

# 職務経歴書

## 基本情報

項目	内容
氏名	安孫子 直弘 (アビコ ナオヒロ)
生年月日	1993.7.17
出身地 / 居住地	埼玉県 / 東京都
最終学歴	早稲田大学 法学部
資格	AT普通自動車運転免許

## 会社経歴

年月日	経歴
2018.4.1	富士ソフト株式会社 入社
2021.6.31	富士ソフト株式会社 退社

## プロジェクト経歴

2021.5~2021.6

### 大手機械メーカーの生産管理システムの保守（C#）

PLと2人で連携して開発エンジニアとして

#### 課題と私の行動

- 利用者が全体検索をした際、検索が終わるまでに30秒~1分かかっていた。
  - jQueryのAjax処理でページローディングを追加。
  - 検索クエリはC#のLINQで書かれていたが、無駄にループしている部分やDistinctで重複のチェックを書けていた箇所があり、メモリの消費量を抑えるために重複のチェックは犠牲にして、Existに書き換えた。

2021.4~2021.5

### ふるさと納税のWebサイトのデータ・ソースコード調査（Java）

3人チームで作業員として

#### 実績・工夫した点

- ソースコードの関数と関数のネストを洗い出す作業。
- 特定の関数や変数をさらに洗い出し、ファイルに書き出す作業。
  - 人の手で作業をする前提で進んでいましたが、私は自動化をした方が良いと考え、シェルスクリプトで文字列検索、操作の処理をし、ファイルに書き出すバッチを作成した。
  - また、Excelでは関数を活用し、該当の箇所は色を付けたり、フィルターを付けたりして視覚的に分かりやすくなるようにしたり、作業者の負担を減らす工夫をした。

2021.3~2021.4

## 外務省のビザ関連サイトとモバイルアプリの連携開発（Java）

15人チームでエンジニアとして

### 私がやったこと

- 主に画像ファイルを処理するプログラム、画像ファイルをDB、ファイルサーバ両方に保存し、APIでアプリと連携させるための処理追加とテーブルを追加する工程で発生したバグをテストし、改修。
  - 画像処理プログラムで使用していたソフトウェアの使用方法についてシステムを開発した他部署の担当者と連携を取り、使用方法を担当者に伝達した。
  - API連携用にテーブルを追加したため、追加したテーブルに合わせてPOSTされたデータを挿入するコーディングをした。

2020.5.1~2021.3

## 大手マスメディアの基幹システム開発（C#）

100人規模の開発チーム(6人)のPL補佐、構成管理チーム(3人)のCI/CD構築エンジニアとして

### 課題と私の行動

- 1日に2回、検証環境へのビルド・リリースをする際、担当者が手作業で行っていたため、ヒューマンエラーが発生していた。
  - シェルスクリプトでCI/CDを構築。Jenkinsを導入しようとしたが、納品までの期限と複雑な環境への適応が難しいと判断し、パッチでインターフェースを作った。
- AWS環境への受入リリースを開始するに当たって、従来のブランチ戦略(git-flow)ではブランチのリリースができない状態だった。
  - 受入環境用のリリースブランチとバージョンタグを作成し、リリースをするタイミングを確認してもらうための予約・確認用チケットの仕組みを作った。
- リリースをしたあとのリグレッションテストが手間で、自動化すると良い。
  - PythonのSeleniumでWeb画面を開いて何かしら操作を試みるだけでも良いので検討をした。これに関してはやはり私だけでは無理だったため、手が足りずにできなかった。
- データベースのデータの種類やカラムの型を調べる必要があった。
  - 元々SQLServerManagementというユーザーインターフェースを使っていたが、A5:SQL Mk-2というフリーソフトを使用することでCSV出力し、Excelに整理して担当者に渡した。
- 検証環境用データベースを安全にバックアップ、インポートする必要があった。
  - 特定フォルダに日付ごとにダンプし、ある特定の日付のダンプファイルをインポートするSQLパッチを作成した。これを単体テスト環境、AWS環境、受入環境で作成。
- 開発チームでチケットが乱立していた。
  - テスターが外部の経験が浅い新人だったため、そもそもテストをする観点が上手くできておらず、PLがイライラして当たっていたが、私は開発者としての視点やテスターの視点も自分の経験を元に、新人に優しく指導した。
  - 煩雑になっていたチケットの管理、整備をした。開発チームが解散する間際でもチケットが大量にあり、整備しきれなかったため、PLや開発者と話し合い一つずつ解決した。
- プロジェクト内でのドキュメントが整理されていない。
  - Markdownと分かりやすいドキュメントとは何かを理解し、例えばリリース方法をGIFで撮って貼ったり、新人が見てもそのままの手順で実行したらリリースができるよう整理された書き方を意識して常時更新した。
- 定期券購入者の電車、バス経路情報をデータベースから反映させる処理を開発。
  - 画面上に反映された経路情報をマウスオーバーすると全体の経路情報がツールチップ上に表示されるようにして利用者が分かりやすいデザインにした。
- 経費処理のSQLクエリを作成、経費画面のデータのソート・フィルタをコーディング。
- データベースのカラム定義書をExcelファイルで管理していたが、カラム定義の管理をVBAを使用して整備した。

2020.5.1~

## 新人の教育担当

- 担当プロジェクトと並行してオンラインで10人程度の部署の新人を教育
- C#の開発教育を担当

### 課題と私の行動

- 教育リーダーは開発経験がなかったため、テキスト選定やフレームワークの選定が現場と乖離してしまう可能性があった。
  - 教育リーダーと話し合い、現場とのズレがないようテキストとフレームワークの選定を行った。
- コロナ下でオンラインでの教育になったため、新人がマンネリする可能性があった。
  - 教育はプロジェクトに配属されるまではテキストを読みながら全員に掲示板の作成をしてもらっていたが、途中で私個人の趣向でCLIのプログラムでじゃんけんゲームを作成するというようなちょっとした課題や アルゴリズムについて調べて書いてもらうような課題も出してみた。
- ミーティングがただの進捗度報告会のようにになっていた。
  - ただ報告会をしただけでは新人のためにはならないと考え、1人1人の進捗を聞いて報告をしてもらうだけのミーティングを辞め、自発的に質問をしてもらうよう、挙手制にし 発言が少ない新人に関してはこちらから聞いて疑問点や発見点等を聞きだすようにした。
  - 新人教育の後、入社するようになった際、新人からは親切的な教育で親近感が湧いていたと言われた。

2019~2020.3.31

## レガシーシステムのマイグレーション（VB.NET）

10人(パートナー社員を含む)のプロパー要員として

### 私がやったこと

- 機械メーカーのVB6.0で開発されたレガシーシステムをVB.NETに移植
- 詳細設計書をVB.NET用に刷新
- IDE(VS2008)による自動マイグレーションでは互換性がない関数が上手く移植されていなかったため、ソースコードを追って開発者で修正していた。
  - 私はif文の論理演算子がエラーは発生していないが、全く違う条件処理になっていたことに気づいたり、文字列操作の処理結果が違うものになっているような関数の処理内容を注意深くデバッグし、発見したことをミーティングで提示して、横展開した。

2018.6.1~2019

## 大手機械メーカーの部品・来客管理・社員管理用メール開発（C#）

20人(パートナー社員含む)のプロパー要員として

### どういうシステムか

- 機械部品を検索して画面に表示する生産管理システムを開発。
- 来賓・来客の駐車場の予約、宿泊施設の指定、訪問する建物、滞在日数の登録等、または社内承認フローのシステムを開発。
- 人事管理用のメール送信機能を開発。

### やったこと

- 単体、結合テスト、本番用に環境変数の整備、ビルド、リリース
  - 検証環境用と本番用で分ける必要があった変数を分けてそれぞれconfigファイルを分けて管理した。
  - 本番リリースは失敗できない緊張下でconfigファイルを充分に区別したうえで私の手でビルド成果物、configファイルをリリースした。
- Webサーバー、DBサーバー、メールサーバーのインフラを構築
  - メールサーバーに使用するソフトを導入したが、社内でこのソフトを使用するのは審議が必要だったため、情報システム部と連絡を取ってソフトを使用して、SMTP、POP3の設定をしてアプリ側とのメール送受信するサーバーを構築。結局一度は許可されたソフトであるものの、再審議で使用不可となり、パートナーからは隠れて使用すればいいのようなことを言われたが、Microsoft Outlookに切り替えて検証機に設置した。

- メール送信機能を実装
  - 他の開発者が仕様を追加した際、リグレッションが発生していることに気づき、修正をした。
  - メール文を考えてXMLに書き出して実装。
- 詳細・基本設計書の仕様追加時の修正
  - 他の担当者が書いた設計書を参照しつつ、要件定義書から詳細設計書に必要な内容を書いた。
- システムの使用方法をPDF化
  - 他担当者が作成したWordでのシステムの使用方法を分かりやすくするためにPDF化。PDFフリーソフトを探してみ、てPDFをブックマーク化し、お客様のマニュアル冊子を作成した。

## 個人開発

### Ocha

2023.8~2023.9

いわゆるリンクをまとめておけるプロフィールサービスです。自分が欲しいので作りました。[わたしのプロフィールはこちら](#)

#### 使用した技術

PHP / PostgreSQL / Apache / Bootstrap / Docker

#### 学習した点

- RenderにDockerからデプロイするために初めてDockerを学びました。Apacheを含めてインフラ関係の知識を開発現場で使用する際に支障がないくらいに学習するために2, 3週間ほど費やしました。
- PHPも初めてでした。また、Webの知識を付けるためLaravel等のフレームワークは使用していません。
- セキュリティも対策しました。CSRF対策、XSS、SQLインジェクション等
- まだ実装したい機能がありまして、自動でログインしたい人のために自動ログイン機能を追加し、Cookieを用いてセキュアな実装をしたいです。

## 絵描き友達のポートフォリオ

2023.9~2023.10

### [デモサイト](#)

#### 使用した技術

React / TypeScript / Bulma

#### 課題

- アニメーション(アイコンのスクロールの位置によるスタイル変化)を実装したいですが、今はまだできていません。
- 最初はNext.js、GatsbyJS、Astroを使用して開発しようと考えていましたが、フレームワークゆえのできないこともあり、ReactとTypescriptのみで開発することにしました。

## 私の個人開発ポートフォリオ

2020~現在

<http://naopem.com/>

#### 使用した技術

HTML5 / CSS3 / jQuery

#### 課題

- CSSフレームワークを使用していないサイトなので、デザイン共にそろそろ刷新したいです。
- 自作ブログを開発し、その中に私の自己紹介やポートフォリオを組み込むような形にしてみようという考えもあります。

## Youtubeコメントを取得し検索/ソートをするWebサイト

2019~数週間程度

[Github](#)

YoutubeAPIを使用した何かを作りたいと思い、コメントを取得してくるサイトを作ってみました。

### 開発したことで分かったこと

- APIを使用する方法を理解したこと。
- Webサービス開発はUIもさることながら、目的や達成できることの価値をもっと考える必要があることが分かった。

### 使用した技術

.NET C# / Azure / jQuery / Bootstrap / YouTubeAPI

## 英和⇄和英翻訳LINE bot

2019~数週間程度

[Github](#)

すぐに翻訳できるサイトかアプリが欲しいと思い、LINEBotの開発をしました。

### 開発をしたことで分かったこと

- これを利用すると何が良いのかということをまず考える必要がある。
- まずは自分が欲しいものかどうかという基準があった方がよい。

### 使用した技術

GoogleAppScript / LINE API

## 保有スキル

### プログラミング

- C# .NET Framework
  - データベースと連携したWebアプリの一部機能を開発することができます。
  - 個人開発でもAPIを用いてWebアプリを開発することができます。
- Java
  - ソースコードを修正することができます。
  - 機能追加までは実務上やっていますが、やることはできます。
- PHP (v8.2.9)
  - データベースを用いてWebアプリを個人開発することができます。
- React (v18.0.0)
  - シンプルな静的Webサイトを作成することができます。
- Typescript (v4.4.2)
  - シンプルな静的Webサイトを作成することができます。
- jQuery
  - Ajax処理、アニメーション機能を追加することができます。
- HTML5/CSS3 (Bootstrap-v5.3.0, Bulma等)
  - 静的Webサイトを作成することができます。
  - CSSフレームワークは根本は押さえているので他のCSSフレームワークも使用できます。

### データベース

- MySQL
  - 簡易的なDBを構築し、クエリを用いてCRUD処理を書くことができます。
- SQLServer

- 簡易的に複数テーブルからデータを取ってくるサブクエリを書くことができます。
- PostgreSQL
  - 簡易的なDBを構築し、クエリを用いてCRUD処理を書くことができます。

## その他

- Docker
  - Dockerfile、yamlを書いて個人開発用の環境を構築、DockerHubにプッシュすることができます。
- Sass
  - CSSフレームワークにないクラス、要素を追加することができます。
- Node.js (v16.7.13)
  - サーバーとして使用し、npmを操作することができます。
- Shell
  - シェルスクリプトを用いて、Gitブランチを統合、ビルド、リリースまでのインフラを自動化することができます。
  - 文字列を操作したりファイル、フォルダの操作等をし、面倒な作業を自動化して効率化を図ったツールを作れます。
- Windows
  - Windowsコマンドを用いて、DBダンプ、バックアップ、バッチ処理を作成することができます。
- Apache
  - Webアプリのサーバーとして個人開発用に使用することができます。
- AWS
  - リリースのためにEC2起動、ログインをしていた程度です。
- チケット, Git管理
  - 現場でチーム開発をすることができます。
- 英語
  - 日常会話レベル (ビジネス会話レベルはちょっとやってみないと分かりません。)
  - ゲームでフレンドといつもテキスト、会話でコミュニケーションをしています。

## アピールポイント

**私は没頭する性質があり、エンジニアスキルを向上させるのに適してる性格です。**

私は2年間のゲームスキルを向上させる期間がありました。人生においてこれだけ没頭することは他にないと思います。エンジニアという職種においてスキルを向上させるためにはこの何かしらにおいて異常な向上心というのも大切になると思います。前職を辞めてから月日があるのでその点が不安になると思いますが、2カ月で経験がなかったPHPとReact、TypeScriptでWebサイトを開発しています。

また、フロントエンドは会社に勤めている間はお手上げ状態だったのですが、CSSフレームワークをいくつか使い、フレームワークに共通する概念やパターンも理解し、たとえ他のフレームワークであったとしても開発に使えます。また言語に限らずとも、ApacheやDocker、Node.js、PostgreSQL、MySQL等使ったことがなかった技術を学び、開発に使っています。

このように、私はエンジニアの生活スタイルに切り替え、たとえ触ったことがない技術だったとしても情報をキャッチアップして開発することができます。そして没頭する性質であるが故、エンジニアとしての技術を常に磨くために色々な情報媒体から情報を吸収し、自分で考え、より良いサービスを開発できるエンジニアになれるよう、トライ&エラーを繰り返している人間です。

**チャレンジに前向きな職場であればあるほど率先して動きます**

例えば、大手メディア企業の基幹システム開発で、構成管理としてチームに入った時は、そのリーダーが新しいことに前向きだったということもあり、CI/CDを構築するためにShellを学び、構築しました。それでも、やはりビルドリリースをするに当たって、ヒューマンエラーが生じてしまうという課題がどうしてもあったため、Jenkinsを取り入れて自動化を試みたり、ドキュメントをアップデートして情報を共有することに常に前向きに取り組んでまいりました。

たとえ新しいことには前向きではない職場だったとしてもなるべく言われたことだけをやらないよう、フリーソフトを使用したり、業務の効率化のため自動化をして改善してきました。

## 効率を大切にしています

せっかくエンジニアという職種についているので、時間とマンパワーで解決するような考えはなくしていきたいです。そのためにも自動化は素晴らしいという考えを広めていきたいです。私自身、面倒なことは常に効率を考えています。

## これからやりたいこと

### DevOps環境あるいはDevOpsを目指している職場でのアジャイル開発、スクラム開発

そもそもアジャイル開発が良い理由は、ソフトウェアの開発において最も適している方法はトライ&エラーだと思っているからです。

最初から計画通りに行くはずがないことは自分でサービスを開発、あるいはウォーターフォールで開発していると分かります。人は完璧ではないという発想のもと、スプリントを回して修正し改善していく発想が私の個人的な仕事の仕方や今までの課題の解決方法から定性的にも理解できるため、ソフトウェアにおいてもアジャイル開発が好ましいと考えています。

カルチャーフィットしている開発現場とDevOpsに理解がある組織で開発インフラが自動化されたエンジニアの快適空間を作りたいという思いも当然あります。

DevOpsの環境で働きたいからDevOpsを牽引するSREもしたいですし、DevOpsがある組織で基本は開発エンジニアをしていきたいと思っています。

### 現状よりも高いスキルを要求されるかどうか

保身的に現状のやれることをやるだけではなくて、自分よりも高いパフォーマンスを要求される環境にいたいです。

例えば、新しい言語の開発やパフォーマンスチューニングの挑戦、サービスを刷新していくために必要な情報のキャッチアップを更新していくというような常に新しいことに挑戦をしていきたいです。

### 明確な「なぜ？」が分かるかどうか

リーダー、上司との相性になるとは思いますが、基本的にやることに全て「なぜ？」を考えて働きたいです。自分の仮説や意見を表現しやすいかどうか大切ですが、これに関してはカルチャーフィットしているかどうかや様々な要因があると思いますので、できるだけ自分の意見を持ち、表現できる環境にいたいです。

### フィードバックがあるかどうか

BtoB、BtoCでのエンドユーザーのフィードバックや感想を感じられる、あるいは上司やリーダーとのコミュニケーションにおいて基本的に不透明にしないでフィードバックをしようとしている職場で働きたいです。

以前の職場ではフィードバックが不透明な職場とフィードバックやコードレビューをする職場の両方を経験してます。教育担当をした際にもフィードバックがあるかどうかでやはり断然相手の反応や親近感の感じられ方が違いました。

当然フィードバックがある方が人生やサービス、パフォーマンスに良い影響があるということが私自身の経験から分かりましたので、職場として大切にしていたりサービスにフィードバックを感じられる開発に携わりたいです。

## 会社を辞めてから2年の空白期間があったことについて

### 辞めた理由

そもそも富士ソフトを退社した理由は会社が目指している方向性と私が目指している方向性が違うということが主な理由です。

具体的には会社の求めていることとしてなるべく多くの大きなプロジェクトを渡り歩いて売り上げ数値を出すということが個人に求められています。

また、必ずしも全ての社員がというわけではありませんが、開発スキルは個人には求められず、ある程度の技術者は外部に求めることで、新卒入社した自社社員はプロジェクト管理をする人材を育てることが主な方向性でした。

私としては会社の制約をなるべく柔軟にもらい、新しい情報やスキルをどんどん取り入れていくことが望ましいという考えのため、方向性の違いで会社を辞めるに至りました。

### 転職活動をする前に辞めた理由

富士ソフトに勤めている間ですと、技術の代わりに時間を投資していくことが求められていたため、個人ではなかなかスキルを上達させることが難しいと判断し、会社を辞めてからスキルを付けようと思い、退社の後に次の会社を探すという判断で退社しました。

## 活動を停止していたことについて

具体的に申し上げますと、何ら診療や病等の問題はなくて、私の個人的な性格として、何かしらにハマると没頭して上位層のレベルになるまでは気が済まないという性質があります。

活動としてはプログラマーのような生活を送っておりました。動画もいくつか投稿しており、大会で賞金を獲得するというような具合です。そして、Githubでも履歴が見れると思いますが、その生活にも十分に満足し、2023/8 から、開発者としてもこういった具合でスキルを身に付けています。

## 好きなこと

- 個人開発が好きです。  
これからもまだまだ作りたいものがたくさんあります。Webで個人開発をしてユーザーもそれなりにいるようなサービスを自作している人はとてもリスペクトしています。簡単な自動化ツール等、開発に便利な物を作っているような人も開発者としてとても好奇心とリスペクトの念が湧きます。
- オンラインマルチプレイヤーゲームも好きです。  
FallGuysはプロレベルにまでやり、動画をアップロードしていました。賞金付き大会も出てました。
- 本を読むのも好きです。  
一時期は4か月で10冊以上技術書を買ってました。Effective DevOpsは読んで感動しています。他にもカイゼンジャーニー、アジャイルサムライ、Team Geek、SOFT SKILLS、達人プログラマーは たまに読むではモデルを形成し、開発体験、キャリア体験に落としたいと思う本です。

## ソーシャルメディア

- [Github](#)
- [Twitter](#)
- [Zenn](#)
- [Qiita](#)