

Masato Ohba (ohbarye) CV

Last updated: 2022.01.16

Japanese version: <https://ohbarye.github.io/ja/cv/>

Contact

key	value
Name	Masato Ohba (大庭 直人)
ID	ohbarye
Email	over.rye+jh [at] gmail.com
Entrypoint	https://ohbarye.github.io

Summary

The core of my strength is web backend. I have experience in full-cycle development for both in-house development and client work. I can be the overall lead in the team from development to operation, starting with the requirements and requirements definition, which is done with timely communication.

I have been developing SPA and MPA with React in the System of Engagement areas such as registration leads and retention measures, and in the Web frontend area for services that require rich experiences.

I can develop complex business logic required in the System of Record fields, such as payment systems and banking systems, with quality, speed, and robustness in mind.

I also served as an engineering manager for an organization with 20-60 engineers. As a result, I have a proven track record and knowledge of team design, project management, recruitment, corporate branding, culture building, community activities, and more.

Skills

- Language
 - Ruby (2.3~3.x), TypeScript (3.4~4.x), Go (1.15~), JavaScript (ES2015~), Java * (~8), Python (~3.7)*, OCaml *

- Frontend
 - React (16.x~), React Admin, Webpack, Redux*, Backbone.js*, jQuery* (1.x~), React Native (0.5x)*
- Backend
 - Ruby on Rails (4.2~6.1), Node.js*, OpenAPI (3.0), REST, GraphQL*
- Testing
 - RSpec, Capybara, minitest, Jest, React Testing Library, Cypress*, Sinon*, Chai*, Mocha*
- Database
 - MySQL (5.7), PostgreSQL (9.5), MongoDB (3.2~3.6)*, Redis
- Infrastructure
 - AWS, Docker, Kubernetes*, Heroku*, Deis*, Netlify*, Vercel*
- Communication tools
 - GitHub, ZenHub, Slack
- SaaS
 - GitHub Actions, CircleCI, Athena, NewRelic, Google BigQuery, Datadog*
- Development framework
 - Agile, Scrum, Waterfall*

Bold indicates core skills.

* denotes a skill that has no work experience, or has work experience but is more than a few years away.

Strength

Design and implementation in consideration of operation and performance

I have experience in designing and implementing designs that not only temporarily meet business requirements, but are also aware of performance, computational complexity, and data growth over time. In addition, I have performed performance tuning for problems that happened during the operation.

Batch processing, retry, data integrity in payment systems, and how developed functions to be operated. I'm confident in designing, developing and reviewing in consideration of these.

Leverage and improve OSS

Although it's nothing special in this day and age, I use OSS in work. When I select a technology, I focus on whether the technology is feasible as a team, as well as convenience and trends, and I also consider sustainability and

disposability. I often make conservative choices as a result of prioritizing the productivity of a team. I've been using SaaS like dependabot for libraries that my product depends on, and I've been upgrading the dependencies periodically

In addition, I believe that products I develop and OSS are connected to each other. If OSS doesn't have a necessary feature, I try to develop it by myself, send patches, and fix bugs and regressions when I notice them. (OSS contribution by me is also described in the [Personal Projects](#) and [Public Output](#) sections below.

Team-oriented

I have a keen interest in achieving results as a team and have worked to create mechanisms and foster a culture where collaboration can occur. As a manager, I also have experience of determining team assignments by considering variables such as strengths, weaknesses, orientation, and business characteristics.

In the past, when my team has had process issues, I have implemented Scrum to "clarify goals", "be accountable", and "prevent overwork", and I have even taken on the role of a scrum master.

Problem solving in engineering organizations of up to 60 people

As an Engineering Manager in an organization with 20-60 engineers, I have experience in organizational design, team design, recruiting, mentoring, onboarding, culture building, and community activities.

Issues that arise when a team or organization grows in size are often the first ones experienced in that organization. Therefore, without clear identification of the true cause and prioritization and assignment, it is often left unchecked and distortions are magnified. I believe that my experience in knowing and dealing with these characteristics of organizational issues can be applied to teams and organizations of the same size or dozens of people, whether my role is as a member or manager.

Output-oriented

I try to summarize what I've researched and learned through my hobbies and work and output it. Repeating output-conscious contemplation and trials can hone my skills in abstraction and commonalization. Moreover, I believe that it is possible to enhance the overall ability to find and solve problems.

I'm not very good at having presentations. However, I challenge it as possible as I can for feedbacks I'd get and things I'd learn through those opportunities. Please refer to the [Public Output](#) section for the results.

Weakness / I've never done that before.

I can play to the above strengths in product development with monolithic applications on the backend and in teams of a few to 20 engineers, but I have little or no experience with the followings.

- Microservices, distributed systems
- Huge scale traffic
- Mobile App Development
- Architecture design and infrastructure as code (Terraform, Ansible, etc.), and using Cloud services (AWS, GCP, Azure, etc.)
- Selection, operation, and tuning of middlewares.
- Hardware and embedded software
- Development in a large team of over 20 people

I am motivated to learn in areas where I have no experience. I think it's possible because I've done some catch-up in the past on the technology needed for the job.

Work Experience

SmartBank, Inc. (2020.08 ~)

[SmartBank, Inc.](#) is a BtoC Fintech company that is categorized as a [challenger bank](#) like [Monzo](#). The company is an "Issuer" that issues prepaid cards, and develops and operates a product named [B/43](#) that visualizes and automates spending management in conjunction with payment with cards.

The main achievements of the company are described below.

- Developed various features for users.
 - Withdrawal / Money transfer / Account by purpose / Joint account / Spending management
- Developed an in-house system that performs the necessary work as an issuer.
 - Identity verification (eKYC) / Card delivery system
- Developed tools and systems for internal developers.
 - Introduced [Feature Toggles](#) and separated development and release.
 - Made APIs compatible with [Idempotency-Key Headers](#) and make it robust so that data can be protected even if multiple requests occur.
- Contributed to development productivity.
 - Built CI/CD pipeline and speed up when any bottleneck appear.
 - Designed and built an automated deployment/release flow.
 - Designed and built schema migration and data migration flow in production environment.

- Contributed to recruitment activities.

Quipper Limited (2015.09 ~ 2020.05) 4 yrs 8mos

[Quipper](#) is a company that operates both B to C and B to B education businesses. We develop and operate Quipper School and Quipper Video outside of Japan, and [StudySuppli](#) in Japan.

Under the umbrella of the Study Supplements brand, there are courses for elementary, junior high, and high schools, English for working adults, and forSchool for schools and municipalities. Some of the projects I was involved in are listed below.

Title	Period	Appeal Points				
Code cleanup	2020.03	Toughness to remove huge amount of codes				
Price Change	2019.10~2020.01	Payment systems, complex business logics				
Migration from React Native to PWA	2019.07~2019.09	Technology selection, paying back technology debt				
Develop Coaching Service for Junior High School Students	2018.08~2019.03	9 people x 7 months project management / Scrum				
Entry Form Renewal	2018.03	Technology selection, EFO				
Upgrade grape	2017.10~2017.12	Paying back technical		Research to Abolish	2017.10	A/B testing, Data driven

gem Title	Period	debt, OSS Appeal Points		Payment Method		decision making
Develop Coaching Service for High School Students	2016.12~2017.02	Short term development				
Benkyo Sapuri Migration	2016.06~2016.12	Short term development				
Hiring	2016.07~2020.03	Organization design				
Develope In-App Purchase for iOS	2016.04~2016.06	Payment systems, OSS				
Jyukun Sapuri Migration	2015.09~2016.02	Short term development, catch-up tech stacks				
Other Activities	-	Continuous improvement				

Code Cleanup

Because our team's monorepo, which used to share code across multiple products, has been forked, a lot of our code base got unused. I thought that cleaning up those dead codes would bring better development experience, so I almost single-handedly suggested and executed code cleanup.

It took about three weeks to remove 400,000 lines of code, and the removal of unused models in Rails was able to reduce the number from 390 to 281. I made significant changes to the system in operation, but completed them without causing a failure that would affect the end-users.

Period	Team	Role	Tech stacks, tools
2020.03	1 (Web×1)	Web	Ruby, Rails, React, Redux, TypeScript (Deletion Targets) NewRelic, BigQuery (Usage research)

Migration from React Native to PWA

This project is a renewal of an iOS/Android business internal app made by React Native with PWA. I was primarily responsible for the launch and backend of the project. (The UI/UX was revamped at the time of the relaunch, so the backend had to be changed as well.)

Since there was no mechanism to share the API specification before the project, I wrote the API specification using OpenAPI and laid the groundwork for sharing it. And leveraging Rack middleware, which only works in an automated testing and staging environment, I also put in place a mechanism to detect specification violations. (I wrote more about it in a [blog post](#).)

I also [made a presentation at JSConf 2019](#) as a recap of this project.

Period	Team	Role	Tech stacks, tools
2019.07~2019.09	6 (PdM×1, Designer×2, iOS×1, Frontend×1, Backend×1)	Backend	Ruby, Rails 5.0, RSpec, React 16.12, Redux 4.0, TypeScript 3.7, Cypress

Develop Coaching Service for Junior High School Students

In a highly uncertain project with over 10 new members in the majority and a size of over six months, I was able to release it by implementing and practicing practices such as Scrum, MobPro, 1-on-1, etc. And while it's difficult to quantitatively evaluate it, it has led to the growth and retention of our members.

I have summarized the details of this project in “[How to Combat Uncertainty in a Large Project with Many New Members]” (<https://quipper.hatenablog.com/entry/2019/06/27/how-to-defeat-uncertainty>).

Period	Team	Role	Tech stacks, tools
2018.08~2019.03	12 (PdM×1、Designer×1、iOS×1、Android×1、Web×8)	Web, Scrum master	Ruby, Rails, React, RSpec, Capybara, Redux, TypeScript, Jest, React Native

Entry Form Renewal

It is an EFO (Entry Form Optimization) project for the most important acquisition period in BtoC learning services.

- I was responsible for both the front end and back end.
- At that time, the majority of the team's front-end work was MPA, and the

entry form was implemented in Rails View and jQuery.

- I selected Webpacker and React.
- CVR of registration has been improved.

Period	Team	Role	Tech stacks, tools
2018.03	3 (PdM×1, Designer×1, Web×1)	Web	Ruby, Rails, React, RSpec, Capybara, Redux, TypeScript, Webpacker

Upgrade grape gem

The biggest Rails API server in my company had been using a gem named grape, but the API hadn't been updated since v0.12.0 in 2015 when it was introduced, so I upgraded it.

Compared to Rails upgrade, there is less information, and dependent libraries are sometimes dead, so I had to prepare a workaround to write a patch. Finally, I was able to upgrade to the latest version v1.0.1 at that time by gradually increasing the minor version (~v0.19).

As an aside, about a year after this, in December 2018, I had an issue with NewRelic not being able to get metrics for the Rails API. This was a problem on the NewRelic agent side, so I sent a [patch](#) to the upstream to solve it.

Period	Team	Role	Tech stacks, tools
2017.10~2017.12	1 (Web×1)	Web	Rails 4.2, Ruby 2.4, RSpec, grape

Research to Abolish Payment Method

This project examines how much of an impact it will have on the business when it abolishes carrier payments, which have high fees and operating costs.

Using Optimizely SaaS, we were able to practice A/B testing and collect quantitative data while minimizing the amount of development and testing done by engineers.

I've written a [blog post](#) about this effort, and we've presented at events such as [Rails Developer Meetup 2018 Day 3](#) and [Regional Scrum Gathering Tokyo 2019](#).

Period	Team	Role	Tech stacks, tools
2017.10	2 (PdM×1, Web×1)	Web	Rails 4.2, Ruby 2.4, jQuery, Optimizely

Hiring

I made a full contribution, including document screening, first round, code testing, second round, casual interviews, and proactive participation in the referral lunch and referral dinner.

- Wrote job descriptions
- Introduced and polished structured interviews
- [Designed code test](#)
- Designed technical interviews
- Published an interview guide
- Community activities
 - [StudySapuri Meetup #1](#)

I also engaged in the following activities, which are closely related to our recruitment activities.

- Onboarding
- Wrote blog posts summarizing the findings of the above activities
 - [Casual interviews](#)
 - [Interview guide](#)
 - [Alumni meetup](#)
 - [Offboarding](#)

Period	Team	Role	Tech stacks, tools
2016.07~2020.03	4~10 (Webx4~10)	Web	Rails, Ruby, minitest, React, TypeScript (For coding test)

Develope In-App Purchase for iOS

Our service was offering outright purchase feature within iOS app, but sales were not good, and there was a need to provide an auto-renewable payment function like credit cards.

I implemented server-side APIs and a batch process to check subscription statuses. During the development, I realized that [venice](#), AppStore API client library used on the server side, didn't support the automatic renewal, so I implemented it myself and sent a [pull request](#). (The OSS is apparently short-handed, so I did [several contributions](#) even after that.)

I gave a [presentation at iOSDC 2018](#) about the server side implementation of In-App Purchase, including the operation for that.

Period	Team	Role	Tech stacks, tools
2016.04~2016.06	3 (PdM×1、iOS×1、Web×1)	Web	Rails 4.2, Ruby 2.3, RSpec, venice, AppStore

Jyuken Sapuri Migration

This is a project to rebrand the “Jyuken Sapuri” (Learning platform for high school students that our parent company had used to run) on Quipper’s codebase as a “StudySapuri”.

There were not many technical difficulties, but it was insane project. We had to build a system that had originally been developed and operated for several years from scratch in six months.

I was in charge of a registration flow, batches to exchange data with external systems, student management functions for teachers, and automated homework delivery functions. Although I had none of experiences of tech stacks used in this project in the past, I was able to catch up in a short period and to finish my tasks.

Press release on this project:http://www.recruit-mp.co.jp/news/release/2016/0225_2893.html

Period	Team	Role	Tech stacks, tools
2015.09~2016.02	10 (PdM×1、iOS×2、Android×2、Web×5)	Web	Rails 4.2, Ruby 2.3, RSpec, Capybara, Backbone.js, Marionette.js, Sion, Mocha, Chai

Other Activities

The followings are other regular and spot tasks I’ve addressed.

- [Creating a Collaborative Culture](#)
- [Promote in-house study abroad](#)
- Introduce tools to improve development efficiency ([kpt-bot](#), [review-bot](#), [Pull Panda](#))
- Dependencies upgrade with [dependabot](#)
- Release manager
- Quipper School / Video, study supplements release separation
- Introduction of a pull-up machine
- Organizing reading and study groups (Real World HTTP, SRE books, etc.)

SCSK Corporation (2012.04 ~ 2015.08) 3 yrs 5mos

SCSK is a Japanese big information technology services company.

I was a member of a department developing some enterprise systems for real estate companies which needs better property management solutions. I've developed web applications for the purpose.

- Web backend: Java6, Seasar2 (an open source web application framework)
- Web frontend: jQuery
- Web Server: Apache + Tomcat
- DB: Oracle Database 11g

Public Output

Blogs

- Personal (Japanese) <https://ohbarye.hatenablog.jp/>
- Personal (English) <https://dev.to/ohbarye>

Articles on Company's Blog

- [TracePoint Usage: Sentry Captures Local Variable](#)
- [Retry with Idempotency-Key Header](#)
- [Project Management Technique to Overcome Uncertainty](#)
- [Working Out Loud -How to Make Collaborative Culture-](#)
- [Quipper Interview Guide is Published](#)
- [Introduction of Casual Interview](#)
- [Fact-based Decision Making with A/B Testing](#)
- [Effective Onboarding](#)
- [Timezone in Global Applications](#)

Presentations

Here are a few excerpts from <https://speakerdeck.com/ohbarye>. All of them are in Japanese.

- [Safe Retry with Idempotency-Key Header](#)
- [Migration from React Native to PWA\(JSConf JP 2019\)](#)
- [Lightning Talk Session Organizer](#)(Developers Summit 2019 Summer)
- [Verify Hypothesis with A/B Testing](#)(Regional Scrum Gathering Tokyo 2019)
- [How to Find Good First Issues](#)(NodeFest JP 2018)
- [How to Manage Payment Transactions](#)(Meguro.rb#19)
- [Rethink In-App Purchase on Subscription Services](#)(iOSDC Japan 2018)

- [Fact-based Decision Making and Development](#)(Rails developer Meetup 2018 Day 3 Extreme)
- [Color Accessibility, which Engineers Should Care](#)(Roppongi.js#3)
- [Level up Frontend Skills for Rails Developer](#)(Meguro.rb#15)

OSS Activities

- Pull requests to OSS ([pull requests list](#))
 - An API client gem `venice` of AppStore for In-App purchase didn't support Auto-Renewable, so I developed the feature by myself [pull request](#).
 - A gem `newrelic_rpm` of New Relic Ruby agent was not compatible with grape v1.2.0+, so I fixed with this [pull request](#).
 - I have also sent patches to major OSS such as Node.js, yarn, Ruby, etc.
- Tools
 - [goofi](#)
 - [review-waiting-list-bot](#)
 - [kpt-bot](#)
 - [markdown-server](#)
- [Published gems](#)
- [Published npm packages](#)

Community Activities

- [Engineering Manager Meetup](#)
 - Role: Meetup organizer
 - The purpose of this community is to share knowledge about engineering management, which is still not widely available in Japan.
 - In total, more than 500 people participated in the event.
 - I've been running it on my own for about 1.5 years.
 - I handed over the community management to the community.
- [EOF2019](#)
 - Role: core member
 - The very first festival that puts "Engineering Management" on the theme in Japan.
 - Organize the OST session.
- Meguro.rb (Ruby regional community)
 - Role: member
 - As an administrative staff member of Meguro.rb, one of the local Ruby communities, I hosted the event twice.

Personal Projects

These are artifacts of my personal projects.

Goofi

It is a tool to make contribution to OSS easier. I thought that the biggest obstacle for beginners who want to start OSS contributions is to find the target of contribution, so I created a web service [Goofi](#) to list repositories and issues to which contribution is recommended.

At Nodefest 2018, [I made a presentation about the service](#)

In January 2020, [GitHub has published an official features](#) similar to Googi, then I took it as, “My recognition of the problem was right.

Period	Repository	Tech stacks, tools
2020.08~	https://github.com/ohbarye/goofi	Node.js 13.x, TypeScript 3.8, Next.js 9.1, GraphQL (client), Now.sh v1~2 (現 Vercel), GitHub API v4

React Use Kana

This is a library to create a form that automatically inputs furigana when you type kanji. there are some libraries that have the same function with jQuery and similar libraries for older versions of React, but I couldn't find a library that uses React hooks, so I made my own.

Period	Repository	Tech stacks, tools
2019	https://github.com/ohbarye/react-use-kana	React 16.9, TypeScript 3.8.3

String Pixelater

A library that converts a character to a two-dimensional array of N * N dots. I was addicted to [Processing](#) at the time and implemented it through the process of creating Generative Art.

Period	Repository	Tech stacks, tools
2018	https://github.com/ohbarye/string-pixelater	TypeScript 3.6.4, rollup 1.23

Review bot

It's a Slack bot that encourages reviews in team development. It posts a list of pull requests to Slack that match specified conditions.

It's similar to [Pull Panda](#), acquired by GitHub in June 2019. But my bot was forked by some companies for their internal use, that's probably because Pull Panda was not free.

Period	Repository	Tech stacks, tools
2017.08~	https://github.com/ohbarye/review-waiting-list-bot	Node.js 8.x, GraphQL (client) Slack API, GitHub API v3~4, Heroku

Automaildoc

This is a library that automatically generates HTML to preview email text when running tests with RSpec.

Period	Repository	Tech stacks, tools
2017	https://github.com/ohbarye/automaildoc	Ruby 2.4, RSpec 3

Markdown Server

This is a library that can build a web server to deliver Markdown documents as HTML. I made this for learning Python.

Period	Repository	Tech stacks, tools
2015	https://github.com/ohbarye/markdown-server	Python 3.7, bottle 0.12

Education

- Keio University, Bachelor's degree, European History (2012)

Qualifications

-
- AtCoder
 - (2020.06) Practical Algorithm Skill Test, Advanced Beginner
 - Information-technology Promotion Agency, Japan
 - (2014.06) Database Specialist
 - (2012.12) Applied Information Technology Engineer
 - (2012.06) Fundamental Information Technology Engineer
 - English
 - (2013.02) TOEIC 720
 - Others
 - (2012.03) Mastered Teacher Training Course
 - (2010.12) Chief of Money Lending Operations

Honors & Awards

- (2016.04) MVP2016 in Quipper
- (2015.03) Champion in hackathon conducted by SCSK OSS promotion office

Interests

By 2020, I have developed the technical skills required in the field of software engineering, especially in the application layer of web services development. In the future, I hope to deepen my general understanding of software through learning about whole computer science, algorithms, low layers and middleware in particular.

- Competitive programming
 - AtCoder green <https://atcoder.jp/users/ohbarye>
 - Google Code Jam 2020
 - Passed Qualification Round
- Computer system
 - “[The Elements of Computing Systems: Building a Modern Computer from First Principles](#)” (a.k.a. Nand to Tetris) <https://github.com/ohbarye/nand2tetris>
 - Built an assembler, VM translator, and a compiler in OCaml and Ruby.
- Algorithms
 - Mastered [Princeton University Algorithms Part 1](#) in Coursera
- Database
 - [SQLite clone DB tutorial](#)(Japanese)
- My study note <https://scrapbox.io/ohbarye/>

