職務経歴書

更新日 : 2026/01/13
氏名 : 勝亦 亮

キャリアサマリ

IT業界で約15年働いてきて、営業職、総務・経理補助を経てITエンジニア職がキャリアの中心となっています。直近約9年間は現職のSTORES 株式会社に在籍し、バックエンドエンジニア、チームリーダー、およびエンジニアリングマネージャー（EM）としてサービスの成長に関わってきました。会社の規模が20人から500人程度になる中でECの他に予約サービス、決済、請求基盤、外部システム連携などの機能開発・運用保守を経験。2020年からは最大9名のマネジメントも経験しました。2025年1月より開発経験を再度固めていくため開発者へ回帰し、現在はRuby on Rails を中心に Java/Spring Bootを用いた開発にもあらたに取り組んでいます。

技術スタック

- 言語/フレームワーク: Ruby (Ruby on Rails), Javascript(Vue, AngularJS), Java (Spring Boot), VBA, Shell Script
- インフラ/ミドルウェア: AWS, PostgreSQL, MySQL, MongoDB, Redis, Elasticsearch, Docker
- ツール・その他: Zuora (請求管理), GitHub, Slack, Redmine, Sentry, Datadog, Claude Code

職務経歴

STORES 株式会社（2016年7月 〜 現在）

事業内容: お店を運営する全ての人のためのデジタルプラットフォーム「STORES」の開発・運営

内容	開発環境	概要	役割
⑦2025/07 - 統合基盤開発	【使用言語等】 Ruby, Ruby on Rails, Javascript, React 【インフラ】 Google Cloud 【DB】 PostgreSQL	**バックエンドエンジニア** • **統合基盤開発** ◦ Railsアプリケーションを用いた「強制ダウングレード機能」や「支払督促通知機能」など、収益・債権管理に関わる機能の設計・実装	メンバー
⑥2025/04 - 2025/06 決済サービス開発・改善	【使用言語等】 Java, Spring Boot, Javascript, React 【インフラ】 AWS 【DB】 PostgreSQL	**バックエンドエンジニア** • **決済サービス開発** ◦ Java/Spring Bootを用いたプロジェクトに参加 ◦ ビジネスプロセス改善に係る機能開発	メンバー
⑤2025/01 - 2025/03 請求基盤・予約サービス開発改善	【使用言語等】 Ruby, Ruby on Rails, Javascript, React 【インフラ】 AWS 【DB】 PostgreSQL	**バックエンドエンジニア** EM職を経て開発者へ回帰 • **Zuoraを利用した強制ダウングレード機能の予約向け導入**	メンバー

		○ 全サービス導入前に予約サービスへの導入を主導。Zuora社との要件整理・設計・実装を担当	
④2024/01 - 2024/12 請求基盤・予約サービス開発改善	【使用言語等】 Ruby, Ruby on Rails, Javascript, React 【インフラ】 Heroku 【DB】 PostgreSQL	**エンジニアリングマネージャー** • **Zuoraを用いた請求基盤構築と予約サービス導入** ○ 請求を管理する部門の業務圧迫とプロダクトの請求を統合していくことを目的にサブスクリプション向け管理システムを導入	EM
③2020/01 - 2023/12 ECサービスの開発・改善	【使用言語等】 Ruby, Ruby on Rails, Javascript, Vue.js 【インフラ】 AWS 【DB】 MongoDB	**エンジニアリングマネージャー** 最大9名（正社員7名、業務委託2名）のマネジメントを担当。 • **組織設計・運営:** チームを2つに分割しリーダーを配置。権限委譲を進め、現場が自律的に意思決定できる体制を構築 • **組織横断コミュニケーション:** 開発組織全体（13名）との1on1を継続。チーム間の情報分断を防ぎ、大型プロジェクトの火種を早期察知・解消。 ○ メモ: ■ Salesforce導入阻止 ■ 顧客管理リリース調整 • **請求基盤の刷新:** 外部プラットフォーム「Zuora」を用いた請求基盤導入プロジェクトを牽引。マネジメントのみならず自らもハンズオンで実装に参画	EM

		• **採用・文化醸成:** 採用全プロセスの主導、社内Podcastの立ち上げによるエンジニア文化の活性化、職能横断チームの組成	
②2018/07 - 2019/12 ECサービスの開発・改善	【使用言語等】 Ruby, Ruby on Rails, Javascript, Vue.js 【インフラ】 AWS 【DB】 MongoDB	**チームリーダー** • **プロセス改善:** デリバリーの透明化とチームの達成感醸成のため、スクラムを導入 • **重要案件の完遂:** 消費税増税対応、割賦販売法対応など、法規制に関わるミッションクリティカルな開発をリード	リーダー
① 2016/07 - 2018/06 ECサービスの開発・改善	【使用言語等】 Ruby, Ruby on Rails, Javascript, Vue.js 【インフラ】 AWS 【DB】 MongoDB	**バックエンドエンジニア** 正社員エンジニア5名のタイミングに参画 • **コア機能開発:** 予約販売、再入荷リクエスト、PayPal/Amazon Pay決済連携等を担当 • **外部倉庫連携:** 他社倉庫API（ベータ版）との連携機能を10ヶ月かけて独力で構築。不確実性の高い環境下で仕様調整・実装を完遂	メンバー

blabo 株式会社（2016年2月 〜 2016年6月）

事業内容: 共創コミュニティBlabo!の開発・運営

職種: バックエンドエンジニア（Ruby on Rails）

- 正社員唯一のエンジニアとして、WEBサービスの開発・運用全般を担当。技術顧問と連携し、初期の開発フロー構築や機能開発に従事

内容	開発環境	概要	役割
2016/02 - 2016/06 共創コミュニティBlabo!の開発・運用	【使用言語等】 Ruby, Ruby on Rails, Javascript, Knockout.js 【インフラ】 Heroku 【DB】 PostgreSQL	5人 （プロダクトマネージャー・プランナー・デザイナー<業務委託>・エンジニア<業務委託>・エンジニア<私>）	メンバー

株式会社ネットフォレスト（2010年4月 〜 2016年1月）

事業内容: Web デザイン・Web システムの開発(企画立案から運用まで)、ネットワーク・運用管理(BGP・オペレーションからサーバ構築・運用まで)

内容	開発環境	概要	役割
2013/04 - 2016/01 ホスティングサービスの運用	【使用言語 等】 shell script, Ruby(Ruby on Rails) 【OS】 Linux（CentOS） 【DB】 MySQL, PostgreSQL	チーム規模: 4名 ホスティングサービスの運用をメイン業務として担当しプロジェクトとしてAWS上でのサーバ構築なども担当	メンバー
2011/10 - 2013/03 総務・経理補助	【使用言語 等】 Excel VBA, Ruby	チーム規模: 2名 総務・経理の補佐を担当。営業	メンバー

		サポートも後半で担当した	
2010/04 - 2011/09 WEBサイト制作の営業・ディレクション	【使用言語 等】 HTML, CSS, Javascript, jQuery	営業チームはとしては5人 WEBサイト制作プロジェクト時はデザイナー1名と営業1名という形で担当した	メンバー

職務経歴詳細

Project: 開発に携わった主な機能

- ①エンジニア期（基本的には1名で開発）
    - FAQ機能
    - 再入荷リクエスト
    - ★販売期間
    - 売上失効
    - ★PayPal決済
        - PayPal保留対応
    - レビュー返信機能
    - Amazon Pay
        - 3名で開発（開発中に担当者が退職したため1週間で引き継いでリリース）
    - ★予約販売
    - ★Instagram販売連携
        - 2名で開発
    - ★★倉庫連携
- ②リーダー期（設計とレビューとサポートで実装に関与）
    - 増税対応

        - 割賦販売法案対応
        - ★★Salesforce 導入プロジェクト（撤退対応）
- ③マネジメント期
    - 採用
        - 採用広報・イベント
        - 書類確認・カジュアル面談・面接・オファーレター
        - ★★採用基準の作成
    - 評価・査定・フィードバック
    - プロジェクトマネジメント
    - ★★チーム拡大による疎遠化を防ぐ発信
- ④マネジメント・エンジニア期
    - Zuora導入
- エンジニア期
    - ⑤請求基盤・予約サービス開発改善
    - ⑥決済チーム
        - 機能開発・運用
    - ⑦統合基盤アプリ
        - 猶予期間・強制ダウングレード対応

Project: 倉庫機能開発

- 状況
    - 外部サービスを利用した倉庫連携があった
    - そのリプレイスプロジェクトでまた別の企業の倉庫サービスと連携を行うこととなった
    - 顧客の利用料が下がることが見込まれ機能も増える見込み
    - 先方のAPIがまだ完成していないため要件のすり合わせからプロジェクトがスタート
- 課題
    - 7月完成予定のAPIが結果的には11月にβとして利用可能になった

    - APIドキュメントに不記載の項目がありエラーが頻発した
    - リリース後に知らない倉庫の仕様（保留荷物の在庫化）が存在することがわかった
    - プロジェクト中にデザイナーが退職、PMが休職した
- 行動
    - βリリースまでは社内で完結する部分の実装を進めた
    - APIの仕様に関してはslackで逐一質問できる体制を用意してもらった。APIのリリースが滞るタイミングやエラーが頻発するタイミングでリリースに影響が発生することが見込まれたタイミングで先方の会社に訪問し直接コミュニケーションを取った
    - PMが退職したあとはお金面の部分だけ社長にお任せして、先方のPMの方との対応もうけもってプロジェクトを進めた
    - 保留荷物の在庫化対応についてリリース後に発覚したので、仕様を先方に確認しAPIを急いで用意してもらった
    - その際こちら側はこういう画面でこういう情報を表示したいという内容を合わせてお伝えしてAPIに情報を載せてもらえるよう働きかけた
- 結果
    - 1月中旬にβリリースができた
    - その後も順次既存ユーザーのデータ移行を行った
    - 年内にすべてのユーザーを新倉庫サービスにデータ移行し旧システムの負債を解消し、新倉庫サービスへの完全移行により運用コスト削減と拡張性を確保した

Project: Salesforce導入課題の発見と対応（EM時代）

- 状況
    - 顧客数が激増したタイミングで営業からサポートまでの顧客対応体験を改善するためSalesforceの導入が検討されていた
    - エンジニア工数を使用しないプロジェクトだったため外部委託をビジネス側で検討し依頼していた

    - プロジェクト後半で急遽エンジニア1名で参加した
- 課題
    - 当初は導入に向けたデータ出しと簡単なチャットボット表示用jsの追加ということで声がかけられたが、委託先の仕様確認と成果物レビューなどやることが増えた
    - 成果物レビューの際に社内のエンジニアで保守できないプログラムで上がってきた（c#）、また保守の契約はない。
    - 導入検討していたプロジェクトマネージャーが退職した
    - 責任者不在でカスタマー・マーケ・セールスともに担当分の作業を行っている状態になっていた
    - リリース後Pardotの料金仕様について現責任者に確認したところ仕様を理解されてなく月100万円のコストが掛かることがわかったが予算としては計上されてないことがわかった
- 行動
    - 開発委託の対応
        - 導入に向けたデータの利用についてビジネス部門の担当者にヒアリングし導入時に必要なデータの要件を再整理した
        - 委託先と要件の再確認のMTGを設定。紹介元のSalesforce社の担当者も同席いただくようにした
            - C# から Ruby での納品に変更した
            - スケジュールの再設定
            - GitHubリポジトリを作成しコードレビューのフローを含めていただくようにした
    - 想定外の仕様
        - Pardot利用についてのヒアリングをし直した
        - 担当者に予算についての確認をしたところ認識がなかったため自身の上長と相談をしたうえで、ビジネス部門の部門長に直接報告（年間1000万以上の費用が発生しうるため早急に判断が必要だと考えた / 現責任者にも報告する旨を相談）

- 結果
    - 月100万円以上の不必要なコストの停止
    - 関係者間で認識齟齬があることがわかった
        - 取締役からの要求と実際に導入された内容に齟齬があった
        - プロジェクト途中で責任者が変わってしまい情報伝達が曖昧になってしまった
        - 各担当者が各責任についてフォーカスしてしまったため範囲外のところにまで手が回っていなかった
    - 大きなプロジェクトの停止だったため、社内ドキュメントに経緯と振り返りを行ったドキュメントを書いて公開し、次回以降の導入時に振り返られるようにした（実際に次の導入プロジェクトの際にこのドキュメントを下にヒアリングを受けた）

Project: Zuora（サブスクリプション向け管理システム）導入

- 状況
    - 請求を管理する部門の業務圧迫とプロダクトの請求を統合していくことを目的にサブスクリプション向け管理システムの導入を決定
        - サブスクリプション管理のデータを元にサービスも契約状態を同期し適用する必要がある
    - 請求管理部門と開発者で連携してプロジェクトを進行
    - 部門横断で開発チームを組成
    - 予約システムから導入を開始
- 課題
    - 予約システムの請求パターンが多岐にわたっておりZuoraへの落とし込みで精査が必要
        - 請求方法、請求周期、契約周期、顧客特別対応、プランとオプションなどの組み合わせでパターンが多岐にわたった
    - 依存関係としてユーザーデータ統合、企業データ統合があり連携を各所と取っていく必要がある

    - 全体のスケジュールの中で元々請求書を管理するシステムが3ヶ月後に契約を終了することがわかっていたため早急に請求書データを移行する必要がある
- 行動
    - EMを担当しながら開発に復帰
    - 組成したてのチームなのでチームの進め方、開発のレビューの価値観、ドキュメント生成の基準、新しくサービスに関わる人々に概要を共有してもらう会の設定など開発スタート時にテンプレートを用意した。その後はチームでやり方をアップデートしていった
    - Zuoraのworkflowへの機能追加、予約システムと統合基盤とのユーザーデータ・企業データの整合をとりデータリペア、予約サービスの管理画面からZuoraデータを動悸する処理などを実装
- 結果
    - 2-3月で情報収集と設計をZuora社と詰める
    - 6月末までに請求書支払い分を移行完了、Zuoraを通じたプラン同期を開始
    - 10月末にクレジットカード支払い分のターゲットを移行完了
    - ユーザーデータ、企業データの差分修正統合
    - 年内に移行を終え、次の期から他サービスのZuora移行を行う準備を整えた

■ 自己PR

1.「現場の痛み」がわかるエンジニアリングマネジメント

5年間のEM経験を通じ、採用・評価・組織設計・予算管理まで経験ができました。特に、組織拡大期における情報分断を1on1や独自の施策で解消してきた実績があります。2025年からは再び開発現場にもどり、マネージャーとしての視座を持ちつつ、現場の課題にハンズオンで応えられる点が強みです。

2. ビジネスの継続性を支える「手堅い」開発力

決済連携、法規制対応、請求基盤、外部倉庫連携など、データの整合性と信頼性が強く求められる領域での実績が豊富です。不確実な仕様や技術的負債に対しても、ビジネスゴールを見失わず、段階的に改善を進める粘り強さを持っています。

3. 非エンジニア領域を含む幅広いビジネス視点

キャリア初期の営業・総務・経理経験により、技術を「手段」として捉え、会社の収益やオペレーション効率を最適化する視点を持っています。専門用語を使わず非エンジニアと意思疎通を図り、要件を技術に落とし込むディレクション能力も備えています。