

職務経歴書

2021/8/29

佐佐木基顕

目次

1. 個人データ
2. 自己PR
 - 2.1. 新しいことへのキャッチアップ能力
 - 2.2. 難易度の高いタスクに対する対応力
3. 職務経歴概要
 - 3.1. 株式会社レスタス
 - 3.2. 株式会社NSC
4. 職務経歴詳細
 - 4.1. B2B向け新規自社グループECサイト構築
 - 4.2. B2B向け新規自社ECサイト構築
 - 4.3. B2B向け新規自社ECサイト構築
 - 4.4. 既存自社ECサイトの機能追加及び改善
 - 4.5. B2B向け新規商材向け自社ECサイト構築
5. スキルシート
 - 5.1. 技術
 - 5.2. その他
6. 書籍
 - 6.1. 読む書籍の傾向
 - 6.2. 書籍例
7. アカウント

1. 個人データ

- 氏名: 佐佐木基顕(Sasaki Motoaki)
- 誕生日: 1992年11月8日
- 年齢: 28歳
- 住所: 大阪

2. 自己PR

2.1. 新しいことへのキャッチアップ能力

株式会社レスタスに在籍した1年3ヶ月の間に、5つのプロジェクトに参画してきた。また、バックエンドに軸足を置きつつ、フロントエンド、インフラなど、必要に応じて様々な技術を効率よく身につけ、業務レベルで扱うことができるようになった。このことから、貴社で現在扱っている技術や、将来、貴社で新しく導入することになる技術において、迅速にキャッチアップしていく自信がある。

2.2. 難易度の高いタスクに対する対応力

これまでに、複数のアプリケーションにまたがるような機能を機能単位で設計～開発まで担当したり、開発リーダーを担当したりするなど、当時の自分の能力・経験に対して難易度の高いタスクを何度も経験してきた。その際、事前に入念に調査を行い、その情報を元に熟考し、そのあとに実行に移す、ということを意識することで、大きな問題もなくそれらのタスクを遂行してきた。このことから、貴社においても、新規機能や、新規サービスの開発など、難易度の高いタスクを遂行していく自信がある。

3. 職務経歴概要

3.1. 株式会社レスタス

- 設立: 2011年6月
- 資本金: 5億1,062万円（資本準備金を含む）
- 社員数: 43人
- 在職期間: 2020/4～現在

自社ECサービスの開発に従事。これまでに、4つの新規開発、1つの追加開発のプロジェクトに参画し、内2つのプロジェクトでは開発リーダーを兼任。メインはバックエンドの開発だが、フロントエンド、インフラも必要に応じて担当。現在は、新規サービスの開発リーダー兼開発メンバーとして、要件定義～テストを担当している。

3.2. 株式会社NSC

- 設立: 1971年3月11日
- 資本金: 4,550万円
- 社員数: 523人（2021年2月現在）
- 在職期間: 2018/4～2020/1

化学メーカーの製造職として、社内の新規技術を用いた製品の製造に従事。独自の分析と工夫により、作業の正確性及び効率が工程内でトップであったほか、サブリーダーとしてマネジメントにも関わった。さらに、空いた時間を利用してデータ収集及び分析による提案や、現場の改善なども積極的に行った。

なお、本企業での経験の詳細については、業界業種ともに、異なるものであるため、以下の職務経歴詳細には記載していない。

4. 職務経歴詳細

4.1. B2B向け新規自社グループECサイト構築

4.1.1. 概要

新しく加わったグループ会社の事業拡大を目的とした、顧客便益性が高くかつ様々な商材の追加に対応しうる開発効率の高い汎用的なB2B向け新規ECシステムの構築プロジェクト。

4.1.2. 期間

4.1.3. メンバー

プロジェクト全体:10人

- プロジェクトマネージャー:1人
- サーバーサイドエンジニア:2→1人
- フロントエンドエンジニア:1人

- デザイナー兼フロントエンドエンジニア:1人
- デザイナー:2人

4.1.4. 担当領域

- 開発リーダー(プロジェクト中盤以降)
- フロントエンドエンジニア参画までのフロントエンドアプリの実装
- フロントエンドアプリで使用するAPIの半分以上の作成
- 担当機能における、要件定義、テーブル設計、実装、単体テスト
- GoogleCloudFunctionsを用いた、バックグラウンド処理の作成
- N+1問題を検出するライブラリの導入を通じたパフォーマンス向上
- GitHubActionsを用いたCIテスト環境構築

4.1.5. 参考

- 作成テーブル/総テーブル数: 28/56
- 作成機能例: 外部API連携による決済方法の導入、クーポン機能、商品レビュー機能、APIでの商品検索機能など

4.1.6. プロジェクトにおける課題とそれに対して発揮したバリュー

4.1.6.1. テーブル設計の課題

本プロジェクトは、元々問い合わせベースで受注していたグループ会社の商品を、ECサイトから発注可能にすることで、顧客便益及び事業の拡大を目指したものである。そのため、様々な商材・商品、注文方法に対応しやすく開発効率を高く保つような設計及び実装である必要があった。そこで、私は、過去の自分の実装や、その際の運用から得た知見、外部ライブラリの実装、様々なユースケースなどを参考にしながら、熟考した上で実装を行なった。その結果、企画サイドの方からは高評価をいただいた。

4.1.6.2. アーキテクチャの課題

本プロジェクトでは、現状、フロントエンドアプリとバックエンドアプリの2つがあり、これらをどのようにドメイン上に配置するかという議論になった。そこで、私はそれを実現するためにはどういった方法があるのか調査し、プロキシサーバーと、サブドメインまたはサブディレクトリを利用することで機能要件を満たせることを開発環境で確認した上で、提案し、そのうちの一つが採用された。

4.1.6.3. コードを書く上での課題

0から複雑な機能を開発する中で、メソッド同士が密結合になるなど、徐々に煩雑で変更しにくいコードになっていってしまっていた。そこで、依存性の注入といったデザインパターンや、クラスの継承などを効果的に用い、開発→リファクタリングを繰り返すことで、コードの保守性を保ちながら開発を進めることができた。

4.1.7. 習得技術

- Ruby on Railsを用いた開発では、扱ったことのないActiveJobの使用や、カスタムバリデータの積極的な使用、アプリケーション全体で使用するロジックのモジュール化、など技術的に様々な挑戦しつつ、さらに洗練されたコードを書くことができたようになった。
- GCPのサービスを用いたバックグラウンド処理の導入や、上記アーキテクチャの課題の経験から、扱ったことのない技術に対するのキャッチアップ能力や、開発においてアーキテクチャレベルで多角的に考える力を十分に養うことができた。

4.2. B2B向け新規自社ECサイト構築

4.2.1. 概要

既存の自社サービスにはない、リッチなフロントエンドアプリをベースとした、スプレッドシートライクなデータ管理機能付きECサイトの構築プロジェクト。

4.2.2. メンバー

プロジェクト全体:7人

- プロジェクトマネージャー:1人
- サーバーサイドエンジニア:1人
- フロントエンドエンジニア:3人
- デザイナー兼フロントエンドエンジニア:1人

4.2.3. 担当領域

- 開発リーダー(プロジェクト中盤以降)
- フロントエンドアプリ上でのユーザーアップロード関連機能開発
- BFF(BackendForFrontend)の構築における、技術選定、設計、開発、テスト
- フロントエンドアプリのDocker化

4.2.4. プロジェクトにおける課題とそれに対して発揮したバリュー

4.2.4.1. プロジェクト体制における課題

あまり開発に詳しくないプロジェクトマネージャーと、開発メンバーの間で、背景知識の有無からか、お互いにうまくコミュニケーションが取れないという課題があった。そこで、今回の担当領域が各領域の橋渡しの立ち位置であることなどから、他の開発メンバーから開発リーダーに推薦していただいたので、エンジニア歴はメンバーの中で最も短いながらも開発リーダーに就任し、それ以降のプロジェクトにおける仲介役となることで、プロジェクトの円滑な進行に貢献した。

4.2.4.2. タスク量の課題

プロジェクトの途中から開発リーダーになったため、急にタスクが増え、一時的に自分自身のタスクがキャパオーバーになりかけていた。そこで、他の作業者に依頼できるタスクは依頼し、依頼できないタスクは常に一覧化して整理し、それらの優先順位を常に意識しながら進めることで、問題なく仕事を進めることができた。

4.2.4.3. 技術選定の課題

BFFでは、ユーザーアップロード機能の補助的な役割で導入するため、処理が高速である必要があり、Ruby on Railsではない技術の選定が期待されていた。そこで、処理速度の他に、習得に際してのハードル、並行処理が簡単に記述できるなどの理由からGo言語を選択し、0から習得して、実装を行い、コンテナ化してGoogleCloudRun上に配備することで、処理が高速で、かつ処理が増加してもオートスケールするようなBFFの構築に成功した。

4.2.5. 習得技術

- 今回、初めてフロントエンドチームで開発を行った。最初は慣れない面もあったが、Nuxt.jsを用いたフロントエンドのチーム開発に学ぶことができた。
- フロントエンド、BFF、インフラと、領域横断的な設計や開発を行ったため、各領域における知識や、それらを組み合わせる設計力・構成力を培うことができた。
- Go言語を用いたことで、静的型付け言語を用いて開発する力を養うことができた。
- Go言語やGoogleCloudRunなど、状況に応じて新しい技術を積極的に使用することで、新技術へのキャッチアップ能力を向上することができた。

4.3. B2B向け新規自社ECサイト構築

4.3.1. 概要

本プロジェクト発足当時、特定の商材に特化した自社サービスしかなかったため、商材の種類を制限しない汎用的ECサイトを作ること、次に需要のある商材を見極めるためのテストマーケティングを可能にすることを目指したプロジェクトである。

4.3.2. メンバー

プロジェクト全体:3人

- プロジェクトマネージャー:1人
- サーバーサイドエンジニア:1人
- デザイナー兼フロントエンドエンジニア:1人

4.3.3. 担当領域

- 要件定義、テーブル設計、実装、単体テスト
- 外部APIと連携した在庫数量連携バッチ処理(KubernetesのCronJobを用い、処理が定期実行されるようにマニフェストファイルを記述)
- 本番環境への反映計画の調整

4.3.4. プロジェクトにおける課題とそれに対して発揮したバリュー

4.3.4.1. テーブル設計の課題

機能要件として、異なる性質を持つ商材に対応できる必要があるというものがあり、それが開発上の一番のネックであった。私は、自社の他のサービス、掲載予定の商材などを分析し、一つの商品に対して、価格や納期などを付加したオプションを自由に設定できるように設計にすれば要件を満たせるということがわかり、そのように実装した。結果として、リリース以来、現在までに商材単位で40種類以上、商品単位で800点ほどを追加することができており、概ね要件を満たすことができた。

4.3.4.2. 能力的な課題

実務4ヶ月目に参画したこともあり、技術的に難易度が高いと感じる面が多々あった。そこで、不安点や不明点があるときは、自分なりに仮説を考えた上で、先輩社員に相談をし、この実装で問題があるか、もし先輩自身ならどうするかを聞くなど、より良い方法がないか、常に熟考しつづけた。その結果、プロジェクト終了後に特に大きなトラブルなどは起きなかった。

4.3.5. 習得技術

- バックエンド開発を通して、状況に応じて、メタプログラミングを用いたり、単一責任の原則などの設計面を考慮した、より保守性の高いコードを書けるようになった。
- 0から自分でDB設計を行うことを繰り返したことで、DB設計能力が格段に向上した。
- 外部APIとの在庫連携機能開発によって、Kubernetes、APIなどの基本的な知識を身につけた。
- 企画段階から参画した初めてのPJであり、抽象的な課題を具体化して実装まで落とし込む能力を培うことができた。
- ローンチまでの残り期間、各実装の開発工数、各機能の重要度、緊急度など、様々な観点で考えた上で、企画サイドと折衝をし、実装予定の調整や、機能の取捨選択などを行なったことで、コミュニケーション能力や、スケジューリング能力を培うことができた。

4.4. 既存自社ECサイトの機能追加及び改善

4.4.1. 概要

目的は、運用コストの削減及び、既存のシステムへの需要の高い機能の追加であった。

運用コストの削減においては、運用にあたって特に問題視されていた、新規商品登録時におけるコスト削減、既存のシステムへの機能追加では、需要の高かった注文方法を追加できるようにすることを通じて目的を達成することを目指した。

4.4.2. メンバー

プロジェクト全体:3人

- プロジェクトマネージャー:1人
- サーバーサイドエンジニア:1人
- アドバイザリー:1人

※当初は、要件定義～設計及び、開発のアドバイザリーとして参画したが、PJ途中の体制変更により、サーバーサイドエンジニアへ。

4.4.3. 担当領域

- 要件定義、テーブル設計、実装、単体テスト
- 外部APIと連携した在庫数量連携バッチ処理(KubernetesのCronJobを用い、処理が定期実行されるようにマニフェストファイルを記述)
- 本番環境への反映計画の調整

4.4.4. プロジェクトにおける課題とそれに対して発揮したバリュー

4.4.4.1. 運用コストの課題

運用コストの削減のために、以前は、購入可能数量(100, 200...)に対し、異なる単価をそれぞれ登録していたところを、基準となる単価1つを登録すれば、数量ごとの単価が自動計算されるようなロジックに関する要件を定義及び、実装し、結果、商品追加にあたって、登録が必要な単価データ数を30分の1程度まで削減することができた。

4.4.4.2. 納期的な課題

フロントエンドアプリに必要なSVGファイルは、特殊なDOM構造である必要があり、これは以前までは手動で変換していた。また、今回の機能追加に当たって、900個ほどのファイルを追加する必要があり、PJの期限を考慮すると手動で変換したり、外注する時間もなかったため、DOM構造を一括で変換できるようなプログラムの作成を提案及び作成し、本来1ファイル当たり5分、トータルで9〜10人/日かかる作業を、数秒で完了できるようにした。

4.4.5. 習得技術

- フロントエンドアプリでは、TypeScriptやVueCompositionApiなど、使ったことのない技術が使われていたため、機能追加を通じてこれらの技術の基本的な使い方を理解できた。
- バックエンドの開発では、APIの開発や、DOM構造の操作に関してより学ぶことができた。
- アドバイザリーを通じて、保守性を意識したコードレビューをする力を培うことができた。

4.5. B2B向け新規商材向け自社ECサイト構築

4.5.1. 概要

東証2部上場企業との資本提携をきっかけとした、提携先企業の商材を販売するプラットフォームを作るための新規事業である。

4.5.2. メンバー

プロジェクト全体:3人

- プロジェクトマネージャー:1人
 - サーバーサイドエンジニア:2人
 - フロントエンドエンジニア:1人
 - デザイナー兼フロントエンドエンジニア:1人
- ※プロジェクトの途中からバックエンドエンジニアとして参画。

4.5.3. 担当領域

- バックエンド機能の開発・テストを担当。
- 一部機能は、仕様書を元に、自ら設計〜開発まで担当。

4.5.4. 習得技術

- 初めての実務レベルの開発を通じて、コード読解力、設計力、Ruby on Rails を用いた基本的な開発をする力を培うことができた。
- Git, GitHubを用いたプルリクエストベースのチーム開発や、Dockerを用いた開発にも十分に習熟することができた。

5. スキルシート

5.1. 技術

カテゴリー	スキル	経験期間	備考
OS	macOS	1年3ヶ月	難なく使用可能。業務で常に使用しており、プライベートでは7年前から使用。
OS	Linux	1年3ヶ月	難なく使用可能。Dockerを用いた開発や、production、staging環境のサーバーなどで経験している。
言語	Ruby	1年3ヶ月	難なくプログラミングが可能。 株式会社レスタスでのバックエンド開発で主に使用。
言語	Go	2ヶ月	基本的なプログラミングが可能。株式会社レスタスでBFF構築の際に使用。
言語	JavaScript	1年	難なくプログラミングが可能。 株式会社レスタスでのフロントエンド開発で主に使用。
フレームワーク	Ruby on Rails	1年3ヶ月	難なくプログラミングが可能。メタプログラミング、クラスの継承、APIを用いた開発なども可能。
フレームワーク	Gin	2ヶ月	基本的なプログラミングが可能。APIの作成、Errorインターフェースを用いたエラーハンドリングや、エラー監視ツールSentryの導入などの経験がある。
フレームワーク	Vue.js	1年	OptionsAPIを用いたプログラミングが難なく可能。 単一ファイルコンポーネント、VueRouter、Vuexなどを用いたSPAを作成可能。 CompositionAPIは基本的なコードの読み書きができる。
フレームワーク	Nuxt.js	3ヶ月	難なく開発可能なほか、設定ファイルの編集も可能。
データベース	SQL(PostgreSQL)	1年3ヶ月	ORMを用いた、クエリの組み立て、調整などが可能。素のSQLを用いた、基本的なCRUD操作が可能。
インフラ	Kubernetes	1ヶ月	k9sといったコマンドラインツールを用いたアプリケーションのデバッグや、マニフェストファイルを記述し、Kubernetesリソースを追加した経験がある。
DevOps	GitHubActions	1ヶ月	特定のブランチへのpushを起点として、自動でテストを行うような、CIの構築経験あり。
クラウド	GCP	1年3ヶ月	いくつかのマネージドサービス(特に、CloudFunctions、CloudRun、CloudStorageなど)を難なく使用可能。

5.2. その他

カテゴリー	スキル	経験期間	備考
マネジメント	教育	3ヶ月	質問しやすい空気作りや、自発的な思考を促せるよう意識しながら、後輩エンジニア教育をした経験がある。
マネジメント	開発リーダー	5ヶ月	開発以外の担当者との調整や、機能要件のドキュメント化、メンバーの進捗管理などを経験。

6. 書籍

6.1. 読む書籍の傾向

基本的に、業務で必要になった書籍などを最優先で読む。

それがないときは、コンピュータサイエンスや、ネットワークなどの影響範囲の大きい基礎的な勉強を主に進めている。

6.2. 書籍例

- オブジェクト指向設計実践ガイド 〜Rubyでわかる 進化しつづける柔軟なアプリケーションの育て方
- UNIXという考え方ーその設計思想と哲学
- コンピュータシステムの理論と実装 ーモダンなコンピュータの作り方
- [試して理解]Linuxのしくみ ~実験と図解で学ぶOSとハードウェアの基礎知識
- 図解入門 TCP/IP

7. アカウント

サービス	アカウント
GitHub	https://github.com/Sasaki-Motoaki
Qiita	https://qiita.com/motoakii