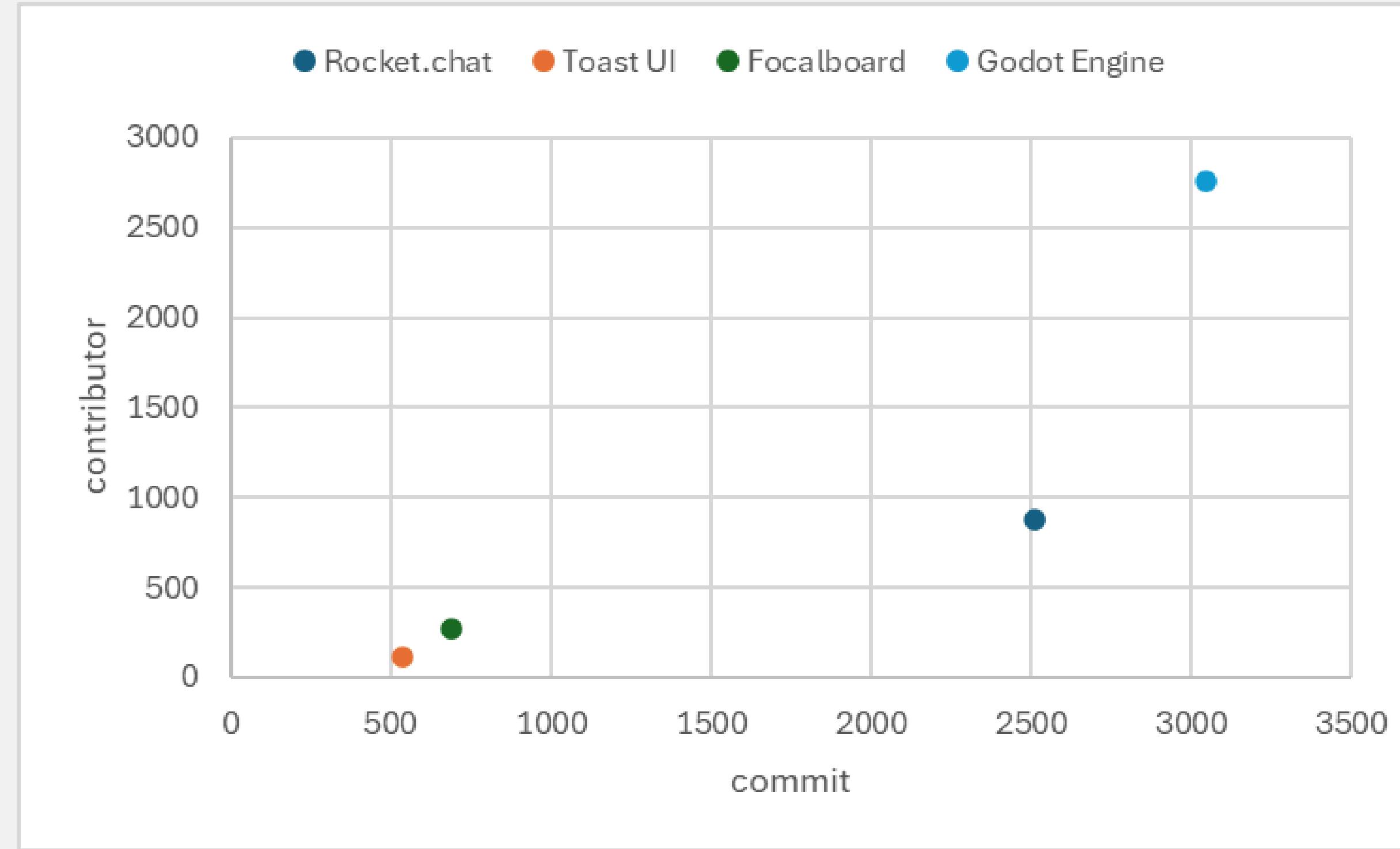
1-3. Analysis for Community Activity

Comprehensive Comparison



1-3. Analysis for Community Activity

Comprehensive Comparison



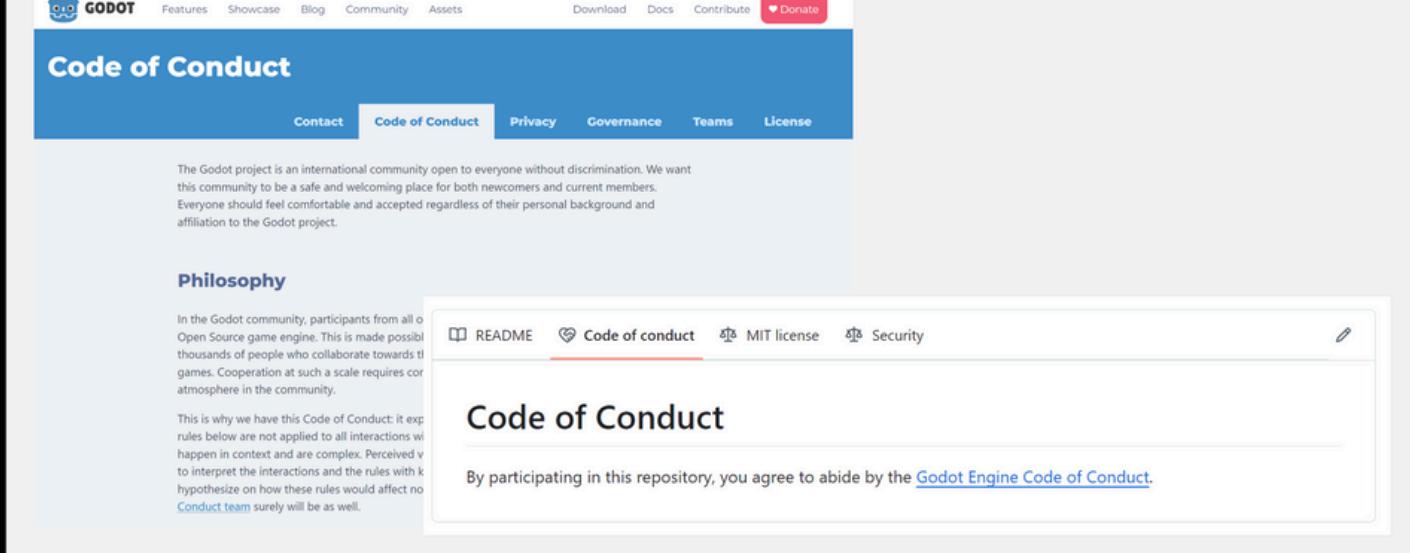
Contributor ↑
Number of commit ↑

1-3. Analysis for Inclusivity and Support

Beginner-Friendly Issues & Community

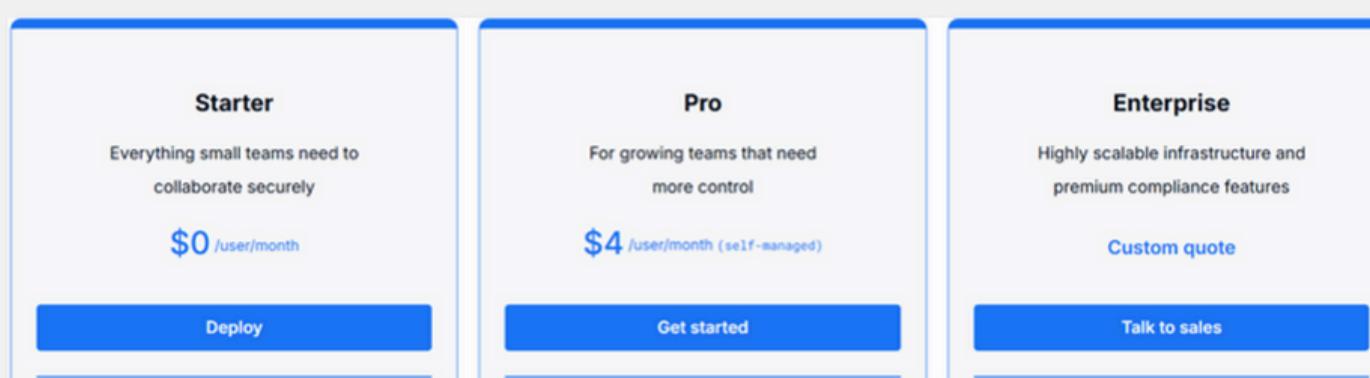
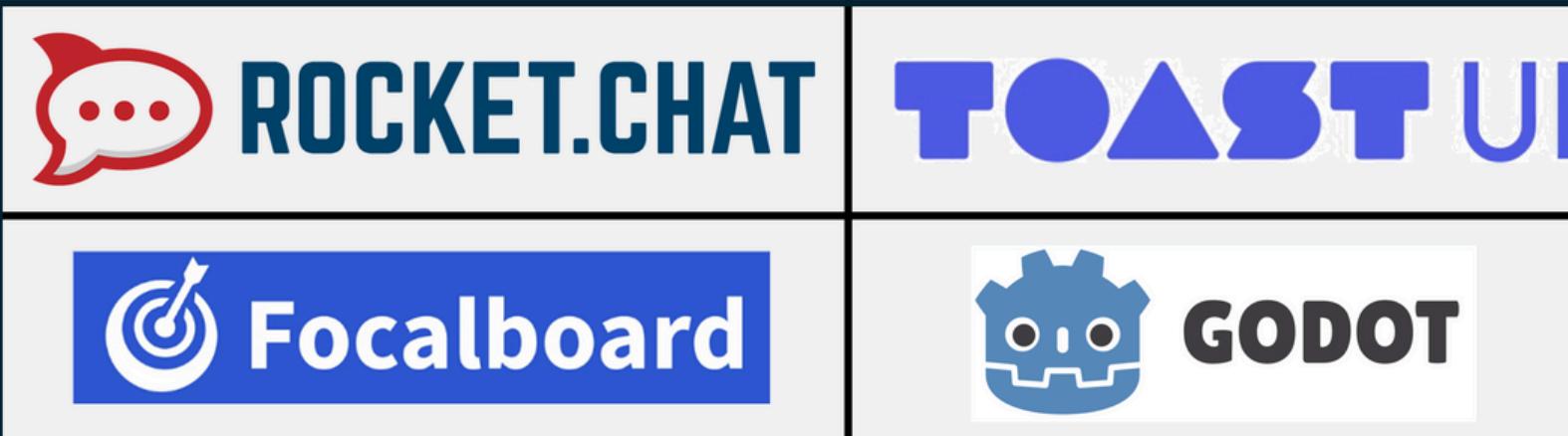
<p>contrib: good first issue</p> <p>Labels 151</p> <p>Want to contribute to RocketChat/Rocket.Chat? If you have a bug or an idea, read the contributing guidelines before opening an issue.</p> <p>Community Join thousands of members worldwide in our community server. Join #support and #general for help from the community.</p> <p> ROCKET.CHAT</p>	<p>Good First Issue 🤖🤖</p> <p>48 labels</p> <p>Want to contribute to nhn/tui.editor? If you have a bug or an idea, read the contributing guidelines before opening an issue.</p> <p>No official community exists</p> <p></p>
<p>Labels 57</p> <p>Good First Issue</p> <p>Good Idea Fairy</p> <p>Mattermost</p> <p>Log in to your account Collaborate with your team in real-time</p> <p>Want to contribute to mattermost-community/focalboard? If you have a bug or an idea, read the contributing guidelines before opening an issue. If you're ready to tackle some open issues, we've collected some good first issues for you.</p>	<p>63 labels</p> <p>good first issue</p> <p>Want to contribute to godotengine/godot? If you have a bug or an idea, read the contributing guidelines before opening an issue. If you're ready to tackle some open issues, we've collected some good first issues for you.</p> <p> GODOT</p>

1-3. Analysis for Inclusivity and Support Code of Conduct

<p> Code of conduct</p> <p>Contributor Covenant Code of Conduct</p> <p>Our Pledge</p> <p>In the interest of fostering an open and welcoming environment, we as contributors participation in our project and our community a harassment-free experience for everyone, regardless of age, size, disability, ethnicity, gender identity and expression, level of experience, nationality, personal appearance, race, religion, or sexual identity and orientation.</p> <p>Our Standards</p> <p>Examples of behavior that contributes to creating a positive environment include:</p> <ul style="list-style-type: none">• Using welcoming and inclusive language• Being respectful of differing viewpoints and experiences• Gracefully accepting constructive criticism	<p> Code of conduct</p> <p>Contributor Covenant Code of Conduct</p> <p>Our Pledge</p> <p>In the interest of fostering an open and welcoming environment, we as contributors making participation in our project and our community a harassment-free experience for everyone, regardless of age, size, disability, ethnicity, gender identity and expression, level of experience, nationality, personal appearance, race, religion, or sexual identity and orientation.</p> <p>Our Standards</p> <p>Examples of behavior that contributes to creating a positive environment include:</p> <ul style="list-style-type: none">• Using welcoming and inclusive language• Being respectful of differing viewpoints and experiences• Gracefully accepting constructive criticism
<p> ROCKET.CHAT</p> <p></p>	<p></p> <p> GODOT</p>
<p>Does not include a dedicated Code of Conduct file</p>	<p> The Godot project is an international community open to everyone without discrimination. We want this community to be a safe and welcoming place for both newcomers and current members. Everyone should feel comfortable and accepted regardless of their personal background and affiliation to the Godot project.</p> <p>Philosophy</p> <p>In the Godot community, participants from all over the world come together to work on an Open Source game engine. This is made possible by the fact that there are thousands of people who collaborate towards the same goal. Cooperation at such a scale requires a certain level of trust and respect between the participants.</p> <p>This is why we have this Code of Conduct: it explains how we expect people to interact with each other. The rules below are not applied to all interactions, as some may happen in context and are complex. Perceived violations of the rules may be interpreted differently by different people. It is up to the community to interpret the interactions and the rules with care, and to hypothesize on how these rules would affect the community.</p> <p>Code of Conduct</p> <p>By participating in this repository, you agree to abide by the Godot Engine Code of Conduct.</p>

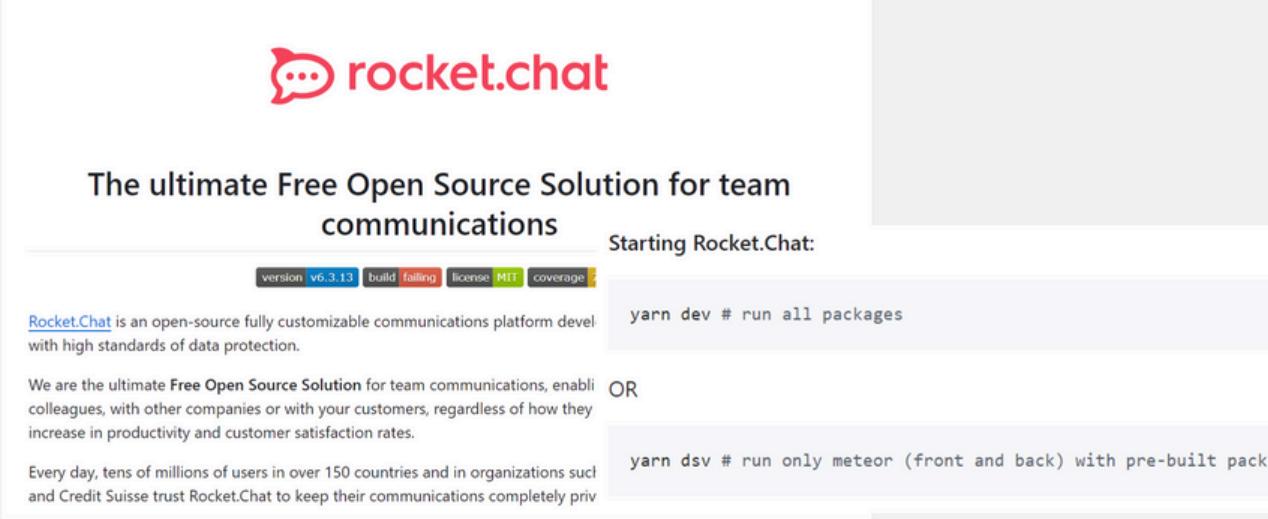
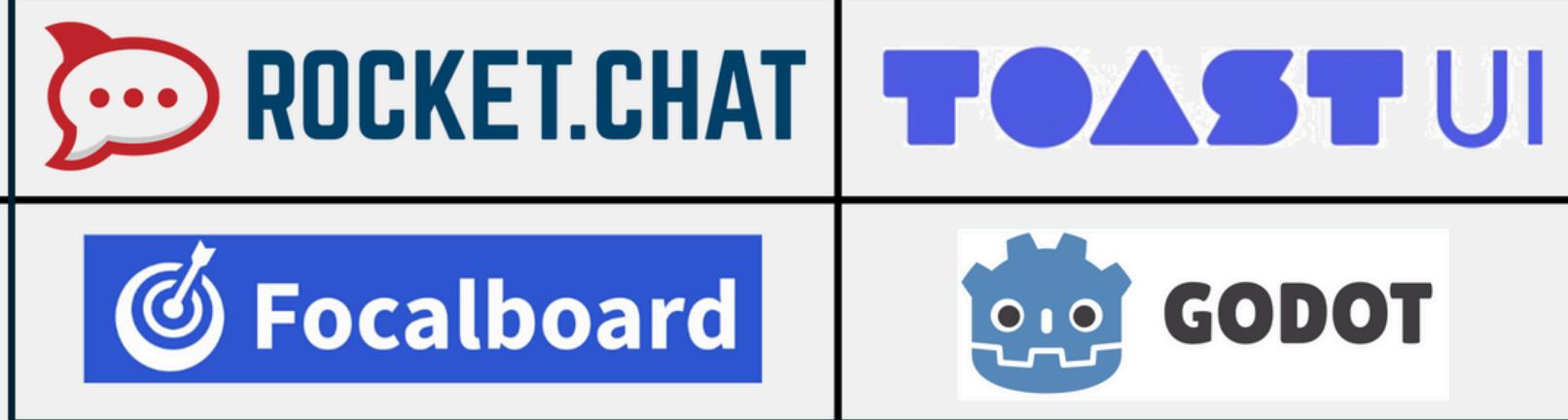
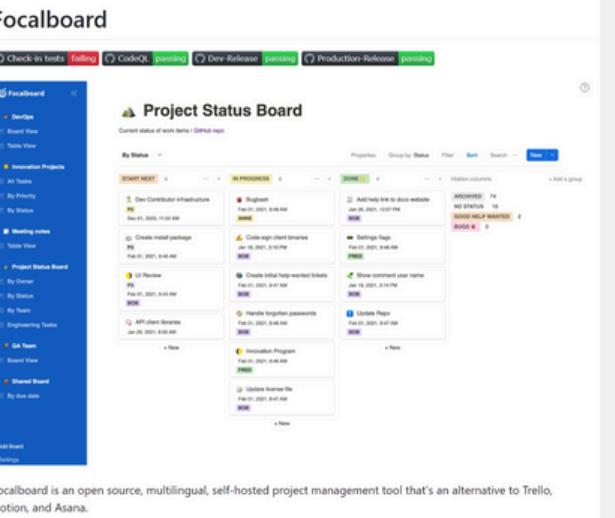
1-3. Analysis for Commercial Orientation

Business Model

 <p>Try Rocket.Chat immediately with our SaaS trial. Free for 30 days.</p> <p>SaaS free trial</p>	<p>Does not specify Business Model</p>
 <p>Focalboard</p> <p>GODOT</p>	<p>Does not specify Business Model</p> <p>Does not specify Business Model</p>

1-3. Analysis for Documentation Quality

README.md Clarity

 <p>The ultimate Free Open Source Solution for team communications</p> <p>Starting Rocket.Chat:</p> <pre>yarn dev # run all packages</pre> <p>OR</p> <pre>yarn dsv # run only meteor (front and back) with pre-built packages</pre>	<h3>Why TOAST UI Editor?</h3> <p>TOAST UI Editor provides Markdown mode and WYSIWYG mode. Depending on the type of use you want like production of <i>Markdown</i> or maybe to just edit the <i>Markdown</i>. The TOAST UI Editor can be helpful for both the usage. It offers Markdown mode and WYSIWYG mode, which can be switched.</p> <h3>Internationalization (i18n)</h3> <p>release editor@3.2.2 npm v3.2.2</p> <ul style="list-style-type: none">PackagesWhy TOAST UI Editor?FeaturesExamplesBrowser SupportPull Request StepsContributingTOAST UI FamilyUsed ByLicense	
 <p>ROCKET.CHAT</p> <p>TOAST UI</p> <p>Focalboard</p> <p>GODOT</p>		
 <p>Focalboard</p> <p>Project Status Board</p> <p>Focalboard is an open source, multilingual, self-hosted project management tool that's an alternative to Trello, Notion, and Asana.</p>	<p>Create an <code>.env</code> file in the focalboard directory that contains:</p> <pre>EXCLUDE_ENTERPRISE="1"</pre> <p>To build the server:</p> <pre>make prebuild make</pre> <p>To run the server:</p> <pre>./bin/focalboard-server</pre>	<h3>Godot Engine</h3>  <p>GODOT Game engine</p> <p>2D and 3D cross-platform game engine</p> <p>Godot Engine is a feature-packed, cross-platform game engine to create 2D and 3D games from a unified interface. It provides a comprehensive set of common tools, so that users can focus on making games without having to reinvent the wheel. Games can be exported with one click to a number of platforms, including the major desktop platforms (Linux, macOS, Windows), mobile platforms (Android, iOS), as well as Web-based platforms and consoles.</p>

1-3. Analysis for Documentation Quality

Initial Setup Instructions

<p>Prerequisites</p> <p>You can follow these instructions to setup a dev environment:</p> <ul style="list-style-type: none">• Install Node 20.x (LTS) either manually or using a tool like nvm or volta (recommended)• Install Meteor (version here): https://docs.meteor.com/about/install.html• Install yarn: https://yarnpkg.com/getting-started/install• Install Deno 1.x: https://docs.deno.com/runtime/fundamentals/installation/• Clone this repo: <code>git clone https://github.com/RocketChat/Rocket.Chat.git</code> <p>Server Environment Setup</p> <p>You can create a Rocket.Chat server development environment to modify and contribute to the Rocket.Chat source code, regardless of your system. Install and configure necessary dependencies on your machine following the steps for your specific operating system.</p> <p>To deploy Rocket.Chat for your day-to-day use, follow the recommended methods in the official deployment guide.</p> <p>Gitpod: Use the online development environment pre-configuration to develop Rocket.Chat server. Linux: See how to set up a Rocket.Chat server development environment on any Linux distribution. Mac OS X: Learn how to set up a Rocket.Chat server development environment on Mac. Windows 10: Set up a Rocket.Chat server development environment on Windows.</p>	<p>Setup</p> <p>Fork <code>main</code> branch into your personal repository. Clone it to local computer. Install node modules. Before starting development, you should check if there are any errors.</p> <pre>\$ git clone https://github.com/{your-personal-repo}/tui.editor.git \$ npm install \$ npm run build toastmark \$ npm run test editor</pre>
 ROCKET.CHAT	 TOAST UI
<p>Install prerequisites</p> <p>All</p> <ul style="list-style-type: none">• Git (if using Windows, see below)• Go• Node.js (v10+)• npm <p>Windows</p> <ul style="list-style-type: none">• Install MinGW-w64 via Chocolatey• Install Git for Windows and use the git-bash terminal shell <p>Mac</p> <ul style="list-style-type: none">• Install Xcode (v12+)• Install the Xcode Command Line Tools via <code>xcode-select --install</code> <p>Linux</p> <ul style="list-style-type: none">• <code>sudo apt-get install libgtk-3-dev</code>• <code>sudo apt-get install libwebkit2gtk-4.0-dev</code>• <code>sudo apt-get install autoconf dh-autoreconf</code>	<p>Personal server setup guide</p> <p>This guide will help you configure your developer environment for the Focalboard Personal Server. For most features, this is the easiest way to get started working against code that ships across editions. For working with Mattermost Boards (Focalboard as a plugin), please refer to the Mattermost Boards Plugin Guide.</p> <p>Getting the engine</p> <p>Binary downloads</p> <p>Official binaries for the Godot editor and the export template.</p> <p>Compiling from source</p> <p>See the official docs for compilation instructions for every supported platform.</p> <p>Note</p> <p>For the Agility SDK's DLLs you have to explicitly choose the kind of workflow. Single-arch is the default (DLLs copied to <code>bin/</code>). If you pass <code>agility_sdk_multi_arch=yes</code> to SCons, you'll opt-in for multi-arch. DLLs will be copied to the appropriate <code>bin/<arch>/</code> subdirectories and at runtime the right one will be loaded.</p>

1-3. Analysis for Documentation Quality Contribution Guidelines

<p>Modes of Contribution</p> <ul style="list-style-type: none">> Participate in Rocket.Chat DevelopmentAssist with Testing Rocket.ChatContribute Through Bug ReportingDocumentation Feedback Guidelines> Security Contribution GuidelinesTranslation Contribution GuidelinesPromote Rocket.ChatContribute to Rocket.Chat FAQs <p>ISSUE_TEMPLATE</p> <ul style="list-style-type: none">bug_report.mdrelease.md  <h2>ROCKET.CHAT</h2>	<h3>Contribution Process</h3> <p>Contributing to an open-source project like Rocket.Chat can be an exciting and fulfilling experience. However, it's crucial to understand the process that your contribution will undergo before it can be integrated into the main codebase. This guide aims to help you comprehend the stages involved in the Rocket.Chat contribution process.</p> <ul style="list-style-type: none">• Stage 1: Forking and cloning: The first step in contributing to Rocket.Chat is to fork the project on GitHub and clone it to your local development environment. This allows you to work on your changes in a controlled environment without impacting the main project.• Stage 2: Create a branch: It's best to create a new branch for each new feature or bug fix. This helps to isolate your changes and makes it easier to track the work that has been done.• Stage 3: Implementing changes: You write your code during this stage. Endeavor to follow Rocket.Chat's coding standards and guidelines to ensure consistency and maintainability. Your code should be clear, well-commented, and tested thoroughly.• Stage 4: Committing changes: Once you've made your changes, you need to commit them. Each commit should be a single logical change, and the commit message should describe what the change does, why it's needed, and any potential implications it may have.• Stage 5: Submitting a pull request: After you've committed your changes, it's time to submit a pull request (PR). The PR should be detailed and include a description of the changes, the reasons behind them, a <code>changeset</code> if the PR is a fix or feature, and any issues that they address. It should also include any necessary documentation changes.• Stage 6: Code review: Now, the Rocket.Chat team will review your PR. This involves checking the code for any potential issues, ensuring it aligns with the project's coding standards and that it accomplishes what it is intended to do. This stage might involve some back-and-forth discussions, so be prepared to answer questions and possibly make revisions based on the feedback you receive.• Stage 7: Acceptance or rejection: Once the review process is completed, the team will decide whether or not to merge your PR into the main codebase. If your PR is accepted, it will be merged. If it's rejected, you'll be given feedback on why this decision was made. It's important to remember that the decision to merge a PR is at the sole discretion of the Rocket.Chat team. <h3>Contributing to TOAST UI</h3> <p>First off, thanks for taking the time to contribute! 🎉 😊 🌟</p> <p>The following is a set of guidelines for contributing to TOAST UI. These are mostly guidelines, not rules. We welcome contributions and are free to propose changes to this document in a pull request.</p> <h4>Reporting Bugs</h4> <p>ISSUE_TEMPLATE</p> <ul style="list-style-type: none">bug_report.mdfeature_request.mdquestion.md  <h2>TOAST UI</h2>
<p>ISSUE_TEMPLATE</p> <ul style="list-style-type: none">bug_report.mddoc_improvement.mdenhancement.md  <h2>Focalboard</h2>	 <h2>GODOT</h2> <p>Communicating with developers</p> <p>The Godot Engine community has many communication channels, some used more for user-level discussions and support, others more for development discussions.</p> <p>To communicate with developers (e.g. to discuss a feature you want to implement or a bug you want to fix), the following channels can be used:</p> <ul style="list-style-type: none">• Godot Contributors Chat: You will find most core developers there, so it's the go-to platform for direct chat about Godot Engine development. Browse the Directory for an overview of public channels focusing on various engine areas which you might be interested in.• Bug tracker: If there is an existing issue about a topic you want to discuss, you can participate directly. If not, you can open a new issue. Please mind the guidelines outlined above for bug reporting.• Feature proposals: To propose a new feature, we have a dedicated issue tracker for that. Don't hesitate to start by talking about your idea on the Godot Contributors Chat to make sure that it makes sense in Godot's context. <p>ISSUE_TEMPLATE</p> <ul style="list-style-type: none">bug-report.mdfeature---enhancement-request.md
<h3>Contribution Checklist</h3> <p>Thanks for your interest in contributing code!</p> <p>Follow this checklist for submitting a pull request (PR):</p> <ol style="list-style-type: none">1. You've signed the Contributor License Agreement, so you can be added to the Mattermost Approved Contributor List.2. Your ticket is a Help Wanted GitHub issue for the project you're contributing to.<ul style="list-style-type: none">◦ If not, follow the process here.3. Your code is thoroughly tested, including appropriate unit tests, and manual testing.4. If applicable, user interface strings are included in the localization file (en.json)<ul style="list-style-type: none">◦ In the webapp folder, run <code>npm run i18n-extract</code> to generate the new/updated strings.5. The PR is submitted against the <code>main</code> branch from your fork.6. The PR title begins with the GitHub Ticket ID (e.g. <code>[GH-394]</code>) and the summary template is filled out.	<p>CONTRIBUTING</p> <ul style="list-style-type: none">How to contributeContribution workflow <p>Contributing code</p> <p>Testing and reporting issues</p> <p>Contributing to the documentation</p> <p>Contributing translations</p> <p>Bisecting regressions</p> <p>Bug triage guidelines</p> <p>Pull request workflow</p> <p>Pull request review process</p> <p>Testing pull requests</p>

1-3. Analysis for Licensing & CLA

<p>Contributor License Agreement</p> <p>To have your contribution accepted you must sign our Contributor License. GitHub will alert you with a new comment asking you to sign and will jning the</p> <p>Please review and sign our CLA at https://cla-assistant.io/RocketChat/Rocket.Chat</p> <p>Version: 2016-11-23</p> <p>MIT License</p> <p>Third-party components → original license of the owner</p>	<p>nhn/tui.editor is licensed under the 0 commits the week of Jan 14</p> <p>MIT License</p> <p>A short and simple permissive license with conditions only requiring preservation of copyright and license notices. Licensed works, modifications, and larger works may be distributed under different terms and without source code.</p> <table border="1"><thead><tr><th>Permissions</th><th>Limitations</th><th>Conditions</th></tr></thead><tbody><tr><td>✓ Commercial use ✓ Modification ✓ Distribution ✓ Private use</td><td>✗ Liability ✗ Warranty</td><td> ⓘ License and copyright notice</td></tr></tbody></table> <p>Follow GitHub's standard PR process</p> <p>ROCKET.CHAT</p> <p>TOAST UI</p> <p>MIT License</p>	Permissions	Limitations	Conditions	✓ Commercial use ✓ Modification ✓ Distribution ✓ Private use	✗ Liability ✗ Warranty	ⓘ License and copyright notice
Permissions	Limitations	Conditions					
✓ Commercial use ✓ Modification ✓ Distribution ✓ Private use	✗ Liability ✗ Warranty	ⓘ License and copyright notice					
<p>MIT License</p> <p>AGPLv3 Free Software <i>Free as in Freedom</i></p> <p>Follow GitHub's standard PR process</p>	<p>godotengine/godot-demo-projects is licensed under the 0 commits the week of Jan 14</p> <p>MIT License</p> <p>A short and simple permissive license with conditions only requiring preservation of copyright and license notices. Licensed works, modifications, and larger works may be distributed under different terms and without source code.</p> <table border="1"><thead><tr><th>Permissions</th><th>Limitations</th><th>Conditions</th></tr></thead><tbody><tr><td>✓ Commercial use ✓ Modification ✓ Distribution ✓ Private use</td><td>✗ Liability ✗ Warranty</td><td> ⓘ License and copyright notice</td></tr></tbody></table> <p>Follow GitHub's standard PR process</p> <p>APACHE HTTP SERVER PROJECT</p> <p>MIT License</p>	Permissions	Limitations	Conditions	✓ Commercial use ✓ Modification ✓ Distribution ✓ Private use	✗ Liability ✗ Warranty	ⓘ License and copyright notice
Permissions	Limitations	Conditions					
✓ Commercial use ✓ Modification ✓ Distribution ✓ Private use	✗ Liability ✗ Warranty	ⓘ License and copyright notice					

2. Project Proposal

① Introduction

② Project Overview

③ Community and Governance

④ Licensing and Commercial Potential

⑤ Initial GitHub Setup

⑥ Github

IT-
MBTI

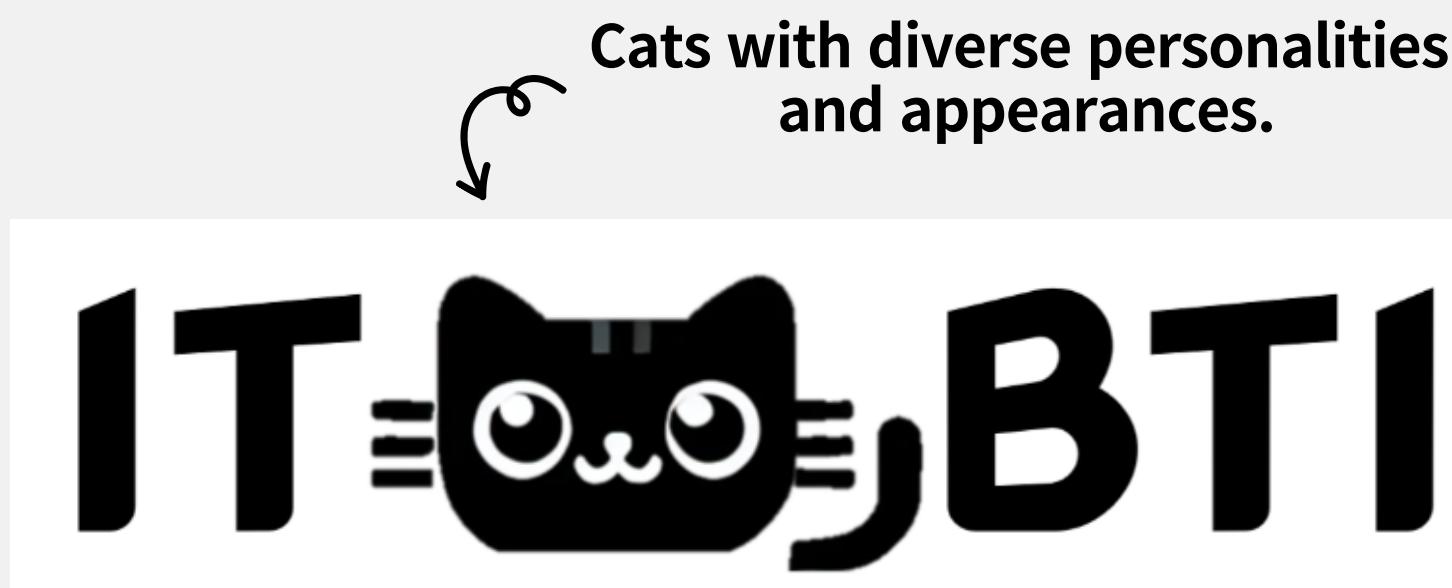
2-1. Introduction

Insights from OSS Community Analysis

Key Insights :

- ✓ Limited engagement with contributors in discussions and issue tracking
 - Community Dynamics
- ✓ Unclear decision-making process
 - Governance Models
- ✓ Lack of roadmap for scaling contributors and feature requests
 - Scalability and Sustainability

2-2. Project Overview



Project Name: **IT-MBTI**

Philosophy

ITM students can explore their unique career traits in an easy and enjoyable way

“Understanding yourself and designing a better future!”

Project Type

Web-based career exploration tool

: Accessible without installation, making collaboration and contribution easier

Tech stack

JAVASCRIPT : standard language for web, dynamic and responsive interfaces

REACT : component-based structure

STYLED-COMPONENTS : modular and dynamic styling,

2-2. Project Overview

The diagram shows a user flow from a start screen to a results screen.

- Start Screen:** A box containing the text "What is your IT-MBTI?" and a "START" button.
- Question 1:** "Q1. Do you like to solve problems with creative ideas?" with a 5-point Likert scale from Agree (0) to Disagree (5).
- Question 2:** "Q1. Do you like to solve problems with creative ideas?" with a 5-point Likert scale from Agree (0) to Disagree (5).
- Question 3:** "Q1. Do you like to solve problems with creative ideas?" with a 5-point Likert scale from Agree (0) to Disagree (5).
- Result Screen:** A box showing the user is a "Master of Problem Solving" and "Developer type". It includes a bar chart comparing developer types across various metrics (e.g., 8%, 8%, 8%, 8%, 8%) and a "I recommend ITM's professor" button.

Features

- Career Aptitude Test:** Evaluate your aptitude for **7 IT career paths** through specific questions.
- Personalized Results:** Recommend the most suitable IT career path based on your score.
- Simple Question Format:** Users respond on a 5-point scale to express their preferences easily.
- User-Friendly Interface:** Designed to be intuitive and easy for everyone to use.

2-3. Community and Governance

Community

- Welcoming New Contributors
- Operating Diverse Community Channels



Slack
Real-time communication



Github-issue
Bug tracking and management

Github-Discussions
Idea sharing and discussion



E-mail
Official inquiries and
feedback collection

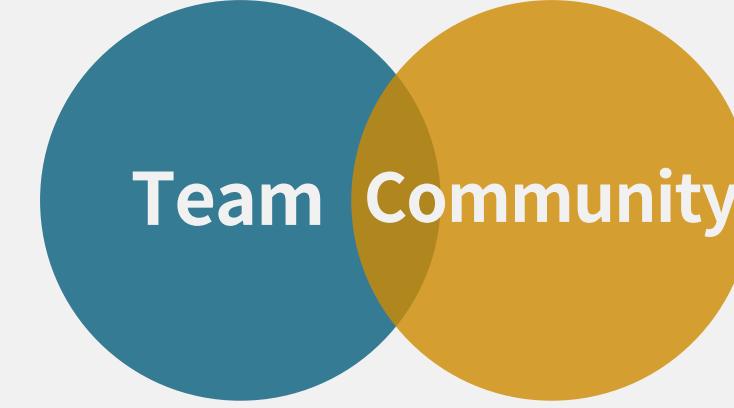
2-3. Community and Governance

Governance



Initial Phase

- 1-6 months
- Detailed Documentation
- Activate GitHub Discussions
- Community Feedback Collection



Intermediate Phase

- 6-12 months
- Expand Community with Early Contributors
- Strengthen Collaboration



Long-Term Phase

- 12+ months
- Transition to Community-Driven Model
- Introduce Optional Voting System
- Assign Leadership Roles (Reviewers, Maintainers)

2-3. Community and Governance

documentation

Contributor Covenant Code of Conduct

Our Pledge

We as contributors and maintainers pledge to make participation in our project and community a harassment-free experience for everyone, regardless of age, body size, disability, ethnicity, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, religion, or sexual identity and orientation.

We commit to acting in ways that contribute to an open, welcoming, diverse, inclusive, and healthy community.

Our Standards

Examples of behavior that contributes to creating a positive environment include:

- Using welcoming and inclusive language.
- Being respectful of differing viewpoints and experiences.
- Gracefully accepting constructive criticism.
- Focusing on what is best for the community.
- Showing empathy towards other community members.

Examples of unacceptable behavior include:

- The use of sexualized language or imagery and unwelcome sexual attention or advances.
- Trolling, insulting, or derogatory comments, and personal or political attacks.
- Public or private harassment.
- Publishing others' private information, such as a physical or electronic address, without explicit permission.
- Other conduct which could reasonably be considered inappropriate in a professional setting.

Enforcement Responsibilities

Project maintainers are responsible for clarifying the standards of acceptable behavior and are expected to take appropriate and fair corrective action in response to any instances of unacceptable behavior.

Project maintainers have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, or to ban temporarily or permanently any contributor for other behaviors that they deem inappropriate, threatening, offensive, or harmful.

Scope

This Code of Conduct applies within all project spaces and also applies when an individual is representing the project or its community in public spaces. Examples include using an official project email address, posting via an official social media account, or acting as an appointed representative at an online or offline event.

Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by contacting the project team at [\[ITMBTI@gmail.com\]](mailto:[ITMBTI@gmail.com]). All complaints will be reviewed and investigated and will result in a response that is deemed necessary and appropriate to the circumstances. The project team is obligated to maintain confidentiality with regard to the reporter of an incident. Further details of specific enforcement policies may be posted separately.

Project maintainers who do not follow or enforce the Code of Conduct in good faith may face temporary or permanent repercussions as determined by other members of the project's leadership.

Attribution

This Code of Conduct is adapted from the [Contributor Covenant](#), version 2.1, available at https://www.contributor-covenant.org/version/2/1/code_of_conduct.html.

For answers to common questions about this Code of Conduct, see <https://www.contributor-covenant.org/faq>.

Contributing to IT-MBTI

We appreciate your interest in contributing to IT-MBTI!

Contributions are what make open-source projects great.

This document will guide you through the process.

CONTRIBUTING.PULL REQUESTS

1. Fork the repository:
Click the "Fork" button on the top-right corner of the repository page on GitHub.
2. Clone your fork:

```
git clone https://github.com/OSS-TeamProject/IT-MBTI.git  
cd it-mbt
```
3. Create a new branch:

```
git checkout -b feature/your-feature-name
```
4. Make your changes:
Implement your feature or fix
5. Commit your changes:

```
git commit -m "Add: Your feature description"
```
6. Push your changes:

```
git push origin feature/your-feature-name
```
7. Submit a Pull Request (PR):
Go to the original repository on GitHub and click "New Pull Request." Select your branch and explain your changes.

WRITING UNIT TESTS

We recommend adding unit tests for any bug fixes or new features:

1. Write tests for both successful and failing cases.
2. Include your tests in the same pull request as your feature or bug fix.
3. Ensure all tests pass before submitting your pull request:

```
npm test
```

Unit tests ensure the project's reliability and help prevent regressions in future updates.

THANK YOU

Thank you for contributing to IT-MBTI! We look forward to collaborating with you to make this project better for everyone.

— The IT-MBTI Development Team

Commit Message Convention

Commit Message Format

<Type>: Short description (fix #1234)

Longer description here if necessary

BREAKING CHANGE: only contain breaking change

- Any line of the commit message cannot be longer than 100 characters!

Revert

revert: commit <short-hash>

This reverts commit <full-hash>

More description if needed

Type

Must be one of the following:

- feat: A new feature
- fix: A bug fix
- docs: Documentation only changes
- style: Changes that do not affect the meaning of the code (white-space, formatting, missing semi-colons, etc)
- refactor: A code change that neither fixes a bug nor adds a feature
- perf: A code change that improves performance
- test: Adding missing or correcting existing tests
- chore: Changes to the build process or auxiliary tools and libraries such as documentation generation

Subject

- use the imperative, present tense: "change" not "changed" nor "changes"
- don't capitalize the first letter
- no dot(.) at the end
- reference GitHub issues at the end. If the commit doesn't completely fix the issue, then use (refs #1234) instead of (fixes #1234).

Body

- use the imperative, present tense: "change" not "changed" nor "changes".
- the motivation for the change and contrast this with previous behavior.

BREAKING CHANGE

- This commit contains breaking change(s).
- start with the word BREAKING CHANGE: with a space or two newlines. The rest of the commit message is then used for this.

his convention is based on [AngularJS](#) and [ESLint](#)

CODE_OF_CONDUCT.md

CONTRIBUTING.md

COMMIT_CONVENTION.md

2-4. Licensing and Commercial



"IT-MBTI will remain non-commercial, free, and open for everyone."

- To ensure free and unrestricted access for all students and communities.
- To uphold the educational purpose of the project.
- To provide freedom of use, modification, and redistribution



MIT License

The MIT License aligns with our commitment to openness and accessibility

"Encourages broader adoption and adaptation by making the project as flexible as possible."

check the [LICENSE](#) file.

<- Our project contains a license file

2-5. Initial GitHub Setup

1. Clone the Repository

```
$ git clone https://github.com/OSS-TeamProjectt/IT-MBTI.git
```



Node.js

: Fast and Efficient Development Environment

npm

: easy management and installation of dependencies required for the IT-MBTI project.

2. Navigate to the Project

```
$ cd IT-MBTI
```

3. Install Dependencies

```
$ npm install
```

4. Build the Project

```
$ npm run build
```

5. Start the Application

```
$ npm run start
```

2-6. Our GitHub

Link ↗

IT-MBTI

IT-MBTI is a tool that analyzes the user's answers based on the MBTI test format and recommends the best career path in the IT field.

What is your IT-MBTI?

IT-MBTI

IT-MBTI

IT-MBTI

IT-MBTI

Table of Contents

- Tech Stack
- Who Is This For
- About
- Features
- Set Up
- Contributing
- License
- Contact

Tech Stack

REACT JAVASCRIPT STYLED-COMPONENTS

Who Is This For

This document is written for:

- Developers who want to contribute to the project.
- Users looking to solve specific problems using this tool.
- Students and researchers who want to learn about the project's features and how to use it.

About

IT-MBTI is a tool designed to help users explore their career aptitude in the IT field. Inspired by the MBTI test format, users answer questions, and based on their scores, the tool recommends the most suitable IT career path. It is a helpful tool for discovering interests and setting career directions in IT.

Features

- Career Aptitude Test: Evaluate your aptitude for 7 IT career paths through specific questions.
- Personalized Results: Recommend the most suitable IT career path based on your score.
- Simple Question Format: Users respond on a 5-point scale to express their preferences easily.
- User-Friendly Interface: Designed to be intuitive and easy for everyone to use.

Set Up

Follow the instructions below to set up the project and run it locally:

```
$ git clone https://github.com/OSS-TeamProjectt/IT-MBTI.git  
$ cd IT-MBTI  
$ npm install  
$ npm run build  
$ npm run start
```

Contributing

We welcome contributions! For more details, check the [CONTRIBUTING.md](#) file and [COMMIT MESSAGE CONVENTION.md](#) file!

License

This project is licensed under the MIT License. For more details, check the [LICENSE](#) file.

Commercial uses

- Business Model: This project does not include a business model aimed at generating revenue.
- License Type: The MIT License allows all uses, including commercial use.
- Corporate Support: This project is not supported or owned by any specific company.

Contact

If you have any questions or suggestions, please contact us at:

- e-mail: ITMBTI@gmail.com
- GitHub Issues: [Issues](#)

OSS-TeamProjectt/IT-MBTI



3 Contributors 0 Issues 1 Discussion 0 Stars 2 Forks

OSS-TeamProjectt/IT-MBTI

Contribute to OSS-TeamProjectt/IT-MBTI development by creating an account on GitHub.

GitHub

Project Introduction
Overview
Table of Contents
Tech Stack
Who Is This For
About
Features
Set Up
Contributing
License
Contact

Q & A



**Thank you
for listening**

