

職務経歴書

概略

2019年5月 ～ 現在

- ENECHANGE株式会社
- Webアプリケーションエンジニアとして、受託した案件の開発を行っております

2012年12月 ～ 2019年5月

- 大網株式会社
- あみあみ事業部の情報システム部門でECサイトの構築補助や社内の各部署の業務改善を行いました

2006年4月 ～ 2012年11月

- 独立行政法人 日本スポーツ振興センター（JISS）
- 国内のトップアスリートを対象に科学的なサポートを行いました

2011年4月 ～ 2012年11月

- ポールトゥウィン株式会社
- 品質管理エンジニアとして、QA業務を行いました
 - 上記の JISS での業務と並行です

主なプロジェクト詳細

1. エネルギー事業者の連絡会サイトの新規開発

概要

太陽光発電の売買の仕組みが変更することに伴い、新しい仕組みをサポートすることが必要になりました。そこで、関係事業者の中の幹事事業者で作られる「連絡会」が利用するサイトを新規に開発しました。

このサイトを通じて、連絡会は入会事業者とのやり取りを行うことができます。

- サービス提供URL
 - <https://after-solar.fit/>

チーム構成

- ビジネス担当2名、事務担当1名、開発担当2名
 - 私は開発の主担当を務めました

課題

連絡会発足からサイト運用開始までの期間が短く、まずは最低限の機能でリリースする必要がありました。

また、入会する事業者は連絡会からの呼びかけにより募集しなければならず、入会を促進するようなサイトの構築が必要でした。

課題の解決方法

最低限の機能でリリースするために、プロジェクトに求められる内容をチーム全体で検討してから実装に取り組みました。

リリース後は連絡会および会員事業者の意見を聞きながら、重要度の高い機能から実装を進めています。併せて、入会を促進するための施策を実行しています。

結果と考察

提供した機能により、事務局と会員事業者の作業時間を3分の1程度まで減らすことができ、業務効率を向上させることができました。事前に十分な機能の検討があり、たとえば「メールのやりとり作業が占める時間の割合が大きかったこと」などを適切に把握したためと考えています。

また入会促進の施策をいくつか試み、アクセス状況の数値や事業者からのヒアリングなどを元に、改善サイクルを回しています。

用いたツールや技術

- Ruby on Rails
 - Active Storage
- フロントエンドは Webpacker を取り除き webpack でビルド
 - HtmlWebpackPlugin
 - Vue.js
- AWS Elastic Beanstalk
- GitHub Actions
 - 上記を用いる中で得た知見を社内ブログに書きました
 - 絵文字を使用すると GitHub Actions と AWS Elastic Beanstalk に罣が待ち受けている (いた)
 - <https://tech.enechange.co.jp/entry/2019/10/16/152337>

2. 事業者がお客様に提供するサービス (SaaS) の開発・保守

概要

- EMAPサービス
 - <https://enechange.co.jp/service/>
 - 「料金シミュレーション」 (お客様用)
 - 「サービス申し込みシステム」 (お客様用)
 - 「顧客管理システム」 (事業者用)
 - 提供先の例
 - 株式会社ヒナタオエナジー (東京ガス子会社)
 - <https://hinatao.enechange.jp/>

チーム構成

- ビジネス担当2名、開発担当3名

課題

「EMAPサービス」は複数のエネルギー事業者を提供するサービスです。提供先に応じて特定の機能の追加や無効化を行う必要があります。

すでに実装済みの事業者用の動作を壊すことなく、機能の追加や無効化をスムーズに行うことが課題でした。

課題の解決方法

既存のコードを読み込み、同様の流れで実装することに注力しました。

その意図が不明である場合や、適切な書き方が分からない場合などは先輩の方にお尋ねしました。

また、既存のコードを壊さないという観点からはテストを充実させることにも注力しました。

結果と考察

事業部内で最も巨大なコードベースのサービスであったため、実装には苦労をしましたが、前述の解決方法に従い周囲の方の助けも借りながら予定通りにリリースすることができました。事業者の日々の業務を効率化し、想定していた契約数をクリアすることができています。

既存コードの改修にあたっては、コードや業務フローの背景にある意図を理解する必要があります。そのため、新しくジョインしたばかりのときに一人で悩むことはあまり建設的ではなく、積極的に周囲の方に質問を行ったことが結果につながったと考えています。

- 参考
 - プレスリリース (PDF)
 - <https://www.hinatao.co.jp/news/pdf/200310.pdf>

用いたツールや技術

- Ruby on Rails
- Vue.js・Vuex
 - ERB にコンポーネントを埋め込む形で利用しています
- PostgreSQL
- AWS Elastic Beanstalk
 - 上記を用いる中で得た知見を社内ブログに書きました
 - AWS Elastic Beanstalk にサブモジュール込みでデプロイする際は、CLI実行ユーザにサブモジュールにアクセスできる権限が必要になる
 - <https://tech.enechange.co.jp/entry/2020/04/09/084500>
 - AWS Elastic Beanstalk で使用される Ruby のバージョンを指定する方法と、それぞれの方法の特徴について
 - <https://tech.enechange.co.jp/entry/2020/03/28/084500>
- Bugsnag
- RubyGems
 - Sorcery
 - PaperTrail

3. 事業者向けに特化した、エネルギー価格比較サービス

概要

個人向けの価格比較サービスはすでに展開済みであったところ、「事業者」が用いることに特化したサービスの新規開発を行いました。

- サービスURL（ホーム）
 - <https://emap-zelos.enechange.jp/>
- 参考
 - 個人向けの価格比較サービス
 - <https://enechange.jp/>

チーム構成

- ビジネス担当2名、開発担当1名（私）
 - まずはベータ版の提供を目標としていたため、小規模のチームです

課題

事業者が価格比較を行う目的である「新プランの開発」「既存プランの競争力強化」のための機能を、個人向けのサービスよりも使いやすくする必要がありました。

課題の解決方法

「事業者」が行いたいことを丁寧に抽出してまとめ、その要件をひとつひとつ実装していきました。

個人向けサービスでは最安値の提供こそが最重要な課題であるのに対し、事業者向けでは他事業者の価格の網羅性が必要になり、そこに焦点を合わせました。

加えて、新しく商品を開発するにあたって仮想の価格設定値を細かく変動させる機能を追加しました。これにより、どの程度の価格でどの程度他社との競争力が持てるかを手早く検討できるようになりました。

この機能では「キャンペーン価格」も設定でき、特定の時期に広告を打って集客する用途にも使えるようにしました。

結果と考察

実装した機能は先方の求めていたものと合致し、正式な契約を受注することができました。

これまで先方は専任の担当者を当てて Excel の関数で独自に価格を算出していたところ、本サービスの提供によって担当者が不要になり、誰でも計算ができ、その結果を容易に共有することができるようになりました。これにより人的リソースの効率化が達成できました。

ただし、先方にヒアリングしたところ、このサービスを利用する時期は価格改定の際の年一回が基本であるとのことと、その点で契約交渉が難しかったところがありました。サブスク形式やスポット課金形式などでの契約形態を考えていますが、現状では月額の高額提供にとどまっています。今後、多くの事業者にサービスを展開することを考えると、この点は課題点として残ります。

また、失敗談として、表示価格の一部に消費税10%の適用することを失念してしまい、先方に多大なご迷惑をおかけしたことがあります。以後は価格情報にQAを導入してミスを無くしました。

用いたツールや技術

- Ruby on Rails
- jQuery
- Vue.js
- CircleCI で PDF のメタ情報をチェックする方法について、技術ブログにまとめました
 - PDFファイルに不適切なメタデータ（「作成者」や「タイトル」など）が存在している場合にCIで検知する方法
 - <https://tech.enechange.co.jp/entry/2019/10/31/144920>

4. 広報チーム用のWebアプリの開発

概要

広報チームが日々の仕事の煩雑さに苦労しており、それを支援するWebアプリを開発しました。

チーム構成

私が一人で担当しました。

課題

ECサイトで出品された商品を Twitter で PR する必要があったところ、それまでは「出品商品の確認と選別」「ツイート本文の作成」「ツイートの実行」の一連の流れを全て手動で行っていました。

これを可能な限り自動化し、業務負担を軽減することが課題でした。特に「出品商品の確認と選別」がボトルネックになっており、他部署の作業状況が終わらなければこのステップに進めない状態でした。

課題の解決方法

課題に挙げた状況が起きていた理由としては「各課のDB権限の相違により、見られる情報が異なること」「情報にリアルタイム性が無いこと」の2点が挙げられました。

DB権限の相違については、すべての課が参照しても良い情報を抽出し、それをWeb API化して利用できるようにしました。

リアルタイム性が求められる情報については、上記のAPIが返す値がほぼリアルタイムで更新されるようにし、広報チームがツイートの作業を迅速に行えるようにしました。

結果と考察

導入前はツイートをするためだけに出品待ちをして残業していた広報チームが、事前の情報確認が可能になり、それをもとにツイートを予約することができ、退勤時間を約1時間早めることができました。

また、作成したAPIが他の部署でも用いられるようになり、例えば仕入担当チームの発注業務や在庫管理業務の効率の上昇にも寄与しました。

用いたツールや技術

- Ruby on Rails

- RubyGems
 - Sidekiq
 - 上記に関する知見を Shibuya.rb で発表しました
 - <https://www.slideshare.net/corselia/sidekiq-mysql-filemaker-gem>
- Runderck
- Embulk
- ImageMagick

5. ECサイトのマーケティング業務

概要

ECサイトの売上を向上させるため、Web広告の展開やアクセス解析、SNSでのユーザの声の収集などを行いました。企画や分析とともに、それを元にした実装も行いました。

チーム構成

仕入チーム・海外担当・広報チーム・商品企画・実店舗担当からそれぞれ1名と、それに私を加えて6名のチームでした。

課題

Web広告、雑誌広告、街中看板広告などの広告を出向するにあたり、その費用対効果を高めることが課題でした。

課題の解決方法

それまで十分に実行できていなかった、基本的な施策に取り組みました。例えば 以下のような施策です。

- クリエイティブや広告テキスト毎のABテスト
- アフィリエイトサービス事業者ごとの運用管理
- QRコードを用いたコンバージョンの測定
- ECサイト内の回遊状況と離脱状況の調査
- SNSでのPRによるCPRやCVRの調査
- 毎日の売上金額やサイトアクセス状況、カテゴリ別売上情報のダッシュボード化
- 有力アフィリエイターを介した商品紹介
- おすすめ商品のレコメンドシステムの導入

結果と考察

まずはデータを正確に集め、それを可視化し、全社的に数字を意識してもらうことが目的でした。そしてその目的は一定程度達成することができました。

その次の段階である、データから意味を読み取って新しい施策を回していくという点に関してはまだ不十分なところはあるものの、部署を問わず「数値」を意識して業務を行うように変化していきました。

たとえば商品情報管理チームでは、過去の商品の販売数や利益を見てから新商品のマスタデータ作成にかかるコストを考え、作業内容に重みを付けるようになりました。

用いたツールや技術

- Google Analytics
 - データの取得にあたって gem を作成しました
 - simple-ga-reporting
 - <https://github.com/nikukyugamer/simple-ga-reporting>
- MySQL
- Embulk
- Google AdWords (Google 広告)
- Redash
- Superset
- 他社SaaS

6. その他のプロジェクト

上記以外に、以下のようなプロジェクトにも携わりました。

6-1. 他社サイトの価格調査

競合他社のサイトを Puppeteer を用いてスクレイピングし、価格の調査を行いました。

6-2. 業務支援のためのOSSの導入

アナログな方法中心だった業務を改善するためにOSSを複数導入しました。

Git (GitLab) やカンバンツール (Wekan) 、知見集積ツール (Knowledge)やBIツール (Redash、Superset) 、全文検索サービス (Fess) などのOSSです。

ポータビリティを考慮し、すべて Docker にて運用しました。様々な言語やフレームワークに触れることになったため、技術的な裾野が広がりました。

6-3. コーポレートブログの運用

主に技術的な点からコーポレートブログ (<https://news.amiami.jp/>) の運用をサポートしました。フレームワークは WordPress です。

それまではベンダーの方に実質的に運用もおまかせしていたところ、社内で運用を行うことによってスタイルの素早い調整やプラグインの導入など、柔軟なカスタマイズを素早く行うことが可能になりました。

6-4. 最適箱選択

複数商品の注文があった際に、どの大きさの箱を選ぶことが最適なのかを選択するアプリケーションを開発しました。

このアプリケーションを導入したことにより、海外からの注文のキャンセル理由の多くを占めた「送料」を導入前よりおよそ5%程度減らすことができました。また、一部の大きさの箱が梱包には使われないことが分かり、箱の購入費用と物理スペースの削減にも繋がりました。加えて、梱包担当作業者の「詰め直し」作業もほとんど無くなり、単位時間あたりの梱包数が約10%程度向上しました。

その他の活動

- 2016年の Qiita のアドベントカレンダーにおいて、全日程を一人で書き切りました

- FileMaker Advent Calendar 2016
 - <https://qiita.com/advent-calendar/2016/filemaker>
- ブラウザゲーム用のツールを開発し公開しています
 - 「窓の杜」で取り上げて頂きました
 - <https://forest.watch.impress.co.jp/docs/news/1197695.html>
 - ユーザからの不満や要望の声を聞きながらバージョンアップを続けています
- 個人ブログに技術的な内容の記事を定期的には書いています
 - <https://obel.hatenablog.jp/>
- 主にFacebook上にてゲーム「幻想水滸伝」のファンコミュニティの日本代表を務めています
 - <https://www.facebook.com/SuikodenRevival.jp>
 - 本コミュニティの活動がきっかけとなり「百英雄伝」というゲームの開発が決定しました
 - <https://rabbitandbearstudios.com/eiyudenchronicle/>
- 上記に関連し、ファンコミュニティでの人気キャラ投票イベントを共同で開催しています
 - 参加者の増加に伴い、手動の作業では負担が大きいため、2016年からシステムの導入をしています
 - サイトは以下の通りです
 - <https://election.suikoden.info/>
 - フロントエンドのコードは以下の通りです
 - <https://github.com/true-runes/suikoden-election-2020-frontend>
 - ツイート収集のコードは以下の通りです
 - <https://github.com/nikukyugamer/tweet-storage>
 - 2020年度の使用技術は以下の通りです
 - Vue.js
 - Ruby on Rails
 - Heroku
 - AWS Lightsail
 - Cloudflare
 - バリユードメイン
 - Google スプレッドシート
 - Web API の役割を果たしています
 - Twitter API
- 上記とは別作品のファンサイトを制作しています
 - サイトは以下の通りです
 - <https://eiyuden-shinsho.suikoden.info/>
 - フロントエンドのコードは以下の通りです
 - <https://github.com/true-runes/eiyuden-shinsho>
 - 使用技術は前項と同様です
- 「技術書典」へ参加しています
 - <https://techbookfest.org/event/tbf04/circle/21050012>
 - <https://techbookfest.org/event/tbf06/circle/38380001>

各URL一覧

- GitHub
 - <https://github.com/nikukyugamer>
 - <https://github.com/true-runes>
 - 後者は個人で作ったオーガニゼーションです
- SlideShare

- <https://www.slideshare.net/corselia/presentations>
- Qiita
 - <https://qiita.com/corselia>