

基本情報

氏名	近江宏樹 (Ohmi Hiroki)
生年月日	1994/4/13
居住地	東京都
最終学歴	明治大学商学部卒

アカウント

- [GitHub](#)
- [X\(Twitter\)](#)
- 更新頻度低
 - [Qiita](#)
 - [Zenn](#)

経歴要約

2017 年に SE としてテラスカイに新卒入社、その後 2019 年からゆめみに入社しフロントエンドエンジニアとして入社。Next.js、TypeScript を主に使用し、多様なウェブアプリケーション開発に従事。医療系予約サイト、エンターテインメント系コミュニティサイトなどでリーダーを務め、効率的な技術選択や週次振り返り会の実施などでチーム運営を改善。国際チームでの経験も積む。また 5 年間の CMS 開発ではメンバーからリーダーへと成長。Mantine UI や Orval の活用など、常に効率的な開発手法を模索。5〜20 人規模のプロジェクトで要件定義から運用まで幅広く担当し、顧客折衝やスケジュール管理スキルも磨く。チーム全体の生産性向上に貢献した。

興味・関心

- 人に役立つ・自分も気に入るプロダクトをつくっていきたい
 - 不幸な人が生まれないと尚良い
- 使いやすさを考えるのが好き
- 知識を集めるのが好き

仕事でのマインド

- 当たり前を当たり前と思わない
- 誰に対してもリスペクトをもつ
- ネガティブにならない、かといってポジティブになり過ぎない
 - フラットな気持ちを保つ
- 体調に左右させない、整える

就業条件

- リモートワーク
 - 出社が必要な場合は、電車で 30 分前後の範囲が望ましい
- 服装自由

経歴詳細

関わったプロジェクト数が多いため、主要なプロジェクトを記載します。記載できていないプロジェクトは[こちらの Notion](#)へ

株式会社ゆめみ (2019/5～2024/5)

2023 年 11 月～2024 年 3 月 / 医療・医薬 / CMS と予約サイト

- CMS と予約サイトの 2 つを同時並行で開発
- スケジュールの遅延がすでに発生していた
- プロジェクトはパートナー企業とも協力して進行
- 途中からフロントチームのリーダーとして参加

【規模 (人)】

20 (フロント : 9)

【役割】

チームリーダー

【担当フェーズ】

要件定義、設計、開発、テスト

【実績・取り組み】

- デザイナー、サーバー間のコミュニケーションの橋渡しになるよう動いた
 - どのプロジェクトにおいても意識していたが、より一層意識した
- プロジェクト内でしばしば重たい空気になることが多かったが、雰囲気になれずに必要なアクションを判断するよう努めた
- メンバーの能力とスケジュールを鑑みて、タスクの振り分けをするようになった
- メンバーがある機能に詳しい人となるようにタスク割り振りをした
 - 時間が経つにつれ理解度が増し、後工程の作業がスムーズになる
 - 属人化する懸念があるが、レビューでカバー
- Cognito を使った認証機能の実装をした
- 途中から Orval を導入して、コードの統一化を図った
- 項目数の多い画面は GitHub Copilot でバリデーションのベースを自動生成するなどして、時間短縮を図った

【反省・課題】

- デザイン・フロント間で合意がとれている UI で仕様を再調整する必要が発生してしまった
 - 参加時にどの UI がどこまで話が進んでいるかを把握すべきだった

2023 年 3 月～2024 年 1 月 / エンターテインメント / コミュニティサイト

- あるゲームのユーザーをターゲットにした大会やユーザー同士の交流ができるコミュニティサイト
- 短納期

【規模 (人)】

14 (フロント : 3)

【役割】

チームリーダー

【担当フェーズ】

要件定義、設計、開発、テスト、運用・保守

【実績・取り組み】

- 開発規模に対して短納期だったため、開発スピードを重視した技術選択をした
 - フロントエンドでは Next.js(Page Router)と Mantine UI を採用し、開発を効率化した
 - バックエンドでは OpenAPI を使用し、フロントエンドでは MSW と Tanstack query を予定していたため、Orval を採用してコードを自動生成した
 - モック作成時のブレや迷いを防ぎ、Tanstack query のキー管理の手間も省けた
 - 振り返ると、この選択が大幅な開発遅延を防いだと考えられる
 - Scaffdog でコンポーネントの雛形を生成可能にし、開発初期の開発スピードをあげた
 - 雛形を作成しておくことで、どんなコンポーネント設計をしてほしいかを共有できる
- 週1回の開発振り返り会を実施し、知見と懸念の共有の場とした
 - LT のような形式や、不具合の確認、リファクタリング推奨箇所の共有など、様々な形で行われた
 - 緊急度に応じて即時または翌週のタスクとして対応し、チームの当事者意識向上につながった
- デザインに対しては、技術的な実現可能性やスケジュールとの整合性を判断し、フィードバックをした
 - 必要に応じて代替案や工数別の提案をした
 - ユーザビリティ向上のアイデアも提案した

【反省・課題】

- テストコードの充実が不十分だった
 - 特に複雑な状態遷移を持つコンポーネントについては、テストでカバーするべきだった
 - 結果として仕様・設計の考慮漏れが生じた
 - 設計書の更新だけでは不十分で、見積もり時にテストコードも含めるべきだった

2019 年 11 月～2020 年 7 月 / アパレル / EC サイト

- 現行サイトへの追加機能実装とサイトのリニューアル
- 六割くらいが外国人メンバー
- マイクロサービスで複数のベンダーが関わる
- 客先常駐

【規模（人）】

20（フロント：9）

【役割】

メンバー

【担当フェーズ】

開発

【実績・取り組み】

- ネイティブアプリとやりとりするために SDK を利用する処理があり、ローカルでデバッグできなかったため大変だった
 - 開発環境で地道に確認するしかなかった
 - 退場時に改善すべき点としてお客様に共有した
- キャッチアップするために積極的にコードレビューをした
- 他社ベンダーとのすり合わせ
 - 誰に話を通せばスムーズなのかななどを考慮して、事を進めた
- 英語表現を学んだ
 - 語彙を増やすために類語を勉強し、タスク共有時に言い回しをかえて覚える

【反省・課題】

- リーダーの負担を減らしてほしいという意図に参加初期の頃は気づかなかった

- ランチや仕事の合間の雑談で察して、求められる動きがわかった
 - 以降はメンバーや PO からの質問に答えたり、他社ベンダーから得た情報を共有するなどサポートするように徹した

2019 年 5 月～2024 年 1 月 / エンターテインメント / CMS

- アプリに配信するコンテンツの管理画面
- メンバーとして参加し、途中からリーダー（約 2 年）として活動
- 開発フローがかなりカチッと決まっていた安定していた
- 3 ヶ月ごとのリリース
- CMS においてはデザイナーがいなかったため、お客様とすり合わせしながら画面デザインをする

【規模（人）】

20（フロント：3）

【役割】

メンバー、チームリーダー

【担当フェーズ】

要件定義、設計、開発、テスト、運用・保守

【実績・取り組み】

- Notion にチームのページを作成し、新メンバー向けのオンボーディングやリーダー向けのドキュメントを作成した
- 開発フローに則ったタスク管理用の Notion ページを作成し、1 サイクルごとに運用した
- 画面要素の一部を画像として保存する機能を実装した
- リーダーになったタイミングでリファクタを少しずつ進めた
 - マジックナンバーが数多く存在していたので、定数ファイルとして管理した
 - Scss の & が多用され、かつ階層が深くなっていたスタイルをシンプルな構造に見直した
- 複雑なステータス管理があり、お客様・バックエンドチームを交えて調整をした

【反省・課題】

- コンポーネント設計のリファクタリングを完了できなかった
 - 運用に影響はなかったものの、開発スピードの足枷にはなっていた
 - 着手が容易、かつリファクタリングの恩恵が高い箇所から始めるべきだった
- CMS の使い心地について対面で伺う機会があり、そこで意見の引き出しがうまくいってないと感じた
 - 漠然とした質問を投げかけてしまっていた
 - 質問や形式を練ったうえでヒアリングすべきだった（匿名のアンケートなど）

株式会社テラスカイ（2017/4～2019/12）

2018 年 4 月～2018 年 12 月 / ブライダル / スケジュール管理システム

- ブライダルで使用されるプランナー、施設などのリソースをタイムライン上で管理するシステム
- 要件定義、設計をしつつ、先行してプロトタイプをつくっていた
- フロントの 7 割くらいを実装
- 途中からサブリーダー的な位置付けになり、タスク管理やお客様の相談窓口をした

【規模（人）】

4

【役割】

メンバー、チームリーダー

【担当フェーズ】

要件定義、設計、開発、テスト、運用・保守

【実績・取り組み】

- 本格的にフロントの開発をはじめた
 - HTML, CSS, JavaScript, jQuery の勉強をしながら実装
- リッチな画面の実装
 - D&D、要素のリサイズ、要素の表示位置の計算
 - 週表示の Google カレンダーがイメージに近い
- パフォーマンス改善をした
 - API の呼び出しや描画のタイミングなどを調整
 - アニメーションまわりも Devtools の performance をみながら、なにか調整した（記憶が定かではない）
 - 要素取得するときには、なるべく id で取得するように変更した
- お客様の要求とスケジュールの調整ができるようになった
- なんでもテキストとして残しておくことが大事だと感じた
 - 仕様として Fix した機能を変えたいとお客様から要望があったときに、事が進めやすくなる
- フレームワークの不具合をみつけた
 - オープンソースではないので修正するまでには至らなかったが、不具合回避方法をみつけて対処した
- ユーザーマニュアルを作成した

【反省・課題】

- 実装に時間がかかってしまい、多少なりともスケジュールに影響を与えてしまった
 - 社内だけではなく外部コミュニティに技術的な質問をしてサポートを受けてもよかったと思う

運営ブログ

- [個人ブログ](#)
- [個人ブログ\(投稿者別\)](#)

スキルキーワード

Next.js, Nuxt, TypeScript, React, Vue, JavaScript, HTML, CSS, SCSS, Mantine UI, Bootstrap, Tailwind CSS, Styled components, Emotion, Redux, Vuex, Zustand, Jotai, Tanstack Query, SWR, Apollo Client, GraphQL, MSW, Jest, Storybook, ESLint, Prettier, Git, GitHub Actions, Docker, Vercel, Orval, Zod, OpenAPI, レスポンシブデザイン, WebView, CMS 開発, WYSIWYG エディター, JIRA, Confluence, Notion, GA4, Cognito, スクラム開発, アジャイル開発, 要件定義, 設計, テスト, 運用・保守, プロジェクトマネジメント, チームリーダーシップ, 顧客折衝, 工数見積, スケジュール管理, 英語

リンク集

- [経歴を PDF でダウンロード](#)
- [未読記事](#)
- [既読記事](#)
- [YouTube チャンネル](#)