

システム導入・開発者向け  
ユーザーが使いたくなるシステム構築のための  
「要件定義チェックリスト」

1. なぜ要件定義は大変なのか？
2. 「ユーザーが使いたくなるシステム」の要件定義チェックリスト
3. 「テックタッチ」のご紹介
4. 事例のご紹介
5. 会社紹介

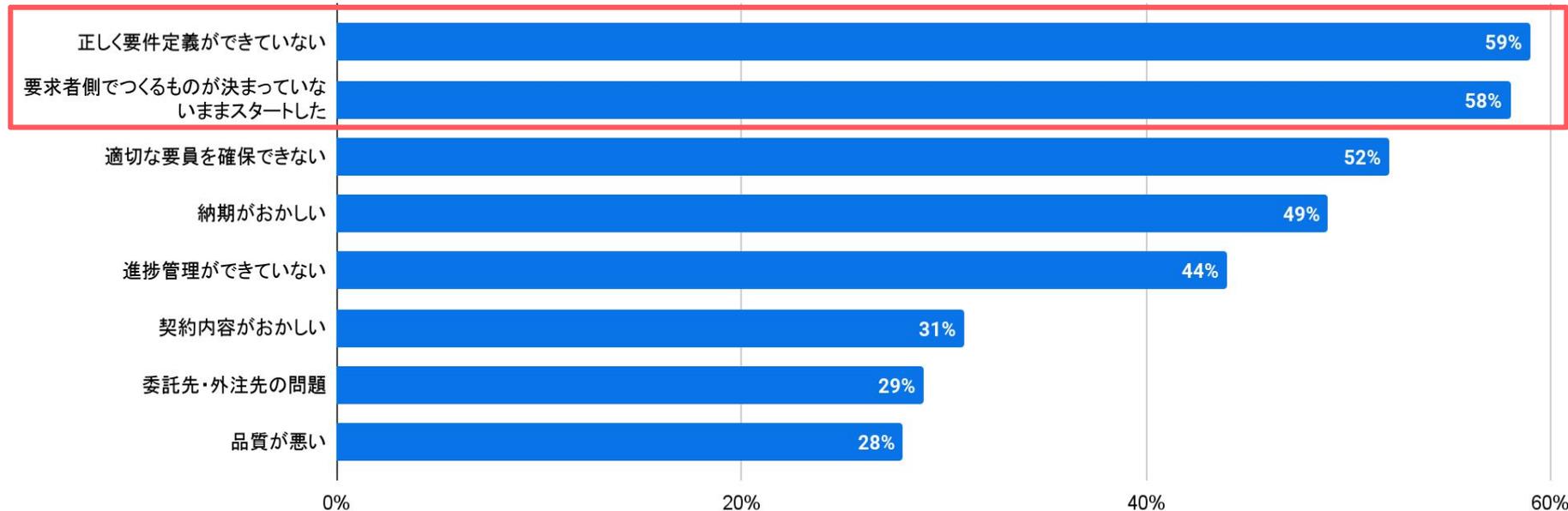
1

なぜ要件定義は大変なのか？

## プロジェクト炎上の理由の約60%以上は「要件定義」が上手くいっていないから

プロジェクトの最初のフェーズ「要件定義」。ここで失敗すると後工程にも影響し、結果的にプロジェクト全体の失敗を引き起こす可能性が非常に高い。

プロジェクトの失敗・炎上は、何が原因だと思いますか？ (n=100)



## 要件定義の失敗は、後工程にいけばいくほど手戻りコストが倍増

コロラド大学情報システム学部教授のアラン・M・デビス氏は、著作『ソフトウェア開発201の鉄則』の中で、ソフトウェア開発での要件定義の失敗が及ぼす手戻りコストについて以下の様に説明している

| フェーズ<br>(要件定義の失敗が判明した時) | 手戻りコスト<br>(要件定義工数比) |
|-------------------------|---------------------|
| 設計                      | 5倍                  |
| 実装                      | 10倍                 |
| テスト                     | 20倍                 |
| リリース                    | 200倍                |



要件定義で失敗すると  
プロジェクト全体の炎上リスク  
も倍増する

## なぜ要件定義は失敗するのか？よくある原因

要件定義の原因は「コミュニケーション起因」が圧倒的に多く、放置すると品質・コスト・納期が守られず、結果的に「誰も使わないシステム」が生まれてしまう

| よくある原因           | 説明                             | 放置した場合の副次的リスク                            |
|------------------|--------------------------------|--|
| 抽象的な要望をそのまま受け取る  | 「使いやすく」などの曖昧表現を仕様にと落とし込まず開発へ進行 | ユーザーと開発者の間で解釈がズレ、仕様変更が連発し工数・コストが増加       |
| ユーザーと開発の認識ズレ     | 言葉の意味や範囲に差異があるが、それを表に出せていない    | 誤解による実装ミスが頻発し、ユーザーの不信感や手戻りが発生            |
| 業務フローとUI/画面設計が乖離 | 実運用の流れとUIが合っておらず、使いづらいシステムになる  | 業務と合わないUIにより現場が使わず、定着しない“使われないシステム”が生まれる |
| 未決事項を決定事項として扱う   | 「あとで決めます」を反映せずにそのまま設計          | 設計後に「やっぱり違う」となる手戻りでスケジュール大幅遅延            |
| 合意内容の記録が残っていない   | 「言った・言っていない」問題が後からトラブル化        | 認識違いから信頼関係が崩れ、トラブルや責任の押し付け合いが発生          |

2

# 「ユーザーが使いたくなるシステム」の 要件定義チェックリスト

# 「使われない」は最も高くつく失敗です

要件定義を上手くできても、ユーザーに使われないと意味がありません。  
ユーザーに使われないとどんなリスクがあるのかご紹介します。

| 観点        | 影響・損失内容   | よくある二次被害や波及効果                       |
|-----------|---|-------------------------------------|
| 経済的損失     | 数百万～数千万円の開発コストが無駄投資化。<br>さらに毎月の保守・クラウド費用もかかり続ける | 「ROIが見合わない」とされ、次のIT投資が止まる／予算が削られる   |
| 現場の信頼喪失   | 「また使いにくいもの作ったな」と現場が萎える<br>→ IT活用そのものへの信頼が低下     | デジタル推進に対するアレルギーが組織に根付く              |
| 業務効率の低下   | 使われないので業務改善が進まない／非効率な手作業やExcel運用が温存される          | 本来削減できるはずの工数・ヒューマンエラーが残存            |
| DX停滞      | “使われる前提”が崩れたことで全社的なDXや業務変革の旗振りが進まなくなる           | プロジェクト全体の意義が問われ、他部門への展開もブロックされる     |
| 組織イメージ低下  | 「現場の声を聞かない」「作りっぱなし」というレッテルが開発・情シスにつく            | 他部署との連携がしづらくなる／IT部門が“相談されない存在”になる   |
| ナレッジが失われる | せっかくのノウハウやシナリオが、使われないがゆえに記録されずに風化する             | 再設計時にゼロからやり直しになる／属人化が進み運用がブラックボックス化 |

つまり、要件定義は単なるQCDの改善ではなく  
「ユーザーに使われるシステムを作る」ことをゴールに実施されるべき

## ✓要件定義チェックリスト～基本編～

| No | チェック項目                             | 詳細                          | 具体的な対応例                  | チェック                     |
|----|------------------------------------|-----------------------------|--------------------------|--------------------------|
| 1  | 画面遷移図やサイトマップを用意し、ユーザーが迷わない構成になっている | 必要な画面と画面間の遷移関係が一覧化されているか    | 画面遷移図やサイトマップを作成しておく      | <input type="checkbox"/> |
| 2  | 各画面の目的(ログイン=認証など)が明確に定義されている       | その画面でユーザーが何を達成したいかが定義されているか | 「ログイン=認証」「一覧=データ閲覧」などを明記 | <input type="checkbox"/> |
| 3  | UIコンポーネントを活用し、直感的に操作できる設計になっている    | ボタン配置やラベルがわかりやすいか           | UIコンポーネントに統一されたパターンを使う   | <input type="checkbox"/> |
| 4  | 配色や余白・フォントなど、デザインに一貫性がある           | 配色・余白・フォントなどが全画面で整っているか     | デザインシステムやUIキットを活用        | <input type="checkbox"/> |
| 5  | 主要アクション(保存・申請など)が目立つように設計されている     | ユーザーが迷わずに行動できる導線か           | 「保存」「申請」ボタンを強調表示         | <input type="checkbox"/> |
| 6  | 入力項目は必要最小限に絞られ、ユーザーの負担を減らしている      | 不要な入力を強いていないか               | 初回は最小限、詳細は後から入力          | <input type="checkbox"/> |
| 7  | 入力ルールやバリデーション(桁数・形式など)が明示されている     | 桁数・形式・未入力などのルールがあるか         | 「メールに@が必要」など仕様を定義        | <input type="checkbox"/> |
| 8  | 入力エラー時に、何が間違っているかが明確に表示される         | 何が間違っているか分かるか               | 該当項目を赤字+エラーメッセージ表示       | <input type="checkbox"/> |
| 9  | ボタンやラベルの文言が具体的で、わかりやすく表現されている      | 専門用語や曖昧な表現を避けているか           | 「送信」ではなく「申請する」など具体的に     | <input type="checkbox"/> |
| 10 | スマホやタブレットでも快適に操作できるレスポンシブ設計になっている  | スマホ・タブレット利用を想定しているか         | 幅ごとにレイアウトが崩れない設計         | <input type="checkbox"/> |

## ✓要件定義チェックリスト～業務・システム要件編～

| No | チェック項目                         | 詳細                    | 具体的な対応例             | チェック                     |
|----|--------------------------------|-----------------------|---------------------|--------------------------|
| 11 | 各画面の役割と機能(参照・編集など)が定義されている     | 各画面での処理内容が明確か         | 一覧＝編集可、詳細＝参照専用など明示  | <input type="checkbox"/> |
| 12 | 管理者・一般ユーザーなどロールに応じた表示制御ができています | 管理者と一般ユーザーで表示を変えているか  | 管理メニューは管理者のみ表示      | <input type="checkbox"/> |
| 13 | 実業務の流れに合った導線とステップで画面が設計されている   | 実業務に合った導線・画面構成になっているか | 顧客登録→確認→完了の3ステップ設計  | <input type="checkbox"/> |
| 14 | ログインしていないと遷移できないなどの制御がされている    | 未ログインでの遷移制御がされているか    | ログインなしでは遷移できないように制御 | <input type="checkbox"/> |

⚠ チェックなしが1つ以上→業務で達成したいことがシステムで実現されていない可能性が高いです！

💡 システム上にナビゲーションを出せる「テックタッチ」なら、業務に沿った操作ガイドを柔軟に実装でき、業務とシステムをうまく連携させることができます

機能イメージを見る

## ✓要件定義チェックリスト～UIUX設計編～

| No | チェック項目                         | 詳細                    | 具体的な対応例             | チェック                     |
|----|--------------------------------|-----------------------|---------------------|--------------------------|
| 15 | 同じ見た目や動作のUIは共通部品として再利用されている    | 同じ動作・見た目のものは統一しているか   | 共通ボタン・フォームを部品化      | <input type="checkbox"/> |
| 16 | 色覚や読み上げ対応など、アクセシビリティが配慮されている   | 色覚・読み上げなどの対応が考慮されているか | コントラスト／alt属性の設定     | <input type="checkbox"/> |
| 17 | アイコンや補足説明があり、ユーザーが迷わず操作できる     | ユーザーが迷わず操作できるか        | 「i」アイコンで使い方説明を表示    | <input type="checkbox"/> |
| 18 | モーダルと遷移の使い分けが適切に設計されている        | 分断を生まない設計か            | 確認はモーダル、詳細は遷移など使い分け | <input type="checkbox"/> |
| 19 | 初期表示の値や未入力状態がわかりやすくなっている       | 入力済み・未入力の判断がつくか       | 「選択してください」を初期表示に    | <input type="checkbox"/> |
| 20 | ステータス(保存済・下書きなど)が視認しやすく管理されている | 入力可否・表示切り替えが一貫しているか   | 「保存済」「下書き中」など状態表示   | <input type="checkbox"/> |

⚠ チェックなしが2つ以上→ユーザーがシステム操作時に使いにくさを感じている可能性が大きいです！

💡 「開発不要」でUIUXの改善をクイックに行いユーザーの不満を解消することができます

成功事例を見る

## ✓要件定義チェックリスト～エラーハンドリング編～

| No | チェック項目                         | 詳細                | 具体的な対応例             | チェック                     |
|----|--------------------------------|-------------------|---------------------|--------------------------|
| 21 | 入力途中での離脱時に、データ消失を防ぐ対策がとられている   | 入力が消えてしまうことがないか   | 離脱時に「入力内容が失われます」警告  | <input type="checkbox"/> |
| 22 | 削除操作などに対し、キャンセルや確認がきちんと設計されている | 誤操作に対する配慮がされているか  | 削除には2段階確認を設ける       | <input type="checkbox"/> |
| 23 | 保存後に戻る場所や完了メッセージの表示が定義されている    | 保存完了後の戻り先や表示が明確か  | 「一覧に戻る+完了メッセージ表示」など | <input type="checkbox"/> |
| 24 | UI命名やルールに一貫性があり、用語が統一されている     | ボタン名や項目名が統一されているか | 「保存」と「登録」が混在しないよう整理 | <input type="checkbox"/> |

⚠ チェックなしが1つ以上→手戻りや問合せが発生するリスクが高いです！

💡 ユーザーが自分で解決し、問い合わせも削減した事例はこちら↓

成功事例を見る

## ✓要件定義チェックリスト～パフォーマンス・表示設計編～

| No | チェック項目                      | 詳細                   | 具体的な対応例           | チェック                     |
|----|-----------------------------|----------------------|-------------------|--------------------------|
| 25 | 表示速度や読み込みストレスを考慮した設計がなされている | 表示にストレスがないか          | 初回はローディング表示／画像軽量化 | <input type="checkbox"/> |
| 26 | 読み込み中にスピナーやメッセージで状態を伝えている   | ユーザーが「待っている」ことに気づけるか | スピナーや「保存中…」表示を入れる | <input type="checkbox"/> |

⚠ チェックなしが1つ以上→ユーザーの不満がたまり、解約や離脱に繋がるリスクが高いです！

💡 ユーザー体験をクイックに向上させ、満足度を高めることができます

成功事例を見る

## ✓要件定義チェックリスト～開発・検証しやすさ編～

| No | チェック項目                         | 詳細                   | 具体的な対応例               | チェック                     |
|----|--------------------------------|----------------------|-----------------------|--------------------------|
| 27 | Figmaやレビューを通じて、開発と仕様の認識が一致している | 実装に食い違いが出ないようにしているか  | Figmaでモック共有、レビューMTG実施 | <input type="checkbox"/> |
| 28 | モックやプロトタイプでも動作確認がしやすく設計されている   | 動作確認がしやすい構成になっているか   | モック段階でもクリック・遷移確認が可能   | <input type="checkbox"/> |
| 29 | ユーザーテストや操作検証を前提にした設計になっている     | 仮説検証しやすい構造か          | 操作フローと所要時間をテストで確認     | <input type="checkbox"/> |
| 30 | 構造や命名が柔軟で、変更や拡張に対応しやすくなっている    | 構造や命名が修正しやすく設計されているか | コンポーネント化・分離設計で柔軟性を確保  | <input type="checkbox"/> |

⚠️ チェックなしが1つ以上→開発スピードが遅い、要件定義が遅い可能性があります！

💡 要件定義の短縮、開発スピードを6倍にアップさせた方法があります

成功事例を見る

いかがでしょうか？

要件定義を楽にかつユーザーが使いたくなるシステムを作るために

役立つのがDAP(デジタルアダプションプラットフォーム)です

次ページ以降でDAPの機能や事例のご紹介をします

3

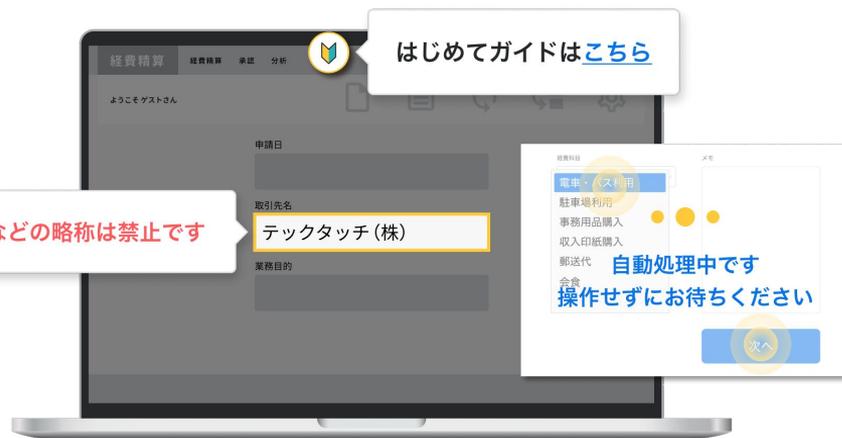
## 「テックタッチ」のご紹介

その壁、テックタッチがあれば、超えられます。わずか数分で「ユーザーが使いたくなるシステム」に変貌  
プログラミング不要でパワーポイントのようにかんたんに、UX改善が実行できます。



貴社のシステムに  
tagを埋め込むだけで…

かんたんに改善ができる！



# 「テックタッチ」ができること（エンドユーザー向け）

「テックタッチ」は、幅広い打ち手でユーザー体験を向上させます。

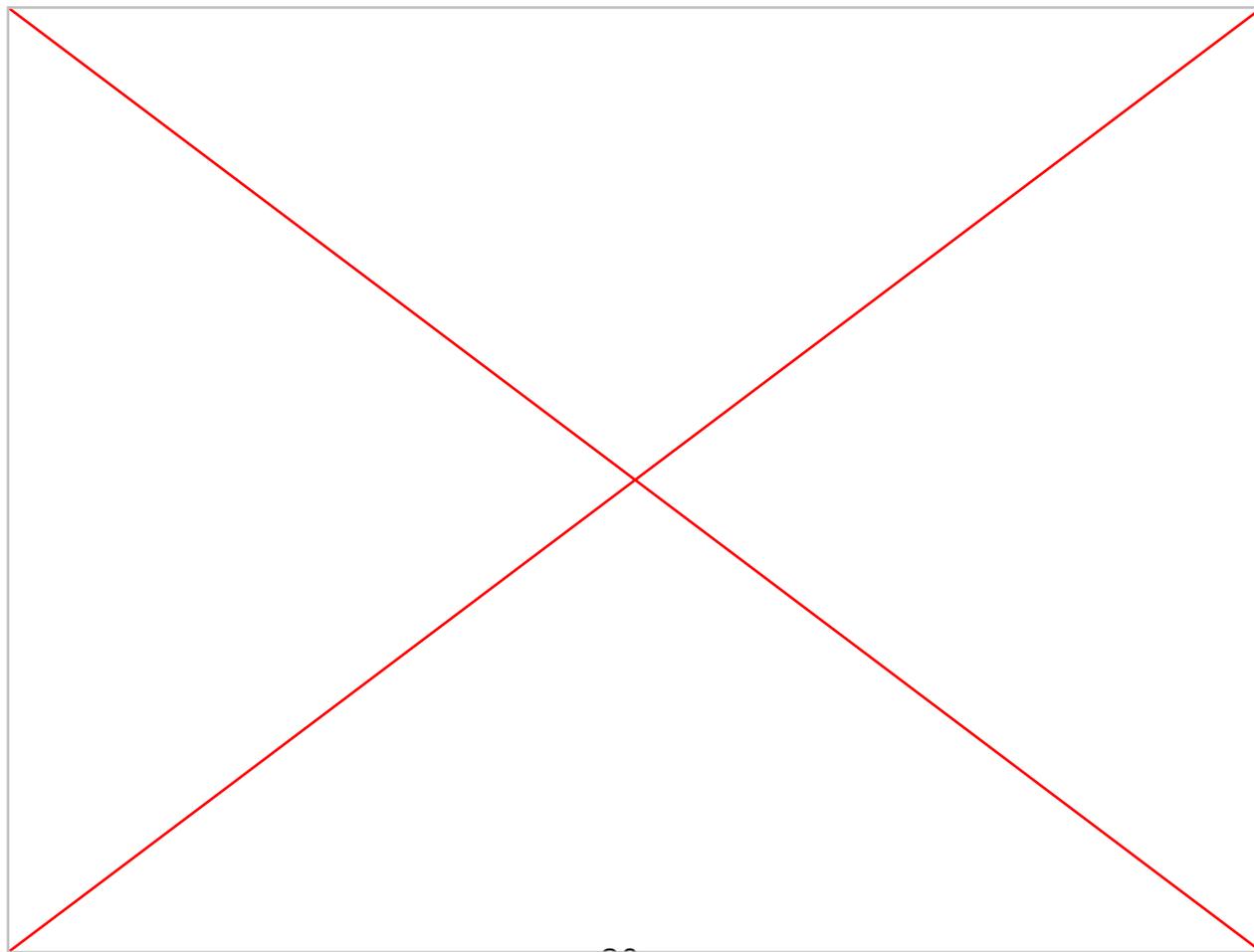
| ユーザー体験       | 「テックタッチ」ができること(一部抜粋)   | イメージ   |
|--------------|--|--|
| そもそも迷わない     | <ul style="list-style-type: none"><li>● マウスホバー時等に、入力ルールや注意点を表示</li><li>● ガイドで次にやるべきことに誘導</li></ul>   |   |
| 迷ってもすぐ解決     | <ul style="list-style-type: none"><li>● 主要操作・よくあるミス等のメニューを画面上に表示</li></ul>                           |   |
|              | <ul style="list-style-type: none"><li>● エラー発生時に対処法を伝えるガイドを自動発動</li></ul>                             |   |
| 単純作業をスピーディーに | <ul style="list-style-type: none"><li>● 定型の操作はユーザーに代わって自動操作</li><li>● 重複する入力内容は自動でコピー＆ペースト</li></ul> |  |

# 「テックタッチ」ができること (システム提供者向け)

「テックタッチ」は、使いやすいUI/UXでシステム提供者の課題を解決します。

| システム提供者体験               | 「テックタッチ」ができること(一部抜粋)  | イメージ   |
|-------------------------|---|--|
| <b>クイック&amp;柔軟な画面改修</b> | <ul style="list-style-type: none"><li>● 即座に画面UIを変更</li><li>● 分かりにくい導線を再整備</li></ul>   |   |
|                         | <ul style="list-style-type: none"><li>● よく使うメニューのボタンを設置</li></ul>   |   |
|                         | <ul style="list-style-type: none"><li>● ユーザーに不必要な情報を隠す</li></ul>  |   |
| <b>必要に応じたカスタマイズ</b>     | <ul style="list-style-type: none"><li>● 個別最適化な打ち手<ul style="list-style-type: none"><li>○ ガイドを「ユーザーごと」「会社ごと」に出し分けるなど</li></ul></li></ul> |  |

まずはこちらのデモ動画をご覧ください



▼テックタッチ デモサイトURL

テックタッチカード

<https://techtouch-card.vercel.app/>

テックタッチほけん

<https://techtouch-hoken.vercel.app/>

4

## 事例のご紹介

## 課題

- 個社別に画面カスタマイズの依頼が来るが、開発側では対応ができない
- 一方で、現場がある業務の場合は個別事情が多く発生しており、汎用画面では活用ができずに解約になってしまうケースもあった

## 打ち手や実装のイメージ

同じUIの画面に対し、テックタッチの機能を使って個社別のUIに画面を変更  
ソースコードを変えずに、画面UIのみ変更。運用はCS・営業で完結

### A社向け

### B社向け

## 効果

- 顧客のニーズに応えることで満足度向上
- テックタッチで顧客ごと画面をカスタマイズし、有償で販売。テックタッチ料金を上回る売上を獲得
- 解約リスクを低減しつつ、開発工数も軽減。

## 課題

- 顧客毎に社員情報入力フォームの項目がカスタマイズされている
- 顧客毎に異なるUIに関する問合せが発生し、サポート担当は全てのパターンを知らないので回答に時間がかかる。顧客満足度の低下を招いていた。

## 打ち手や実装のイメージ

異なるUIのパターンでも「共通して実装されている」機能を中心にテックタッチを実装

パターンA



パターンB



パターンC



共通機能にのみナビゲーションを実装

## 効果

- テックタッチのナビゲーションを顧客ごとに実装し、問合せ件数を**50%以上削減**。
- 本来、外部ベンダーに開発を委託するよりも開発コストを80%以上削減。
- 基本的な問い合わせが減ったのでサポート担当は安心して業務を遂行することができるようになった

## 課題

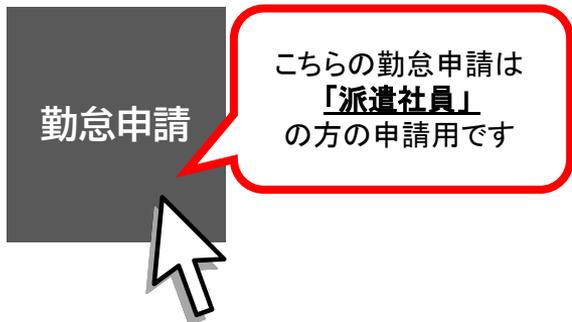
- 同じ文言のボタンが、画面によって意味が異なる仕様になっており、ユーザーの不満の声が多発
- 問合せが多く、カスタマーサクセス部門がコア業務よりも問合せ対応に追われていた
- 軽微なソースコード修正でも影響調査に時間がかかるため開発チームに要望を上げてもなかなか通らない

## 打ち手や実装のイメージ

同じ見た目のボタンに、カーソルを当てたときのみ表示させる補足説明を実装

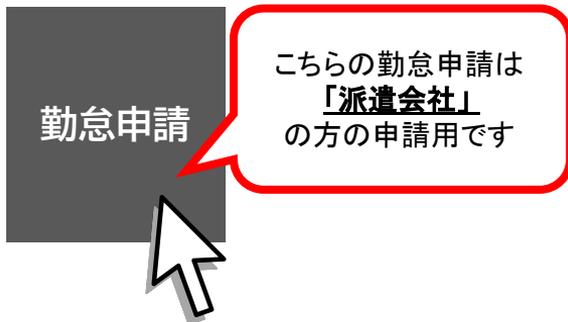
### 画面A

(派遣社員用)



### 画面B

(派遣会社用)



## 効果

- エンジニアの開発リソースやコストをかけずに、用語の意味を画面上に表示
- サポートセンターの説明コストに加え、関連する機能操作の問い合わせを大幅に削減

## 課題

- カード決済によって獲得したポイントを保有しているが有効活用されていないユーザーが一定数いた。
- このポイントの活用率を上げることで送決決済額の向上を狙いたいと試行錯誤していた。

## 打ち手や実装のイメージ

未使用ポイントがあるユーザーに対し通知&活用方法を案内



## 効果

- ポイントの活用率が**15%**上昇し、送決決済額も向上
- ソースコードの修正をせず、開発経験がない**カスタマーサクセスマネージャー自身で15分**で実装→リリースを実現

## 課題

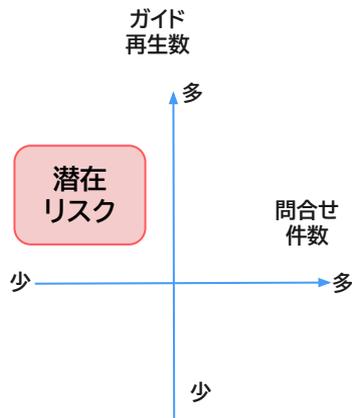
- 問合せ件数が少ない操作にも関わらず、テックタッチのガイド再生数が多い箇所があった
- この操作は、システム上で操作を行わなくても別途手作業でできてしまうものだったので問い合わせをするまでに至っておらず、潜在的に顧客満足度の低下を引き起こしているリスクがあった

## 打ち手や実装のイメージ

### 分析画面からガイドの再生数を確認

| ガイドタイトル   | 再生数 | ユーザー数 | 完了率   | ステータス | 編集日        |
|---|-----|-------|-------|-------|------------|
| すべて   | 721 | 10    | 90.6% |       |            |
| [UC] デモポータル(null=覧Ver)<br>【社内自主線用】E... > ユースケース | 266 | 8     | 95.5% | 公開    | 2023/12/11 |
| [UC] ガイドポータルデモ<br>【社内自主線用】E... > ユースケース         | 36  | 7     | 88.9% | 公開    | 2023/12/11 |
| [UC] アンケート<br>【社内自主線用】E... > ユースケース             | 34  | 5     | 88.2% | 公開    | 2023/10/13 |

### 潜在リスクの特定



## 効果

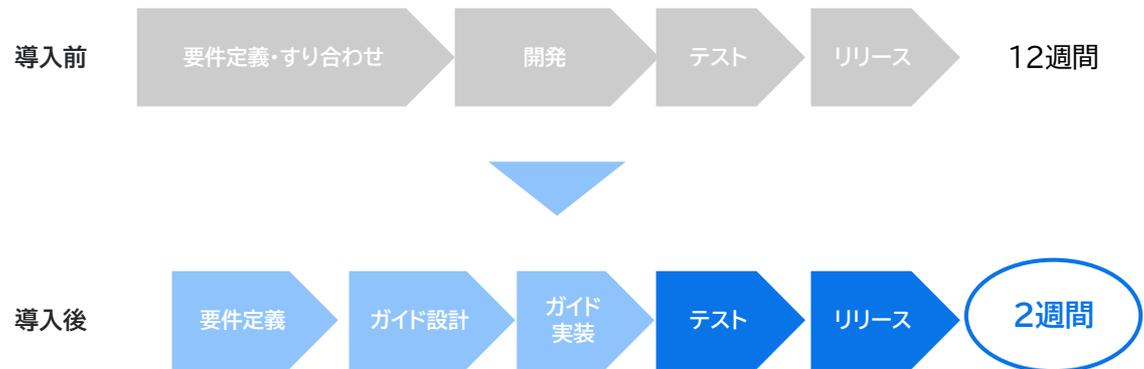
- テックタッチでガイド実装→再生数のモニタリングにより、顧客の「声なき声」を拾うことに成功
- 速やかに開発要望を提出、即開発で改修し、潜在的な解約・離脱のリスクを減らすことができた

## 課題

- 開発会社へ実装を依頼する工数・コスト・スピードがかかり、顧客要望にスピーディーに対応できていない。

## 効果

### 開発コストの大幅削減と実装スピードの高速化



- 機能開発の一部をテックタッチに置き換えることで、**開発会社に依頼していた開発コストを大幅に削減。**
- また、開発会社ではなく自社のCSがテックタッチで実装をすることで、**開発会社とのコミュニケーションコストを無くし、実装スピードも6倍に向上。**

## 課題

- カードの破損・読み取り不良のユーザーへのカード再発行数最が増えており、再発行コストが圧迫していた

## 打ち手や実装のイメージ

カード再発行の申請理由をお選びください

必須

申請理由



カードの破損・読み取り不良/サインミス/タッチ決済機能付きカードへの変更

以下に該当する場合はカードの再発行は不要です

- ・決裁が保留になりカードが使えなかった
- ・有効期限、セキュリティーコード入力エラー

このケースに該当する場合はこちらの画面から詳細をご確認ください

## 効果

- カード再発行の申請画面(左)にて、申請理由別に案内を表示。**本来は再発行不要のユーザーが再発行をせざるに済むようになった**
- 結果、**カード再発行数が30%以上削減**

# 教育関連システム(予備校)

## 課題

- わかりづらい表記や導線があっても自社内ですぐに改修ができない。開発を外部の開発会社に委託しているため、コスト高くスピードも遅い
- 結果、中々システム改修が進まないことが原因で、操作を完結せず離脱するユーザーが多発

## 打ち手や実装のイメージ



検索 検索結果：326件が該当しました

公開ステータス: 公開, 非公開

| <input type="checkbox"/> | ID ↑ | 画像 | 商品名 ↑    | コード ↑ ? | 価格       | 在庫数 ↑ |
|--------------------------|------|----|----------|---------|----------|-------|
| <input type="checkbox"/> | 326  |    | 商品デモ_001 |         | ¥ 10,000 | 無制限   |
| <input type="checkbox"/> | 325  |    | 商品デモ_001 |         | ¥ 10,000 | 無制限   |
| <input type="checkbox"/> | 324  |    | 001 商品デモ |         | ¥ 10,000 | 無制限   |

商品登録時に入力していない場合は表示されません。登録することにより検索しやすくなります。

## 効果

- 通常、左図のような実装だと外部委託業者への見積、社内稟議、作業で1か月ほどかかっていた改修が担当者自身で最短1日で可能になった
- テックタッチの「ファネル分析機能」を使い離脱率をモニタリングしたところ、実装前後で離脱率が20%以上改善されシステムの有効活用&顧客満足度の向上を実現

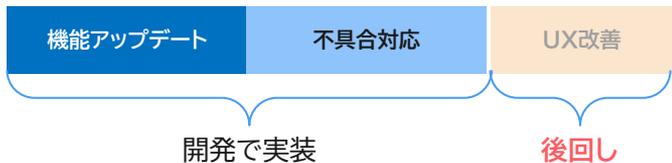
## 課題

- 新システムのリリース初期のため機能アップデートや不具合対応を優先
- ユーザーの増加に伴い申込に必要な操作方法等に関する問い合わせ対応が想定された
- UX改善の必要が出ていたものの、開発リソースが確保できない、予算制約で開発コストをかけられない

## テックタッチ導入前後の変化

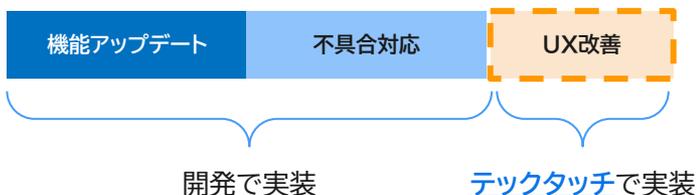
### UX改善が後回しに...

Before



### UX改善も同時に実現

After



## 実装イメージ

問合せが発生しそうな箇所に案内を設置して事前に問合せリスクを回避

認証キー設定

認証キーをお持ちの場合は入力後、「登録

認証キー ?

認証キーが不明な方は  
認証キーをお伝えした**弊社 営業担当**まで  
お問い合わせ下さい

## 効果

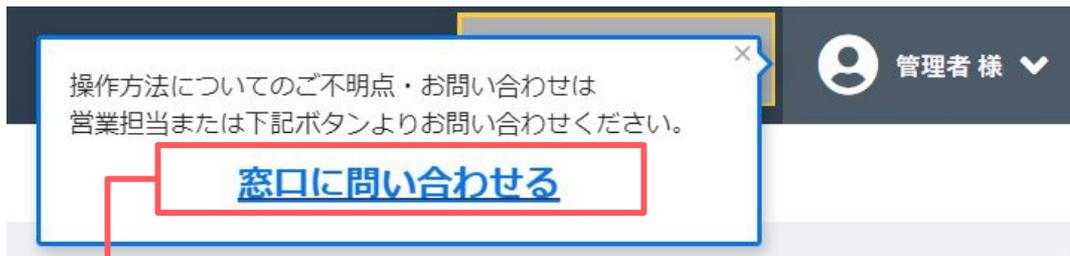
- 開発チームに依頼せず**サポート部門自身でUX改善**を実現
- 開発ではなくテックタッチで実装することで**開発コストを30%以上削減**

## 課題

- 新システムのリリース初期のため機能アップデートや不具合対応を優先
- ユーザーの増加に伴い操作方法等に関する問い合わせ対応が想定されていたが、開発優先度の兼ね合いで実装が追いついていなかった

## 打ち手や実装のイメージ

ソースコードを改修せずに問い合わせ窓口を実装



- コーポレートサイトの問い合わせフォーム
- サポートセンターへのメール作成画面などに遷移させる

## 効果

- 自社開発なしで問い合わせ窓口への導線を実装。わずか1日で実装完了。
- 同時に、他の開発案件のスピードを落とさずに、機能開発量を増やすことに成功

# 教育関連システム

## 課題

- 画面の変更は開発チームに依頼する必要があった。
- 急な障害があった際、即時情報通知ができずユーザーにエラーが発生、問い合わせ、クレームが発生

## 打ち手や実装のイメージ

### 出荷CSV登録 受注管理

CSVファイルをアップロード ?

CSVファイルを選択



ファイルを選択

選択されていません

一括登録を実行

管理

テックタッチでボタンを上書き配置し、障害発生している機能を強制的に利用できなくする



ファイルを選択

障害発生中

<お知らせ>

現在、システム障害によりCSV一括登録機能ができない状態が発生しています。  
ご迷惑をおかけしますが、復旧まで今しばらくお待ちください。

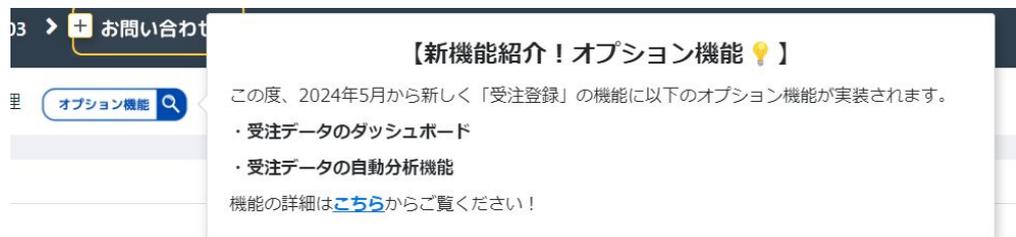
## 効果

- 障害発生時に、障害の認知させるときにツールチップをページ毎・場所ごとに表示
- 開発チームを介さず画面に情報を表示できる
- 利用できない機能を即時告知することでユーザーに余計な操作をさせずに済んでいる

## 課題

- オプション機能の認知率や活用度を高めるために、システム上でお知らせを実装しようとしても開発優先度が上がらず、中々実装されない
- 結果、リリースしたシステムが使いにくく顧客からの問合せが多発。企画部門が問合せ対応に追われる。

## 打ち手や実装のイメージ



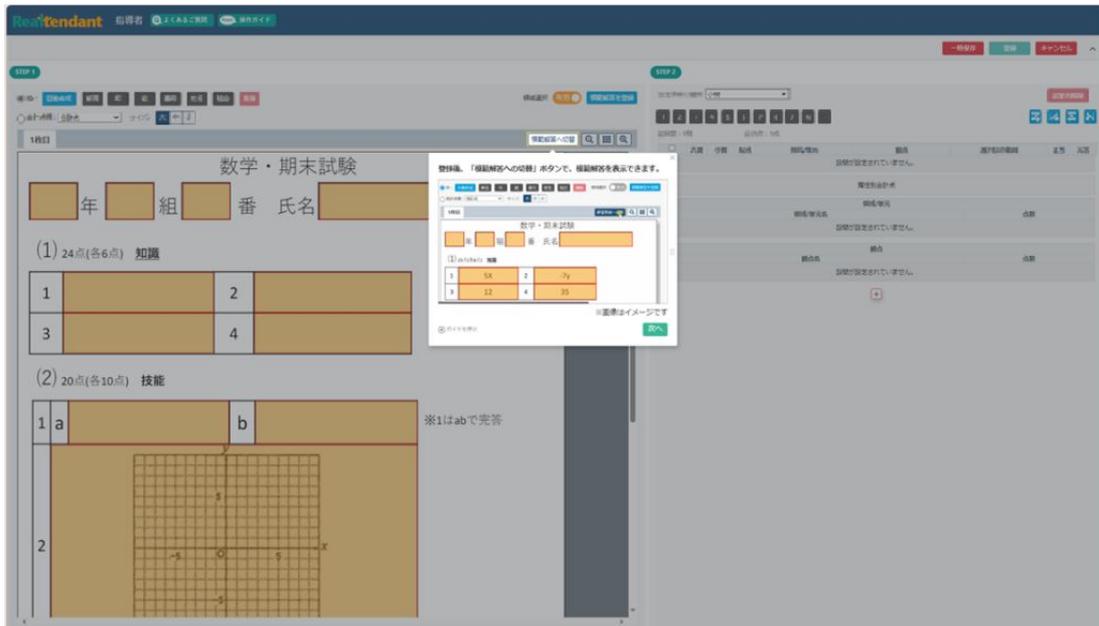
## 効果

- 企画部署のみで画面実装を完結し、顧客に情報伝達ができている
- **開発工数を軽減、リリース速度が向上（コミュニケーションコストを含む）**

## 課題

- システムの利用時期が限定的で、ユーザーが操作方法を忘れてしまう。
- はじめて利用するユーザーは、一部の操作に慣れるまで時間がかかることがあった。
- 基本事項の問い合わせが多く、コールセンターの業務負荷が増大していた。

## 打ち手や実装のイメージ



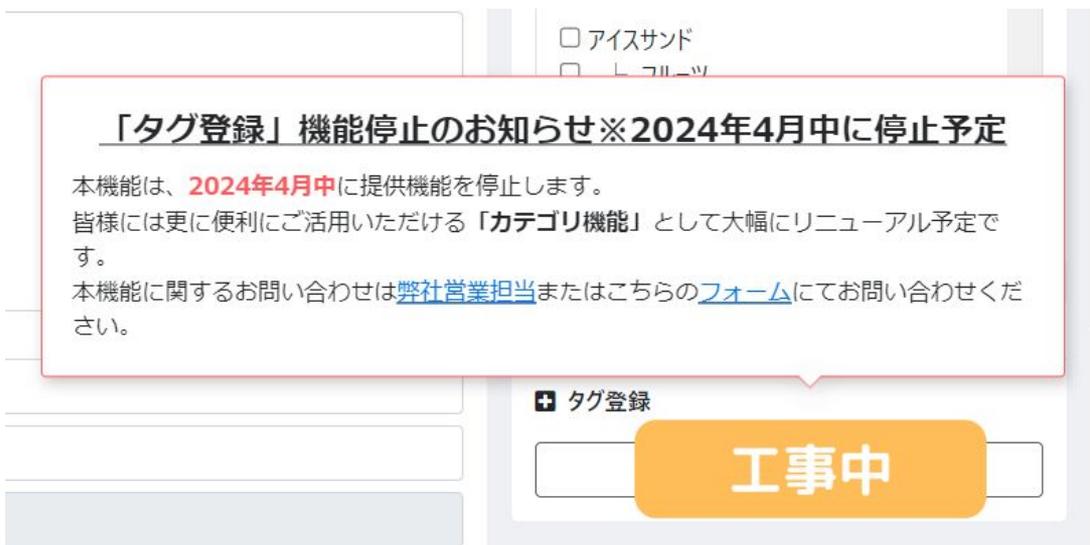
## 効果

- コールセンターへの問合せ数50%削減
- これまで導入支援にかかっていた工数をアップセルや営業の強化に変換
- 利用障壁を下げることでシステムの利活用促進に繋がった。

## 課題

- 刷新予定の機能が一時的に利用できなくなることによる問い合わせを未然に防ぎたい
- 開発コストをかけずにユーザーへの案内を表示したい

## 打ち手や実装のイメージ



## 効果

- ソースコードの改修不要、1日でユーザーへの案内を実装完了
- 機能の提供中止と刷新内容を適切に案内することで、問い合わせを未然に防ぐことに成功

## 課題

- ちょっとしたシステム画面の改修でも社内調整や稟議があり、かなり時間がかかる
- 効果が出ないと厳しくフィードバックされるので、ちょっとした改修でも時間がかかりリリースが遅い
- 要件定義にかなり時間がかかり、企画部門の工数負担が課題に

## 打ち手や実装のイメージ

「正しいと思うが証明できない」という仮説を開発なくすぐに実行できるため、時間とお金を節約しつつも最大の成果を創出可能に

### 通常

開発にお金も時間もかかるため、どのアイデア、施策を実施するか、慎重に精査。結果大半はお蔵入りに。

時間と可能性、そして成果が犠牲になりがち



### with テックタッチ

ソースコードの影響を気にせず、クイックにトライが可能。

そのため、全ての仮説を実行、検証することが可能。

机上議論ではなく実際のPDCAを回せるため、社員のモチベーションも向上。



## 効果

- 要件定義にかかる平均工数を約20%削減
- コア機能の要件定義に集中することができるようになった
- フロントの微修正はテックタッチで高速にPDCAを回すことで従来よりも機能利用率が向上

5

# 会社紹介



# テックタッチ株式会社

設立 2018年3月1日

〒104-0061

住所 東京都中央区銀座8-17-1

PMO銀座Ⅱ 5階

社員数 156名(2025年4月時点)※正社員のみ

累計資金調達額 24億円(シリーズBまでの累計)

DNX Ventures

アーキタイプベンチャーズ株式会社

DBJキャピタル株式会社

三菱UFJキャピタル株式会社

SMBCベンチャーキャピタル株式会社

みずほキャピタル株式会社

電通ベンチャーズ ほか

投資家情報  
一部抜粋

## 受賞歴

週刊東洋経済「すごいベンチャー100」選出(2020年)

グッドデザイン賞 受賞(2022年度)

SAP AWARD OF EXCELLENCE2023 Partner

Innovationアワード 受賞(2023年)

経済産業省「J-Startup」選出(2023年)

ITreview「ITreview Grid Award」2部門にて5期連続

「Leaders」受賞(2024年)

東京都ベンチャー技術大賞 特別賞受賞(2023年)

日本経済新聞社「スタ★アトビッチJapan」りそな銀行賞 受賞

(2024年)

デロイト トーマツ グループ「Technology Fast 50 2023

Japan」8位(2024年)

Great Place To Work® Japan調査「働きがいのある会

社」10位(2025年)



※1 弊社調べ、MAU換算  
※2 出典:株式会社アイ・ティ・アール「ITR Market View:コミュニケーション/コラボレーション市場2024」  
デジタル・アダプション・プラットフォーム市場動向:ベンダー別売上金額シェア(2022~2024年度予測)  
※3 2024年度予測  
※4 出所:株式会社アイ・ティ・アール「ITR Market View:コミュニケーション/コラボレーション市場2024」  
デジタル・アダプション・プラットフォーム市場-従業員10,000人以上:ベンダー別売上金額シェア(2022~2024年度予測)



## 井無田 仲 代表取締役 CEO

慶應義塾大学法学部、コロンビア大学MBA卒。  
新生銀行、ドイツ証券などで投資銀行業務に従事、上場企業の資金調達/M&A案件を数多く手がける。その後入社したユナイテッド社では、アプリ事業責任者、米国子会社代表としてアプリサービスのグロース/スケールを経験。フリーランスを経て、2018年3月にテックタッチを日比野と共同創業。楽しい未来を創るために日々奔走中。



## 中出 昌哉 取締役 CFO / CPO / AI Central 事業責任者

東京大学経済学部、マサチューセッツ工科大学MBA卒。  
野村證券株式会社に投資銀行業務に従事し、素材エネルギーセクターのM&A案件を数多く手掛ける。その後、カーライル・グループにて投資業に従事。ヘルスケア企業のバリュアップや、グローバル最大手の検査機器提供会社への投資等を担当。テックタッチでは、CFOに加えてプロダクト開発責任者、事業戦略立案等も担当。



## 西野 創志 執行役員 VP of Sales

TIS、SAP Japan、Slack Japan、セールスフォース・ジャパンにて、エンタープライズ企業向けの人事SaaSやSlackの販売責任者に従事。  
2022年3月にテックタッチに入社し、エンタープライズ企業のシェアを飛躍的に伸ばす。「すべてのユーザーがシステムを使いこなせる世界」を多くの企業に広げ、サービス・企業共に成長させるべく日々取り組んでいる。



## 岩淵 聖 執行役員 VP of Innovation

外資系分析ソフトウェアのSAS Institute Japanで新製品の立ち上げを担当した後、独立系コンサルティング会社の設立メンバーとしてBPR(業務改革)や経営管理のデザインを担当。2012年にSAP Japanに入社。テクノロジー・プラットフォーム事業の責任者として戦略立案・事業運営を担い、新領域の市場開拓やM&A製品の日本展開をリードし、成長に寄与。2025年2月、テックタッチ株式会社に入社。



## 日比野 淳 取締役 CTO

ファンコミュニケーションズ、ユナイテッドでCRMの開発、広告ネットワーク構築、大規模toCアプリの立ち上げからグロースを経験。  
その後、米国に赴任し現地スタートアップと協業しモバイルランチャーアプリの立ち上げに従事。  
2018年3月に井無田とテックタッチを共同創業。  
プロダクト戦略やロードマップの立案、策定からクオリティチェックまで幅広く担当。



## 垣畑 陽 執行役員 VP of Customer Success

京都大学総合人間学部卒。  
商船三井で法人営業、経営企画(投資審査等)を担当。その後マッキンゼーにて新規事業戦略立案から生産現場の地道なカイゼン活動まで幅広く従事。  
2020年8月にテックタッチにジョインし、カスタマーサクセス領域を管掌。  
顧客にプロダクトの価値を届けつつ、日本のエンタープライズ向けカスタマーサクセス領域の高度化に少しでも貢献するのが目標。



## 綾小路 雅典 執行役員 VP of Strategy / Alliance

2017年Box Japanに参画、営業企画とストラテジックセールスをリード。エンタープライズソフトウェア業界に3社で20年間従事。2020年よりSaaS スタートアップ企業向けにエンタープライズセールス組織の立上げおよび強化を支援、2023年5月にDNX Venturesベンチャーアドバイザーに就任。2024年4月、テックタッチ株式会社に入社しVP of Strategyに就任。



## 西田 雅俊 執行役員 Head of Corporate

慶應義塾経済学部卒。2008年に公認会計士試験に合格し、監査法人にて勤務。退所後エムスリー株式会社において経理業務、M&AおよびPMI業務に従事し、2020年からはアパレルスタートアップであるFABRIC TOKYOにてHead of Financeに従事。2023年5月よりテックタッチに入社し、2025年1月より執行役員 Head of Corporateに就任。

# 経営チーム



## 伊藤 宏樹 事業部CTO

北海道大学大学院理学研究科数学専攻卒。日本IBMのグループ会社にて生命保険会社に対する情報システムの企画・開発・保守にかかわる技術支援を経験。その後、市況情報配信システムの構築・運用、ミドルウェア開発やネットワーク・プロトコル実装に取り組む。2019年11月にテックタッチにジョインし、エンタープライズ SaaS に求められる機能の開発をリードする。2024年5月に事業部CTOに就任。



## 清水 豊子 常勤監査役

慶應義塾大学経済学部卒。公認会計士。監査法人トーマツ(現有限責任監査法人トーマツ)に入所し、国内監査部門にて多業種・様々な規模の企業の監査業務に関与。金融部門に異動し、大手外資系銀行、大手地方銀行、証券会社などの監査業務、J-Sox導入支援業務、IFRS導入支援業務(プロジェクトマネジャー)などを経験。2024年1月にテックタッチに監査役として入社。



## 中釜 由起子 Head of PR / Marketing

中央大学法学部卒。朝日新聞社で記者・編集、新規事業担当、Webメディア「telling」創刊編集長などを経て、2019年に株式会社ジーニーへ。マーケティング、全社広報・ブランディング統括などに従事。2023年4月にテックタッチにHead of PRとして入社。2023年10月より、Head of Marketing兼務。



ユーザー数

600万  
突破<sup>※1</sup>

DAP市場  
国内シェア

4年連続  
No.1<sup>※2</sup>  
2021 - 2024<sup>※3</sup>

※1 弊社調べ、MAU換算

※2 出典：株式会社アイ・ティ・アール「ITR Market View：コミュニケーション/コラボレーション市場2024」デジタル・アダプション・プラットフォーム市場動向  
：ベンダー別売上金額シェア（2022～2024年度予測）

※3 2024年度予測

# 顧客向けサービスへのテックタッチ導入企業（一部抜粋）

2025.05.26 更新

| EC       | マーケティング | 会員管理・コミュニティ | バックオフィス   | フロントオフィス | 金融・保険      | 不動産     |             |          |       |
|----------|---------|-------------|-----------|----------|------------|---------|-------------|----------|-------|
|          |         |             |           |          |            |         |             |          |       |
|          |         |             |           |          |            |         |             |          |       |
|          |         |             |           |          |            |         |             |          |       |
|          |         |             |           |          |            |         |             |          |       |
|          |         |             |           |          |            |         |             |          |       |
|          |         |             |           |          |            |         |             |          |       |
|          |         |             |           |          |            |         |             |          |       |
|          |         |             |           |          |            |         |             |          |       |
|          |         |             |           |          |            |         |             |          |       |
|          |         |             |           |          |            |         |             |          |       |
|          |         |             |           |          |            |         |             |          |       |
|          |         |             |           |          |            |         |             |          |       |
|          |         |             |           |          |            |         |             |          |       |
| 人材サービス   | 建設      | 運輸・物流       | 製造業       | 自動車      | 小売・飲食・メディア | 教育・スポーツ | 医療・ヘルスケア・介護 | ESG・SDGs | 公共    |
|          |         |             |           |          |            |         |             |          | デジタル庁 |
|          |         |             |           |          |            |         |             |          | 農林水産省 |
|          |         |             |           | 大手二輪メーカー |            |         |             |          |       |
|          |         |             | 大手EDIサービス |          |            |         |             |          |       |
|          |         |             |           |          |            |         |             |          |       |
| 大手人材広告企業 |         |             |           |          |            | ネットワーク  |             |          |       |

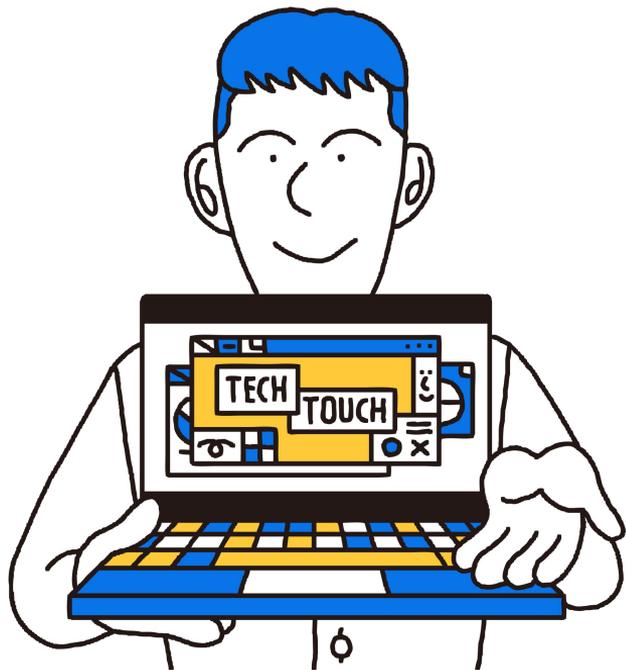
※公表許可いただいている企業・団体様より一部抜粋(企業名、サービス名については50音順)

# 導入企業一覧(社内システムに導入)

2025.04.15 更新



※公表許可いただいている企業・団体様より一部抜粋(企業名、サービス名については50音順)



テックタッチでできることは、他にも沢山あります。  
他の事例をご覧になりたい方は、下記よりご覧ください。

他の事例も見てみる

デモや「テックタッチ」を使った事例の紹介や相談など、  
詳細をご知りになりたい方は、下記よりお問合せください。

詳しく話を聞く



すべてのユーザーが  
システムを使いこなせる世界に

システム導入だけで終わらせない、利活用のためのDXプラットフォーム