

個人情報

名前: 居谷 和彦

年齢: 36 歳 (2026 年 1 月時点)

場所: 福岡県在住

学歴: 帯広畜産大学獣医学部卒業

概要

ソフトフェアエンジニアとして 7 年程度の実務経験があります (2026 年 1 月時点)。

フルスタックエンジニアとして稼働することがほとんどで、機能単位でまるっと開発、リリースまで担当することが多いです。

ざっくりとした要件や要望をもとに、PM やステークホルダーと相談しつつ調整しながら開発することが多いです。

主に SaaS プロダクトの開発に携わり、toB、toC 向けサービス両方の経験があります。

DeNA ではアジャイルチームにおいてプロダクトオーナー兼エンジニアとして半年程度チームリードを担当していました。

職務経歴

大阪市役所 獣医師公務員 (2016 年 4 月 - 2016 年 9 月)

大阪市内の保健所に配属され、主に市内の飲食店の衛生管理業務を担当しました。

個人事業主 (2016 年 10 月 - 2018 年 10 月)

Web ライターとして独立し、Web メディアに掲載する記事の執筆、管理などの業務を行いました。

また、自身でも Web サイトを立ち上げ、掲載する記事の執筆や管理、記事の外注作業などを行なっていました。

Web サイトは WordPress を利用しており、このときに初めてプログラミングに触れました。ゼロから WordPress テーマを制作したわけではありませんが、配布されているテーマのカスタマイズやデザイン変更などを自身で行なっていました。

スターフェスティバル株式会社 (2018 年 11 月 - 2019 年 11 月)

主にフロントエンド、バックエンド両方の開発に携わりました。技術スタックは Laravel、CakePHP、Vue.js、jQuery などです。

WordPress 製 LP の機能開発

フォーム機能の追加や、導入事例一覧の表示機能などを担当しました。

ビジネス側の人と相談しつつ、WordPress 上でできることやできないことの調査や仕様策定を進めました。

メインサービスの CakePHP → Laravel へのリプレイス

メインサービスである「ごちクル」の CakePHP→Laravel リプレイスを担当しました。

担当した主な機能は、カート機能のバックエンドや、販売ページ表示のフロントエンドなどです。

株式会社 ACES (2019 年 10 月 - 2021 年 6 月)

最初は副業として参加していましたが、正社員としての入社を打診され、2020 年 1 月から正社員になりました。主な技術スタックは React (Next.js) 、Python (Flask, Django) 、AWS です。

スタートアップ企業だったので、0→1 開発を何度も経験しました。また、リポジトリの初期設計から担当することが多く、どの技術、ライブラリを採用するかの選定を何度も経験しました。

機械学習 API 提供サービスである SHARON のフロントエンド開発

toB 向けサービス SHARON のダッシュボード機能開発を担当しました。

フロントエンドは正社員が自分一人と、技術顧問的なポジションの方が一人だったため、機能の多くを担当しました。また、利用料金表示機能についてはバックエンドからフロントエンドまで一気通貫で開発を行いました。

サービス立ち上げから関与していて、チームメンバーの一員として要件の擦り合わせなどのフェーズから参加しました。アジャイル的な進め方で、週 1 回のチーム MTG で次週のタスクの決定や、問題点の相談などを行いました。

ただ、当初はエンジニアとしての経験がまだ浅く、コーディングはともかく要件定義において十分に役割を果たせていたかというと微妙なところではあります。

Docker コンテナとして提供する機械学習 API サービスの従量課金機能

要件定義、従量課金を実現するためのログ取得、現在の利用料金を表示するためのダッシュボード、インフラの構築といった機能提供にあたり必要な開発を全て担当しました。

インフラ、API の実装はサーバレスアーキテクチャを採用し、Terraform および Serverless Framework を利用して実装を行いました。

ただ振り返ってみると、必要な API が想定よりかなり多くなったことや、開発環境の構築、運用が難しかったため、サーバレスで構築するより ECS のような環境を採用したほうが良かったかもしれませんと感じています。

ACES Meet の立ち上げ

toB 向け SaaS プロダクトの 0→1 開発です。Next.js を採用し、フロントエンドリポジトリの初期設計を担当しました。どういう構成にすれば簡単に書きつつテストの容易性を担保できるかなどを考慮しつつ設計を行いました。

これまで state 管理に Redux を用いる構成を採用していましたが、このリポジトリでは Context ベースで構築することにしました。

その理由としては、hooks によってほとんどの state がコンポーネントレベルのスコープで十分になり、Redux は state 管理のライブラリとしてはこのリポジトリにおいてややオーバーだと判断したためです。

Context については素振りをしてから採用しましたが、自身の経験不足により十分に使いこなせていたかというと少し微妙に感じています。

無職期間 (2021 年 7 月 - 2022 年 4 月)

この期間はエンジニアとしての活動はありません。自由な時間はあったので、趣味&勉強として React をスクラッチで開発したり、C 言語のコンパイラを作ったりしていました（完全オリジナルではなく、どちらもサイトを参考にしながらです。）

他には、TypeScript で Git を作ったりもしました。

また、この期間に多くの本を読みました。興味の赴くままに本を購入して読んでいましたが、振り返ると Web 技術関連、経済、貨幣関連の本や、行動経済学や心理学の本が多かったように思います。

フリーランスとして活動（2022 年 - 2023 年 3 月）

業務委託として toB、toC 向けの SaaS プロダクトの開発に参加しています。主な技術スタックは TypeScript、Next.js、Nest.js、GraphQL です。

参画当初はテストがない、TypeScript のコンパイルエラーが頻発（300 ファイル以上）、コンポーネント設計の粒度がバラバラなど、コードベース上に課題がいくらでもある状態でした。

そこで、機能開発を進めつつ、以下の施策を導入したり、自身で行なったりしました。

- テストが書ける設計を提案し、それに従ってリファクタリングしつつテストの導入
- TypeScript コンパイルエラーの解消を行い、strict オプションを有効化
- リポジトリにおける見通しの良いディレクトリ構成の提案と導入

ただ、現状の形になったのもなんらかの理由があるはずで、それを考慮せずにいきなり参画してきた業務委託メンバーがあちこち指摘するのはあまり心象が良くないと思ったので、ヒアリングを行いつつ、急ぎすぎない程度で導入を進めていきました。

課題はいくらでもある状態でしたが、リファクタリングに無限に時間が取れるわけでもないので、なるべくインパクトの大きい課題を選定し、解決するように心がけました。

また、このプロジェクトで Chakra-ui を経験し、UI コンポーネントライブラリをベースにしたコンポーネント設計について学びを得ました。

株式会社 ディー・エヌ・エー（2023 年 4 月 - 2025 年 4 月）

正社員として Pococha の開発を担当しました。

入社時はフロントエンドエンジニアポジションで開発を行っていましたが、2024 年からはバックエンド開発を担当し、PO 兼エンジニアとしてチームリードを担当しました。

技術スタックは Next.js、Ruby on Rails です。チームの規模は QA 含めて 6 名でした。

機能開発は主に以下を担当しました。

- 新機能開発・リリース
- 既存コードの package 化。npm package として配布し、複数リポジトリで使いまわせるように
- CI 高速化
- Sentry の導入、およびプロダクトクオリティアップのための Sentry 運用の確立
- 自動テストの導入
 - 知識のベースラインを揃えるための読書会の開催
 - テンプレートコードの整備
 - 各種振り返り会の設定などを行い、テストの文化をチームに根付かせた

PO としてはチーム内タスク調整や意思決定、四半期の開発スケジュール調整、他チームとの連携などを行っていました。

株式会社 Algomatic

月 128 時間（週 4）の業務委託契約でフルスタックエンジニアとして参加していました。

スタートアップなので開発スタイルや組織などはまだ固まっていないフェーズであり、こんな感じの機能がほしいという要望をベースに仕様調整、開発、リリースなどを行なっていました。参画当時、エンジニアは私を含めて 4 名の体制です。

技術スタックは Azure がベースで、CosmosDB、Azure Functions、Hono、React、Prisma などです。

アポドリというプロダクトの新規機能開発やパフォーマンスチューニング、本番環境で利用する生成 AI モデル利用料金のコストカット、生成 AI のプロンプト設計などを主に担当しました。

このあたりからコーディングは主に Claude Code を利用するようになり、自身でコードを書く機会は減っていきましたが、AI コーディング環境下でもスピードを維持しつつコードの質を担保するための工夫や取り組みを行うようになりました。

言語経験など

TypeScript: 7 年

React (Next.js) : 5 年半

Vue.js: 1 年

Backend Node.js: 4 年

Python: 1 年

Ruby: 1.5 年

PHP: 1 年

AWS: 2 年

Azure: 半年

自身の特徴など

エンジニアリングについて

新プロジェクト参画時のキャッチアップの早さには自信があります（技術面、事業面の両方）。

プロジェクトの現時点において有効なエンジニアリングを提案でき、チームを巻き込んで進めた実績があります。

コーディングでは変更に強い設計かつテストが容易に書け、それでいて第三者が読みやすいコードになるよう心がけています。具体的には関数や API の適切な IF 設計、命名などです。リポジトリ単位で俯瞰して一貫性のある設計やコーディングを行うようにしています。

コードベースの保守性をかなり重視しており、ワークアラウンド的なコーディング、エンジニアリングをすることは稀です。基本的には根本的な原因を探し、その対応を行うことで解決を行います。

コードレビューにおいてもその観点からレビューすることが多いです。テキストベースだと自身が想定している以上に強い印象を与えてしまいがちだと思っているので、できる限りレビュイーの意見やコードを尊重してレビューするように心がけています。

フロントエンドから、バックエンド、インフラまで一気通貫して開発を行うのが好きです。

直近では AI コーディングを行うことが多くなりましたが、小さく変更してセルフレビューしたあとコミットするなど設計やコードの質は落とさないように開発を行なっています。

チームマネジメントについて

コミュニケーション力や事業においての視野の広さを認められ、PO 兼エンジニアとしてチームリードを担当した経験があります。

自身がマイクロマネジメントをされるのが苦手なため、自身がリードを行ったチームでもそれは行わず、可能な限りメンバーの意思を尊重するようにしていました。

また、リーダーシップを発揮してグイグイ引っ張っていくタイプではない自覚があるため、皆の前に立ってリードするというよりは、メンバーが仕事をするにあたって障害となりうるものを排除することを目標に動いていました。

具体的には PdM との仕様調整や他チームとのタスク連携、スケジュール管理などです。

とくにチーム全体に影響がある意思決定については、自身が積極的に判断を行うようにしていました（もちろんメンバーの意見や知見を参考にしつつ）

個人開発など

Git 開発

無職からエンジニアとして再活動を開始する前にリハビリとして TypeScript で Git を開発しました。

<https://github.com/kazuhiko-itani/typescript-git>

Rust でインタプリタ開発

<https://zenn.dev/koduki/articles/fb7e20f3719ec5> を参考に、Rust で自作インタプリタ開発にチャレンジしました（ただ、設計をミスったので途中で断念しています）

<https://github.com/kazuhiko-itani/rust-rml-mini>