

# 目次

---

[プロフィール](#) / [経歴要約](#) / [リーダー・マネージメント経験](#) / [スキルセット・経験](#) / [人からの評価](#) / [どんなエンジニアか](#) / [パーソナリティ](#) / [職務経歴](#) / [趣味](#) / [ブログ](#)

## 近影とアイコン

---



## プロフィール

---

2024年7月 現在

- フリーランス エンジニア
- 現在フルリモート
- フルスタック (フロント・バックエンド・インフラ)
- プロジェクトリーダー / プレイングマネージャー
- 1984年生まれ
- 大阪府在住
- [Tandems株式会社パートナー](#) ( 以前の社員としての勤め先で、今もフリーランスとして関わっています )

## 経歴要約

---

- 自営業でWebサイトの開発/運営 (10年程度)
- Ruby on Rails / BigQuery / ansible などによる社内向けツール開発 (4年程度)
- フリーランスで Ruby on Rails / Nuxt / Terraform でのWebアプリケーション開発を中心におこなう ( 2024年現在 )
- アジャイル開発を多く経験し、スクラムマスター、要件定義、クライアントとの直接対話などを担当

## スキルセット・経験

---

### メインのスキル、経験

2024年7月 現在

- Ruby / Ruby on Rails - プロジェクトで7年程度
- Nuxt / Vue - プロジェクトで2年程度
- AWS / Terraform - プロジェクトで1年程度
- SQL (ORMを使わざの直接記述) / BigQuery - プロジェクトで5年程度
- Perl - 自営業時代に10年程度
- Apache - 自営業時代に基本チューニング程度
- Linux / docker / Shell script ( それぞれ、どのプロジェクトでも使うものなので、年数は割愛 )
- Stripe API / IIJ API / Google Vision API
- Ansible ( Provisioning ツール )
- 要件定義 / DB設計 / 技術選定 / アプリケーション設計

#### ツール類

- Git / Github
- Pivotal Tracker / Backlog ( Nu Labo )
- VsCode / Cursor ( AI特化エディタ )

#### 仕事で利用したことがあるもの

- Go lang / C# / PHP / Cake PHP / python (django)
- GCP ( Google Cloud Platform )

## リーダー・マネージメント経験

---

#### フリーランスとして

2023年ごろから2年程度

- 自分を入れてエンジニア最大5人程度の開発リーダー、テックリード、スクラムマスター経験があります。
- 各開発メンバーにはオーナーシップを持って開発にあたってもらいたいと思い、マイクロマネジメントはなるべくしたくないというのが基本方針です。
- 開発メンバーとは日頃よりプロジェクトのチャットで連絡したり、臨時ミーティングをしたり、定例で次にやることのプランニングをしたりという役割です。
- 開発の全体を把握して、クライアントとの定例でもファシリテーション(司会進行)を務めるようになりました。
- プロジェクトの開発チーム、クライアントの要件定義チーム、クライアントの連携開発チーム(複数)、デザイン開発会社のチームなど、多くのセクションとコミュニケーションしながら進めています。
- マネージメント/プロジェクトリーダーとしての役割を担うと共に、自分でもコードを書くという、プレイングマネージャー的な役割をしています、

#### 社員として

- 2017年ごろから2年程度
- 開発メンバー最大6人程度のWeb開発チームのリーダーを務めていました
- 既存メンバー、新規メンバーとの1on1を継続的におこなうなどの取り組みをしていました

## どんなエンジニアか (自己申告)

---

- バックエンド、フロント、インフラ、要件定義、クライアントとの対話、チームリード、マネージング、1on1など幅広く何でもします。
- プロダクト志向で、ビジネス視点も理解しながら開発を進めます。
- オーナーシップを持って物事を前に進めます。
- 開発全体や自分個人のパフォーマンスの向上を測り、常にコスト対パフォーマンスを考えています。

## 言語・コミュニケーション

- あらゆる開発にはコミュニケーション能力は必須だと思いますが、自分の場合は特に言語というものに気をつけるタイプだと思います。
- クライアントとの対話、チームとのコミュニケーション、プロジェクト課題を文章でまとめる時や、PRの概要を書く時など、人に分かりやすい言葉、文章を心がけています。
- 非同期のコミュニケーション(チャットなど)におけるレスポンスは早めです。また、忘れずに返事を返すよう心がけています。

## アイディアマン・代替思考

- ひとつの問題を解決するにあたって、複数の手段を考案することが身についています。手段AがダメならB、もしくはCというように。
- チームとのディスカッションでも代替案を多く出し、クライアントにも複数の提案をすることが多いです。
- プロジェクトやサービスの改善案を出すことも習慣になっています。

## 達成がモチベーションの源

- 技術が大好きな技術屋ではありません。
- 好きなのはチャレンジすること、人とコミュニケーションすること、コラボレーションすること、ものづくりや、物事を達成することそのものです。
- サービス開発がひとつのミッションだとすれば、あらゆる面で自分のパフォーマンスを発揮し、プロジェクト全体を進め、遂行することにモチベーションを感じます。

## ボトムアップ型

- 学者肌ではありません。
- スキル習得はトップダウン式ではなくてボトムアップ式です。その積み上げからやがて全体的な視野を獲得するタイプです。
- ロジェクトの業務の中で必要なこと全てを習得していますが、それがコストパフォーマンス最高の習得法だと思っています。

## キャッチアップはマジョリティ

- 新しい技術のキャッチアップに関してはアーリーマジョリティ、もしくはレイトマジョリティです。
- 周囲にアーリーアダプタの人があり、良いツール、方式を勧めてくれる場合には、自分でも積極的に試すことが多いです。(エディタ、コンソールツール、OSのアプリ、コンソールツール、AIレビュー、ライブラリの自動バージョンアップなど)

## 人からの評価

概して自己評価よりも他者からの評価の方が正確だと言いますが、今まで人から言われたことのある評価は次のようなものです。

- ・「生産性が高い。人の3倍ぐらいこなしてるんじゃないかな」
- ・「チームの士気を上げる」
- ・「人が嫌がる仕事でもこなす」
- ・「文章の概要の書き方が丁寧で分かりやすい」
- ・「仕事の進め方に感心した」

## パーソナリティ

---

### 基本性格

- ・性格的には明るく、寛容で、面倒見が良いタイプです。頼まれれば気前良くなんでも人に教えます。
- ・全体観を持ち、物事を前に進めたいタイプです。些細な事柄に関しては、議論するほうが時間がもったいないと思います。熟慮すべき事柄は熟慮しますが、ほとんどの事柄は素早く判断して物事を前に進めたいと思っています。
- ・人が気が付かないようなディティールにもよく気がつくことが多いです。物事に対して常に疑問を持ち、批判的思考で考えています。
- ・何か行動を始めるにあたっては、完璧主義者ではありません。「思い立ったが吉日」というタイプです。
- ・人に流されにくく、必要な意見/疑問を述べるタイプです。

### 苦手なこと

- ・堅苦しい組織だと窮屈さを感じてしまい合わない気がします。カジュアルな文化が好きです。
- ・単純作業は苦手です。プログラマなので、もちろん頭を使う仕事の方が好きです。(でも誰かがやらなければいけない単純作業であれば、引き受けもします)

### 性格診断

流行りの性格診断 16 Personalities では合理主義者の建築家でした



### 職務経歴

---

並び順は上が未来、下が過去です

# 大手BtoB向けWebサービスの新規・追加・保守開発

- 2023年1月 - 2024年7月 ( 現在 継続中 )
- フリーランスとしての開発参加です。

## 概要

- 大きなプロジェクトの関連プロジェクトとして始まりましたが、ほぼ 0->1 のサービス開発です。
- 初期開発は、開発者は自分1人だけで始まり、粗くスクラッチで必要な機能をだいたい作り上げました。 ( Nuxt + Rails )
- 数ヶ月後に計3人のエンジニア/デザイナーが加わりチーム開発が始まりましたが、自然的に自分がチームリーダー・スクラムマスター的な役割を担うようになりました。
- 開発チームの人数は、自分を入れて同時にエンジニア最大4人/デザイナ最大1人という規模です。
- 開発の全体を把握して、途中からはクライアントとの定例会議でもファシリテーション(司会進行)をするようになりました。
- 機能連携もあるサービスのため、連携先の開発チームと連絡や要件定義などをしながら進めています。
- マネージメント/プロジェクトリーダーとしての役割とは別に、自分でもコードを書くという、プレイングマネージャー的な役割をしています。

## 関連技術・役割

- Ruby on Rails ( API )
- Nuxt3 / Vue3 + Typescript ( + Javascript d3 ライブラリ など )
- Tailwind CSS
- Docker
- AWS / Terraform ( ECS / SES / ALB / WAF / RDS / Lambda / Transfer Family / Synthetics / Route 53 などによる構築・更新・保守 )
- 要件定義、見積もり
- 機能提案・レビュー
- デザイン提案・レビュー
- アジャイル開発のスクラムマスター、開発チームのプランニング
- 技術選定、実装設計、データベース設計
- クライアントとの直接の相談・対話、ミーティングのファシリテーション
- E2Eテスト、VRT ( ビジュアルリグレーションテスト ) の構築やCI設定 ( Playwright + Github Actions )

## プロジェクトでの取り組み

### E2Eテスト

- 高いレベルの品質保証が必要なプロジェクトのため、テストフレームワークPlaywrightによる実環境に対するE2Eテストを組んで、リリースのたびに実行するようにしました。
- CI環境でも同等のテストが実行されるようにして、開発の変更の単位、リリースの単位でそれぞれ品質保証を可能にしました。

### Javascript – d3 ライブラリ

- 「チャートを描画するためのJavascriptライブラリ」には要件に合うものが見つからず、SVGを直接描画できるd3ライブラリを採用しました、

- Nuxt の動きと合わせてユーザー操作に合わせてリアルタイムに描画内容を変えたりと、かなり自由度の高い描画を可能にしました。

## 管理者用のサイト

- 管理者用のサイトは RailsだけのMPAを作るか、もしくはSPA(フロントのWeb + バックエンドAPI構成)で作るか迷いましたが、後者を採用しました。
- 後者の フロント + バックエンド式を採用した理由は、 APIモードでないRailsで管理者サイトを作ってしまうと、 Rails周りの知識が余分に必要になり学習コストが高くなることや、 実装の自由度が低くなることを懸念しました。
- Rails一本の場合は CRUD処理を非常に簡単に作れて低コストではあるので、必要な画面数、機能数によっても損益分岐点は異なるのですが、結果、覚えなければいけない技術分野を減らさせて良かったと思っています。

## プロジェクトでの経験

### 経験 - フロント

- このプロジェクトではじめてフロント技術 (Nuxt3) を経験しました。
- 初期開発の最初の数ヶ月は自分ひとりでの開発でしたが、想像よりも学習しやすく、あまり苦労なく慣れることが出来ました。
- 途中から加わったフロント経験のある開発メンバーにTypeScript周りを整えてもらったり、良い設計を教えてもらったりとしながら、フロントにかなり習熟できたのではないかと思います

### 経験 - TypeScript

- このプロジェクトがはじめてのTypeScript経験でした。 ( Nuxt のスクリプト部分 )
- 最初は型の扱いに戸惑いましたが、やがて品質保証のために必須のものだと思うようになりました。

### 経験 - AWS インフラ

- サーバーレスで本格的にインフラを触るのは初めてでした
- 参考になる Terraform のコードを元に構築しましたが、根本理解のためにそれ以前にAWSマネージメントコンソールから構築してみたり、それを踏襲するTerraformのコードを新しく組んだり、コスト判断次第ではAWSマネージメントコンソールから直接構築したりと、インフラ経験/AWS経験を積むことができました

## 中小企業向け 社内独自会計システムのリプレースの要件定義

2022年12月 - 2022年12月 (1ヶ月間)

- 前のプロジェクトから次のプロジェクトの間の空き期間に、1ヶ月だけ参加しました
- 小規模企業で独自開発された会計精算システムのリプレースにおける要件定義
- Free API を利用しての開発
- 既に長く使われている独自システムをどうスタンダードな会計システムに移行するか、どこを切り落とすか、一般的な会計システムにはない業界独自のビジネス要件をどうアプリケーションに含めるかということが課題でした

## 大手企業 展示会用のスマート家電開発 ( IoT )

2022年10月 - 2022年12月 (4ヶ月間)

- フリーランスとして開発に参加。
- ひとつの展示会に向けての開発だったので、プロジェクト自体がごく短期間の開発参加でした。
- 開発メンバーは自分1人というミニマムな開発でした。
- 実際に「モノ」が目の前にある開発は初めてだったので、刺激的で楽しかったです。
- C# での IoT開発・アプリ開発をしました。
- ノーコードツール Adalo も少し使いました。
- サーバーのバックエンドはRuby on Rails で開発しました。

## 開発支援 A ( 大手BtoB向けサービス )

2022年 (1ヶ月)

- フリーランスとして開発サポートに参加しました。
- コードレビュー・アプリのテストや、オブザーバーとしての開発への意見出しなどをおこないました。

## 開発支援 B

2022年 (1ヶ月)

- フリーランスとして開発サポートに参加しました。
- モンキーテスト・バグ洗い出しでの開発サポートとして参加しました。

## 株式会社 Tandems 社員

2019年6月 - 2022年

- 前職のオフィス勤めからフルリモートでの仕事に変わりました。
- Ruby on Rails での JSON API開発 がメインでした。

## 大手BtoB向けサービスのWeb開発

- Tandems社員として、他会社の開発サポートで1年程度参加しました。
- Ruby on Rails のバックエンドエンジニアとして参加しました。 ( SPAのバックエンドAPI部分 )
- 既に構築済みのシステムに対するアップデート開発でした。
- 開発メンバーはエンジニア・デザイナーで4-5人程度。
- ビジネスロジック的にものすごく複雑なものを作る必要があり、本気を出す必要がありましたが、なんとか乗り切りました。
- 非常に複雑なロジックをフレームワークのORMを使いつつも高速化する必要があり、何段階もの高速化を重ねました。

## BtoC向けサービスのWeb開発

- Tandems社員として、他会社の開発サポートで1年程度参加しました。
- 自社Tandemsから自分を含めて3人の開発メンバーが参加したため、他メンバーとのコミュニケーションやサポートをしながらの開発でした。
- Ruby on Rails のバックエンドエンジニアとして参加しました。 ( SPAのバックエンドAPI部分 )
- 外部連携の多いサービスで Stripe API を初経験しました。
- 連携が多いサービスのため local 開発のしづらさがネックだったのですが、開発環境を改善して開発しやすい環境を整えました。

- この時期の Stripe API は過渡期だったのか、ビジネス要件に合うちょうど良い仕組みがなかったりして、今ある仕組みをどうにかつなぎ合わせて開発したりしていました。
- 重たいDBクエリが発生するロジックの高速化などを起こしました。

## 株式会社Aiming 社員

2015年7月 - 2019年4月

- 主に「剣と魔法のログレス」などのスマホゲームを作っている会社です。
- 当時の社員規模は各拠点を合わせて1000名程度。ちょうど上場した後のタイミングぐらいで、自営業時代のポートフォリオを活かしてWeb開発者として入社しました。
- ゲーム運営用のための社内向けWebアプリケーションの開発、保守を担当していました。( ゲーム本体の開発ではありません )
- Ruby on Rails を中心に集計用SQLの作成、BigQuery / ansible / Golang なども扱っていました。
- Railsがフレームワークではあったのですが、業務内容は割と泥臭く、ゲーム開発本体や運営セクションなど関連各所と連携を取りつつ、仕様変更に追隨していく仕事が多かったです。
- 自営業から転職しての初の就職でしたが、4年程度をかけてエンジニアとして習熟したと思います。
- RailsやLinuxのコミッターもいるような周囲の技術レベルが本当に高い環境でしたが、色々と勉強させてもらいました。
- アジャイル開発で要件定義、機能設計、実装まで含めて担当しました。途中から最大6人程度の開発チームのリーダーをしていました。

## 業務委託など

2014年-2025年ごろ

- 受託開発 ( PHP でのWebサイト作成、改修 ) ( 1年程度 )
- 常駐での業務委託 ( PHP / Cake PHP / Java )

## 自営業でのサービス開発・運営

2005年ぐらい-2015年6月

- PerlプログラムによるWebサイトを立ち上げて、10年ほど開発運営していました。 ( 掲示板・SNS )
- 収入源はAdSense広告でした。
- 「メビウスリンク掲示板」 現在は閉鎖済みです。
- ユーザー数は多い時で1日1万ユニークユーザーぐらいでした。
- 最初は無料配布の掲示板プログラム(Perl)をダウンロードして色々と改造していましたが、だんだんと自分が書いたプログラム部分が大きくなり、サービス内SNSとの連動も作ったりしていました。
- ログインの必要がない掲示板サービスだったので、特に規約違反の投稿への対策が大変で、管理者用サイトは素早く違反報告の確認、削除ができるように作り込んでいました。
- 最初はレンタルサーバーを契約していましたが、ユーザー数増加に伴いroot権限のある専用サーバー契約に移行しました。
- 専用サーバー契約に伴い Linux / Apache / ドメイン設定 など各種インフラも経験するようになりました。

## 趣味

- 
- ボードゲーム：最近はまり始めました

- ヘルスケア、散歩、ウォーキング
- 読書：愛読家です。小説、実用系どちらも。
- ブログを書く：昔からの趣味です。プライベートのブログも技術ブログも書きます。
- 映画館通い：半年ほどマイブームでした。
- 音楽制作：自分で作詞作曲した曲を、歌、ギター、ベース、ドラムの打ち込みをMTR(マルチトラックレコーダー)で組み合わせて録音したりしていました。

## ブログ

---

### 技術系記事

- [Qiita](#)
- [Gist](#)