

基本情報

氏名	近江宏樹（おうみ ひろき）
生年月日	1994/4/13（30 歳）
居住地	東京都
最終学歴	明治大学商学部卒

アカウント

- [GitHub](#)

経歴要約

2017 年に新卒で株式会社テラスカイに SE として入社し、2019 年から株式会社ゆめみでフロントエンドエンジニアとして活動。主に Next.js や TypeScript を使用。医療系予約サイトやエンタメ系コミュニティサイトなどの開発でリーダーを務めました。技術選定、週次振り返り会の実施、チーム間の橋渡しなどでプロジェクトを推進。Mantine UI や Orval を導入し、開発効率化も実現。英語での国際チームへの参加や多様な業界での開発を経験しました。

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） / 正社員 / フルリモート

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

【概要】

規模（人）	20（フロント：9）
役割	チームリーダー
フェーズ	要件定義、設計、開発、テスト
スキル	Next.js, TypeScript, Mantine UI, Orval, GitHub Actions, Cognito, GitHub Copilot, GitHub, JIRA, Confluence, Figma

- 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

【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
 - UI を実現するにあたってサーバー側が懸念していることをデザイナーに噛み砕いて伝える
- 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
 - 設計を担当した人がそのまま開発をする。のちの不具合修正においても主担当になってもらう
 - 時間が経つにつれ理解度が増し、後工程がスムーズになる
 - 属人化する懸念があるが、設計書作成の徹底とコードレビューでカバー
 - くわえてメンバーの能力を鑑みてタスク割り振りをした
- 項目数の多い画面があり GitHub Copilot でバリデーションのベースを自動生成して、時間短縮をした

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

【概要】

規模（人）	14（フロント：3）
役割	チームリーダー
フェーズ	要件定義、設計、開発、テスト、運用・保守
スキル	Next.js, TypeScript, Mantine UI, Orval, GitHub Actions, GitHub, JIRA, Confluence, Figma

- あるゲームに関連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
- 機能数に対して納期が短い
- 新卒メンバーの割合が高い

【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
 - 慣れ親しんだ Next.js の Page Router を使う
 - Mantine UI でコンポーネント開発を効率化
 - バックエンドの OpenAPI に合わせて、フロントでは Orval を採用してコードを自動生成

- 型、MSW のモック、TanStack query のカスタムフックを自動作成して、コードを統一。自前で実装する場合のブレや迷いがなくなる。TanStack query のキー管理の手間省ける
- Scaffolddog 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自然と共有できる
- 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、様々な形式で行った
 - この会自体が負担になるのは避けたいため、内容のまとまりは重視していない。参考になった記事のリンクを貼るだけでも OK
 - 新たにタスクが発生する場合もあり、緊急度に応じて即時または翌週に対応した
 - チームの当事者意識向上につながった。開発フェーズ時の単調さも解消されたと感じた
- 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

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

【概要】

規模（人）	20（フロント：9）
役割	メンバー
フェーズ	開発
スキル	Next.js, TypeScript, Styled components, Redux, GraphQL, Apollo Client, BFF, GitHub, JIRA, Confluence, 英語, Figma

- 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

【実績・取り組み】

- 案件のキャッチアップには英語の環境についていく必要があった
 - 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た
 - チャットで他メンバーが使っている表現を真似する
 - 積極的にコードレビューする
 - MTG で発言する
- 複数ベンダーと協業するいい機会になった
 - まず誰に話を通せばスムーズなのかを考慮した
 - 同じメンバーとして接して壁をつくらず、相談しやすい話しやすい人になろうとした
- 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

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

【概要】

--	--

規模（人）	20（フロント：3）
役割	メンバー, チームリーダー
フェーズ	要件定義、設計、開発、テスト、運用・保守
スキル	Nuxt, Vuex, SCSS, WYSIWYG, GitLab, JIRA, Confluence

- 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不在のため、お客様とすり合わせしながら画面デザインをした

【実績・取り組み】

- 暗黙知を形式知にした
 - Notion にチームのページを作成
 - 新メンバー向けのオンボーディング
 - 各フェーズにおけるステークホルダーや相談先などを記載
 - 既存ドキュメントの役割、みるタイミングなどを記載
- 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、バックエンドチームとお客様を交えて調整した
 - ステータス遷移図や状態・遷移条件を Confluence にまとめ、画面プロトタイプで UI や操作の違いを共有

株式会社テラスカイ（2017/4～2019/12） / 正社員 / 出社

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

【概要】

規模（人）	4
役割	メンバー, チームリーダー
フェーズ	要件定義、設計、開発、テスト、運用・保守
スキル	Salesforce, HTML, CSS, JavaScript, jQuery

- 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担当

【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、id で取得するように変更した
- 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
 - [Qiita で記事](#)にした

運営ブログ

- [個人ブログ](#)
- [個人ブログ\(投稿者別\)](#)

リンク集

- [経歴を PDF でダウンロード](#)
- [既読記事](#)
- [YouTube チャンネル](#)
- 更新頻度低
 - [Qiita](#)
 - [Zenn](#)