Masato Ohba (ohbarye) CV

最終更新日: 2023.08.06

English version: https://ohbarye.github.io/en/cv/

基本情報と連絡先

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

職務要約

Web backend を強みの中心に置きつつ、自社サービスとクライアントワーク両方のフルサイクル開発の経験があります。適時コミュニケーションを取りつつ行う要求・要件定義を始めとし、チームでの開発から運用までの全体のリードを担うことができます。

登録導線やリテンション施策などの System of Engagement 領域、リッチな体験が求められるサービスの Web frontend においては React による SPA, MPA の開発を行なってきました。

決済や銀行システムなど System of Record 領域において求められる複雑なビジネスロジックについても、質・速度・堅牢さを意識した開発が可能です。

また、エンジニア数が 20-60 名規模の組織における Engineering Manager を務め、チーム設計・プロジェクトマネジメント・採用活動・コーポレートブランディング・文化づくり・コミュニティ活動等々に関する実績と知見があります。

スキル

- Language
  - Ruby (2.3-3.x), TypeScript (3.4-5.x), Go (1.15-), JavaScript (ES2015-)
- Frontend
  - React (16.x-), React Admin
- Backend

    - Ruby on Rails (4.2-7.x), OpenAPI (3.x), REST
- Testing
    - RSpec, Capybara, minitest, Jest, React Testing Library
- Database
    - MySQL (5.7-8.0), PostgreSQL (9.5-), MongoDB (3.2-3.6), Redis
- Infrastructure
    - AWS, Docker
- Communication tools
    - GitHub, ZenHub, Slack
- SaaS
    - GitHub Actions, CircleCI, Athena, NewRelic, Google BigQuery
- Development framework
    - Agile, Scrum

太字 はコアなスキルを表現します。

長所 / 強み

運用、パフォーマンスを意識した設計・実装

ビジネスの要求を一時的に満たすだけでなく、パフォーマンス・計算量・将来にわたるデータ増加を意識した設計や実装の経験があります。加えて、運用中に生じた問題についてのパフォーマンスチューニングを行なってきました。

バッチの冪等性・リトライ処理。決済におけるデータの整合性。開発した機能がどのように運用されるか。これらを意識した設計・開発・レビューに自信があります。

OSS を活用、改善する技術力

現代においては特筆すべきことではありませんが、業務においては OSS を利用しています。技術選定の際には利便性やトレンドの最大瞬間風速だけでなくチームとして feasible な技術かどうか？を中心に検討し、並んで持続性や disposability を検討します。チームの生産性を優先した結果として保守的な選択をすることが多いです。製品の依存するライブラリ等については dependabot のような SaaS を活用しつつ dependencies の定期的なアップグレードも行なってきました。

また、開発するプロダクトと OSS は地続きであると考えています。必要な機能がなければ自ら開発したりパッチを送ったり、bug や regression に気づいたら直す振る舞いを心がけています。（後述するPersonal Projects欄、Public Output欄にてもOSS contribution に関する内容を記述しています）

チーム志向

チームとして成果を上げることに強い関心があり、協働が生まれる仕組みづくりや

文化の醸成を行ってきました。また、マネジャーとしてチームメンバー間の信頼関係の構築・強み・弱み・指向性・事業特性などの変数を考慮してチーム編成を決定した経験もあります。

プロセスに問題があったとき、「ゴールを明確にする」「説明責任を果たす」「オーバーワークを防ぐ」ためにスクラムを導入し、スクラムマスターの役割を担ったこともあります。

-60 名規模のエンジニア組織における課題解決

エンジニア数が 20-60 名規模の組織における Engineering Manager として、組織設計・チーム設計・採用活動・教育・オンボーディング・文化づくり・コミュニティ活動等々に関する実績と知見があります。

チーム・組織規模が大きくなる中で生じる課題はその組織において初めて経験される類のものが多く、明確に真因の特定と優先度判断とアサインを行わないと放置され歪みが拡大することがしばしばあります。組織課題のそうした特性を知ったうえで対応してきた経験は、自身の役割がメンバーであれマネジャーであれ、同規模 ± 数十名のチーム・組織でも活かせると考えます。

アウトプット志向

趣味や業務を通じて調べたこと・学んだことをまとめてアウトプットすることを心がけています。アウトプットを意識した思索や試行を繰り返すことによって抽象化・共通化のスキルを磨けるだけでなく、課題の発見から解決までの総合力を高められると考えています。

登壇発表・プレゼンテーションの類は得意ではありませんが、その機会を通じて学ぶことや得られるフィードバックがあるため、可能な範囲で挑戦しています。実績についてはPublic Output欄をご参照ください。

経験のないこと

モノリシックなアプリケーションをバックエンドとするプロダクト開発や数名〜20名弱のチームにおいては上記のような強みを発揮できますが、以下の経験については乏しい、または全く経験がありません。

- Microservices、分散処理
- 大規模トラフィックへの対応
- モバイルアプリ開発
- Cloud services (AWS, GCP, Azure etc.) を活用したアーキテクチャ設計や構成管理 (Terraform, Ansible etc.)
- ミドルウェアの技術選定、運用、チューニング
- ハードウェア、組み込みソフトウェア関連

経験のない領域を学ぶ意欲はあり、業務上必要な技術に関するキャッチアップは過

去に行なってきたように可能だと考えていますが、業務開始直後に初速を出すのは難しいと考えます。

職務経歴

SmartBank, Inc. (2020.08 -)

SmartBank, Inc. は BtoC の Fintech company です。同社はプリペイドカードを発行するイシュアであり、カードでの決済と連動して支出管理を可視化・自動化する B/43 というプロダクトの開発・運用を行っています。

同社における主要な成果を以下に記します。

- サブスクリプションサービスB/43プラスの立ち上げ
  - 他社プラットフォームの調査等を通じて仕様検討
  - 課金機能を含む複数機能の開発、リリース後のグロース
  - 課金機能については発表を参照
- 3Dセキュア対応
  - Kaigi on Rails 2022にて発表を参照
- 対ユーザー向けの各種機能の開発
  - 出金 / 送金 / 目的別口座 / ペア口座 / 支出管理
- イシュアとして必要な業務を行う社内システムの開発
  - 本人確認 (eKYC) 機能 / カード配送機能
- 開発者向けツール・機構の整備
  - Feature Toggles を導入し、デプロイとリリースを分離
  - API を Idempotency-Key Headers に対応させ、多重リクエストが発生してもデータを保護できるよう堅牢化
- 開発生産性に寄与する活動
  - CI/CD の整備・高速化
  - デプロイフローの整備・自動化
  - 本番環境でのスキーママイグレーション、データマイグレーションフローの整備
    - Delayed JobからSQSへの移行
- 採用広報・採用への貢献

Quipper Limited (2015.09 - 2020.05) 4 yrs 8mos

Quipperは BtoC, BtoB 両方の教育事業を営む企業です。日本国外には Quipper School, Quipper Video を、日本国内においてはスタディサプリの開発・運用を行っています。

スタディサプリブランド傘下には小中高大講座、社会人向けの ENGLISH、学校・自治体向けの forSchool 等が存在し、私は主に BtoC のスタディサプリ小中高大講座

の開発・運用に携わりました。関わった Projects のうちいくつかを以下に記します。

タイトル	期間	語れるポイント
Code Cleanup	2020.03	腕力
価格変更対応	2019.10-2020.01	複雑な要件の設計・実装
Migration from React Native to PWA	2019.07-2019.09	技術選定、技術的負債返却
中学生向けコーチングサービス開発	2018.08-2019.03	9 名 × 7 ヶ月規模のプロジェクトマネジメント / スクラム
登録導線リニューアル	2018.03	技術選定、EFO
Upgrade grape gem	2017.10-2017.12	技術的負債返却、OSS
キャリア決済廃止検討	2017.10	A/B テスト、Data driven な意思決定
高校生向けコーチングサービス開発	2016.12-2017.02	短納期開発
勉強サプリ移管プロジェクト	2016.06-2016.12	短納期開発
採用活動	2016.07-2020.03	組織づくり
In-App Purchase 機能実装	2016.04-2016.06	決済機能開発、OSS
その他	-	地道な改善活動

Code Cleanup

複数プロダクトでコードを共用していた monorepo を fork し、dead code が大量に発生しました。その dead code の掃除が全体の開発体験に関わると考え、退職前の最後の仕事としてほぼ独力で提案から実行まで行いました。

約 3 週間で 400,000 行のコードを削除するに至り、Rails の利用されていない model の削減では数を 390 から 281 に減らすことができました。稼働中のシステムに大きく手を入れたものの顧客に影響を及ぼすような障害を起こすことなく完遂しました。

期間	チーム構成	役割	利用技術、ツール
2020.03	1 名 (Web×1)	Web	Ruby, Rails, React, Redux, TypeScript (削除対象) NewRelic, BigQuery (機能の利用状況調査)

Migration from React Native to PWA

React Native 製 iOS/Android の業務用 internal アプリを PWA にてリニューアルするプロジェクトです。主にプロジェクトの立ち上げとバックエンドを担当しました。（リニューアル時に UI/UX を刷新したため、バックエンドにも変更が必要でした）

プロジェクト以前に API に関する仕様を共有する仕組みがなかったため、OpenAPI による API specification の記述をし、共有のための土台の整備も行いました。自動テストと Staging 環境においてのみ動作する Rack middleware を活用し、Specification 違反を検知する仕組みも導入しています。（詳細はブログ記事にも書きました）

また、プロジェクトの総括をJSConf 2019 にて発表しました。

期間	チーム構成	役割	利用技術、ツール
2019.07- 2019.09	6 名 (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

中学生向けコーチングサービス開発

新規メンバーが大半の 10 名超、かつ半年を超える規模である不確実性の大きいプロジェクトにおいて、スクラム・モブプロ・1-on-1 等々のプラクティスを導入・実践することによりにリリースすることができました。また、定量的評価が難しいものの、メンバーの成長・定着にも繋がりました。

本プロジェクトの詳細は『新メンバーが多い大型プロジェクトでの不確実性との戦い方』にまとめています。

期間	チーム構成	役割	利用技術、ツール
2018.08- 2019.03	12 名 (PdM×1、 Designer×1、iOS×1、 Android×1、Web×8)	Web, Scrum master	Ruby, Rails, React, RSpec, Capybara, Redux, TypeScript, Jest, React Native

登録導線リニューアル

BtoC の学習サービスにおいて最も重要な獲得の時期に向けた EFO (Entry Form Optimization) プロジェクトです。

当時のチームにて扱うフロントエンドの大部分は MPA であり、登録導線は Rails の View と jQuery で実装されていました。リッチな体験が要求されるシーンであったため Webpacker を用いて Rails フロントエンドの一部で React を導入しました。

同時期でなくリリース前後の比較ではありますが登録導線の CVR が有意に向上し、事業成果に直結する取り組みとなりました。

Meguro.rb にて取り組みの内容を発表しました。

期間	チーム構成	役割	利用技術、ツール
2018.03	3 名 (PdM×1, Designer×1, Web×1)	Web	Ruby, Rails, React, RSpec, Capybara, Redux, TypeScript, Webpacker

Upgrade grape gem

社内で最も大きい Rails の API server ではgrape という gem を使っていましたが、この gem が 2015 年の導入時の v0.12.0 から更新されていなかったのでアップグレードしました。

Rails upgrade 等に比べて情報が少なく、依存している周辺 library が dead であるために迂回策を用意するなどの工夫が必要でした。minor version (-v0.19) を少しずつ上げながら最終的に当時の最新の v1.0.1 まで upgrade できました。

余談ながら、この約 1 年後 2018 年 12 月に NewRelic で同 Rails アプリケーションのメトリクスが取れないという問題が起きました。これは NewRelic agent 側の問題だったため本体にパッチを投げて解決しました。

期間	チーム構成	役割	利用技術、ツール
2017.10-2017.12	1 名 (Web×1)	Web	Rails 4.2, Ruby 2.4, RSpec, grape

キャリア決済廃止検討

手数料が高く運用コストも高いキャリア決済を廃止する際に、どれぐらい事業へのインパクトがあるのかを検証するプロジェクトです。

Optimizely という SaaS を用い、エンジニアによる開発やテスト実施の手間を極力抑えつつ A/B testing の実践と定量データの収集ができました。

この取組についてはブログ記事を執筆し、Rails Developer Meetup 2018 Day 3やRegional Scrum Gathering Tokyo 2019といったイベントでの登壇発表も行いました。

期間	チーム構成	役割	利用技術、ツール
2017.10	2 名 (PdM×1, Web×1)	Web	Rails 4.2, Ruby 2.4, jQuery, Optimizely

採用活動

書類選考・一次・コードテスト・二次・カジュアル面談・リファラルランチ・リファラルディナーへの主体的な参加を始めとし、全面的に貢献しました。

- Job description の執筆
  - HR やエージェントと連携しつつ作成
  - 採用状況の変化に合わせて 2 度書き直しました
- 構造化面接の採用・実践・型化
- コードテストの設計と実装
- 技術面接の設計と実装
- 面接ガイドの公開
- コミュニティ活動
  - スタディサプリミートアップ第一回の運営

また、採用活動と地続きである以下の活動にも取り組みました。

- オンボーディング
- 上記活動に関する知見をまとめたブログ記事の執筆
  - カジュアル面談
  - 面接ガイドの公開
  - 卒業生と在職者の交流会の開催
  - オフボーディング

期間	チーム構成	役割	利用技術、ツール
2016.07-2020.03	4-10 名 (Web×4-10)	Web	Rails, Ruby, minitest, React, TypeScript (作問にて使用)

In-App Purchase 機能実装

当時、iOS アプリ内での課金手段として買い切りを提供していたものの売上が芳しく無く、クレカ等と同じように自動更新（Auto-renewable）による決済機能を提供したいというニーズがありました。

私はサーバサイドの API と、subscription status を確認するバッチ処理を実装しました。その過程でサーバサイドで利用している AppStore API client library の venice が自動更新の形式に対応していなかったと気づいたため、自前で実装し、pull requestを送りました。（同 OSS はどうやら手が足りていないようなのでそのあとも何度か contributionしました）

このあとの運用も含めて In-App Purchase のサーバサイドに関する知見が得られたのでiOSDC 2018 で登壇発表を行いました。

期間	チーム構成	役割	利用技術、ツール
2016.04-2016.06	3 名 (PdM×1、iOS×1、Web×1)	Web	Rails 4.2, Ruby 2.3, RSpec, venice, AppStore

その他

その他、定常的・突発的に行なった業務を以下に記します。

- 協調しあう文化づくり
- 社内留学の推進
- 開発効率向上に関するツールの開発・導入 (kpt-bot, review-bot, Pull Panda)
- Dependencies upgrade withdependabot
- Release manager
- Quipper School / Video、スタディサプリのリリース分離
- 健康促進のための懸垂台の導入
- 読書会・勉強会の運営 (Real World HTTP, SRE 本 etc.)
- 社内図書館の発案と実践

SCSK Corporation (2012.04 - 2015.08) 3 yrs 5mos

SCSKは住友商事子会社の大手 SIer です。

私はシステムエンジニアとして不動産業界向けの業務システム開発に携わっていました。複雑な要件のデータモデル設計・帳票出力プログラムの作成・Web アプリケーションやバッチ処理の開発を通じてソフトウェア開発の基礎を学びました。

2014 年以降は基本設計・テスト設計・運用・進行管理・ベンダーマネジメントが業務の大半を占めました。

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

Public Output

https://lapras.com/public/AVL0ALR に概ね集約されています。

ブログ

- 個人 (日本語) https://ohbarye.hatenablog.jp/

    - 人気順の記事一覧
- 個人 (英語) https://dev.to/ohbarye

企業ブログへの寄稿

- 『B/43 Tech Talk ～ Fintech×サブスクリプションサービス立ち上げの裏側～』を開催します
- reviewdog x Custom FormatterでRuboCopの自動修正を提案させるようにしました
- ActionMailer::Baseのサブクラスで値を変更すると全てのMailerに反映されてしまう設定がある
- データベースのメタデータ整備をRails generatorで楽にする工夫
- GitHub Appを使ってDependabotが作るpull requestを自動マージさせる
- 3Dセキュア入門 -B/43の3Dセキュア開発・運用の裏側-
- CRAからViteへ移行して190倍高速なdev server起動を得る
- TracePoint 活用事例: Sentry のローカル変数キャプチャ機能
- Idempotency-Key Header を使ったリトライと、オンラインイベントの”Kaigi感”
- 新メンバーが多い大型プロジェクトでの不確実性との戦い方
- Working Out Loud 大声作業（しなさい）、チームメンバー同士でのトレーニング文化の醸成
- より良い面接を実現するために “Quipper 採用面接ガイド” を公開しました
- カジュアル面談への扉
- プロダクトの「負債」を「機能」と呼び直す ～A/B テストを用いた”価値”の定量化～
- 退職の作法、あるいはオフボーディング実践入門
- グローバルサービスでのタイムゾーンとの向き合い方

プレゼンテーション

https://speakerdeck.com/ohbaryeに概ね集約されています。いくつか抜粋します。

- サブスクリプションサービスをつくる時にエンジニアが考えること (B/43 Techtalk)
- RuboCop Custom Formatter for Reviewdog Diagnostic Format (Gotanda.rb#53)
- DB外の副作用をトランザクションから分離しよう (Gotanda.rb#52)
- Balance Security and Usability in the Field of 3D Secure (Kaigi on Rails 2022)
- Safe Retry with Idempotency-Key Header(Kaigi on Rails 2021)
- Migration from React Native to PWA(JSConf JP 2019)

- Lightning Talk Session Organizer(Developers Summit 2019 Summer)
- プロダクトの「負債」を「機能」と呼び直すために (Regional Scrum Gathering Tokyo 2019)
- 貢献できる OSS の見つけ方 -完結編-(NodeFest JP 2018)
- 決済のトランザクション管理術 (Meguro.rb#19)
- サブスクリプションサービスにおける In-App Purchase 再考 (iOSDC Japan 2018)
- Quipper が実践する 定量データに基づく意思決定と開発 (Rails developer Meetup 2018 Day 3 Extreme)
- エンジニアも気にしたい色のアクセシビリティ (Roppongi.js#3)
- フロントエンドのレベル上げ (Meguro.rb#15)

OSS 活動

- Pull requests to OSS (pull requests 一覧)
  - AppStore の In-App purchase 用 API クライアント gem `venice` が Auto-Renewable (自動更新) に対応していなかったのでその機能を自ら開発したpull request
  - New Relic Ruby agent の gem `newrelic_rpm` が grape v1.2.0+ に対応していなかったので対応したpull request
  - その他 Node.js, yarn, Ruby などのメジャー OSS にもわずかながらパッチを送った経験あり
- Tools
  - goofi
  - review-waiting-list-bot
  - kpt-bot
  - markdown-server
- 公開している gems
- 公開している npm packages

コミュニティ活動

- Engineering Manager Meetup
  - 役割: オーガナイザー
  - 日本ではいまだ母数の少ない Engineering Manager という position についての情報共有を目的としたコミュニティです
  - 述べ 500 人以上が参加しました
  - 約 1.5 年にわたり独力で運営し、2020 年 4 月よりコミュニティでの運用にシフトしました
- EOF2019
  - 役割: コアメンバー

    - 日本ではあまり例のない「エンジニアリングマネジメント」をテーマに据えたフェスティバル
    - OST セッションの開催を担当
- Meguro.rb
    - 役割: メンバー、ホスト
    - 地域 Ruby コミュニティの 1 つである Meguro.rb の運営スタッフとして、イベント開催のホストを 2 度行いました

Personal Projects

趣味の開発や、学習の過程での成果物です。

Goofi

OSS への貢献をより簡便にするためのツールです。OSS 活動を始めたい初心者にとって最大の壁が「貢献対象を探すこと」だと考え、コントリビューションが推奨される repository と issue をリストアップする Web サービスGoofiを作りました。

Nodefest 2018 では同サービスに関する発表を行いました。

2020 年 1 月にGitHub が公式の類似機能を公開したので役目を終えそうですが、課題設定が正しかったことが追認された心持ちです。

時期	Repository	利用技術、ツール
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

漢字を入力するとふりがなが自動入力されるフォームを作成するためのライブラリです。jQuery で同機能を実現するものや、React の古めのバージョンに対応する類似ライブラリはありましたが、React hooks を利用したライブラリは見つからなかったため自作しました。

時期	Repository	利用技術、ツール
2019	https://github.com/ohbarye/react-use-kana	React 16.9, TypeScript 3.8.3

String Pixelater

文字を N * N dots の 2 次元配列に変換するライブラリです。当時Processingにハマっており、Generative Art を作る過程で実装しました。

時期	Repository	利用技術、ツール
2018	https://github.com/ohbarye/string-pixelater	TypeScript 3.6.4, rollup 1.23

Review bot

チーム開発でのレビューを促進させる Slack bot です。 指定した条件にマッチする pull requests 一覧を Slack に投稿します。

2019 年 6 月に GitHub が買収したPull Pandaに類似していますが商用で有料だったためか、この bot を fork して利用していただける企業が数社ありました。

時期	Repository	利用技術、ツール
2017.08-	https://github.com/ohbarye/review-waiting-list-bot	Node.js 8.x, GraphQL (client) Slack API, GitHub API v3-4, Heroku

Automaildoc

RSpec によるテスト実行時にメール文面をプレビューできる HTML を自動生成するライブラリです。

時期	Repository	利用技術、ツール
2017	https://github.com/ohbarye/automaildoc	Ruby 2.4, RSpec 3

Markdown Server

Markdown で記述された文書を HTML で配信する Web サーバを簡易に構築できるライブラリです。Python の学習のために自作しました。

時期	Repository	利用技術、ツール
2015	https://github.com/ohbarye/markdown-server	Python 3.7, bottle 0.12

学歴

- 慶應義塾大学 文学部 人文社会学科 西洋史学専攻 (2012 年卒業)

資格

- AtCoder
  - (2020.06) アルゴリズム実技検定 初級
- IPA
  - (2014.06) データベーススペシャリスト
  - (2012.12) 応用情報処理技術者
  - (2012.06) 基本情報処理技術者
- Others
  - (2012.03) 教職課程履修完了
  - (2010.12) 貸金業務取扱主任者

興味 / 関心

2020 年まではソフトウェア工学の分野、特に事業会社の Web サービス開発のうちアプリケーションレイヤーの開発で求められる技術力を業務でも趣味でも伸ばしてきました。今後はコンピュータサイエンス全般やアルゴリズム、低レイヤーやミドルウェアの学習を通じてソフトウェアに関する理解を全般的に深めていきたいと考えています。

- 競技プログラミング
  - AtCoder 緑 https://atcoder.jp/users/ohbarye
- コンピュータアーキテクチャ
  - 『コンピュータシステムの理論と実装』 (a.k.a. Nand to Tetris) の実践 https://github.com/ohbarye/nand2tetris
    - アセンブラ、VM 変換器、コンパイラを OCaml と Ruby で実装
- データ構造とアルゴリズム
  - Coursera: Princeton University Algorithms Part 1修了
  - 『みんなのデータ構造』でデータ構造の基礎を学んだ
  - データ構造とアルゴリズムの基礎を 100 時間学習してみての所感
- データベース
  - C 言語で SQLite のクローンを作るチュートリアル
- 学習ノート https://scrapbox.io/ohbarye/