



新規事業責任者やスタートアップ社長にオススメ！

初めてのSaaS開発の落とし穴

～ビジネス側も最低限知っておきたい開発や技術のこと～

本日のアジェンダ

1 要件定義における落とし穴

2 開発設計における落とし穴

3 開発実装における落とし穴

本資料がおすすめの方



本資料では、ビジネス側の方々向けに、最低限知っておきたい開発や技術のことをお伝えします

ターゲット

これからSaaS事業をやろうと思っている
「新規事業責任者」や「スタートアップの代表の方」向け

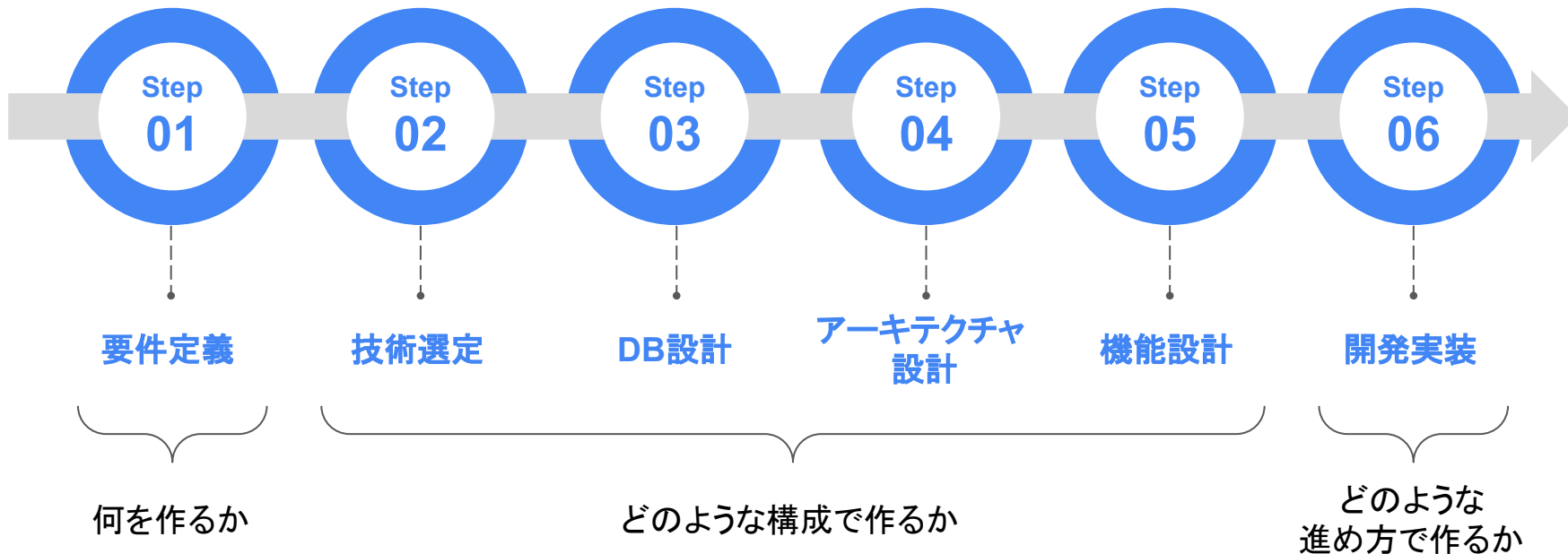
目的

ビジネス側の方々が最低限知っておきたい
開発や技術のことを知ること

一般的な開発の流れ



SaaS開発における失敗を防ぐためには、各工程において、「ビジネスとエンジニアがすり合わせをすること」が重要です



アジェンダ①

要件定義における落とし穴について

要件定義のよくあるミス



要件定義におけるよくあるミスとして、脳内でなんとなくイメージしている機能一覧から議論してしまうことがあげられます



脳内でなんとなくイメージしている
機能一覧から議論してしまう

ログイン機能が欲しい、CSV出力が欲しい、分析機能が欲しい、など

要件定義のよくあるミス



その結果として、機能の抜け漏れや優先順位のミスが発生してしまうリスクがあります



脳内でなんとなくイメージしている
機能一覧から議論してしまう

ログイン機能が欲しい、CSV出力が欲しい、分
析機能が欲しい、など



機能の抜け漏れや優先順位のミスが発生してしまう

要件定義の正しい入口



よくあるリスクを防ぐには、機能を考える前に、業務フローから整理することが重要です



機能の抜け漏れや優先順位のミスを防ぐには・・・？



正しい入口

機能を考える前に、業務フローから整理する

どのような業務をどう改善したいのか、という一連の流れから整理することで、機能や画面構成の洗い出しに漏れがなくなる

要件定義の正しい入口



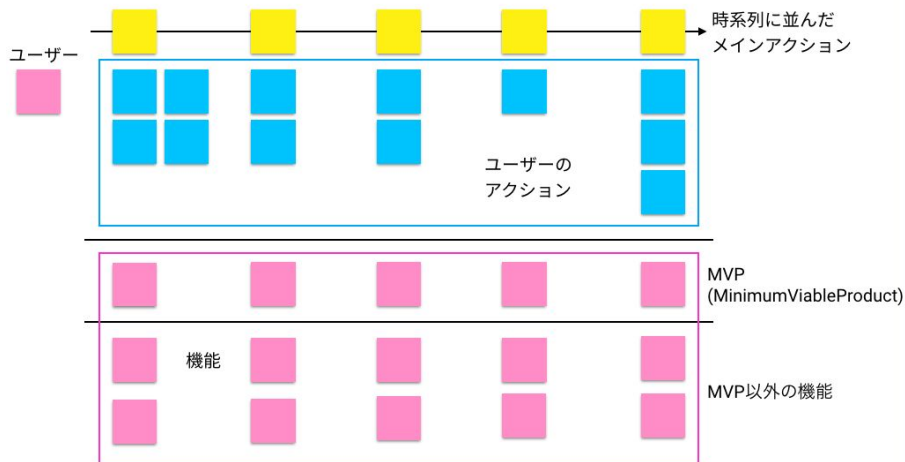
そのために便利なツールとして、「ユーザーストーリーマップ」があげられます

ユーザーストーリーマップ

各業務に対して、

- ① 誰が
- ② いつ
- ③ 何を
- ④ どのように行動している
- ⑤ どのような課題が発生しているか

の全体像を洗い出した上で、機能を整理する手法



<https://note.com/faiman/n/n8f69ef75aa06>

要件定義の正しい入口

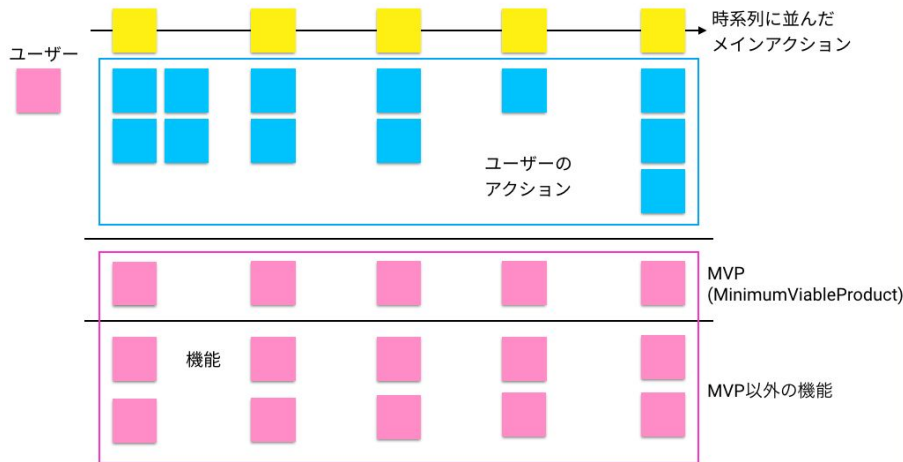


「ユーザーストーリーマップ」の活用tipsとして以下があげられます

ユーザーストーリーマップの tips

・「⑤ どのような課題が発生しているか」を整理する時に、どこに時間がかかっているか、何が属人化しているか、どんなミスが起きやすいか、を整理することで、機能の優先順位を検討できる

・優先順位を考えるときは、何を作るかだけでなく、「初期フェーズでは作らないことを決めること」も重要。最小限の機能(MVP)で早く出し、ユーザーの反応を見てから「正解」を作る



<https://note.com/faiman/n/n8f69ef75aa06>

アジェンダ②

開発設計における落とし穴について

開発設計(技術選定)のよくあるミス



開発設計におけるよくあるミスとして、「安く早く作れるから」という理由だけで技術を決めてしまうことがあげられます



「安く早く作れるから」という理由だけで技術を決めてしまう

開発会社Aの方が開発会社Bよりも、費用が300万円安くて、リリースも2か月早い！

開発設計(技術選定)のよくあるミス



その結果として、将来的に、ユーザー数 / 機能数 / データ処理が増加したときに、機能追加や修正がしづらくなってしまいうリスクがあります



「安く早く作れるから」という理由だけで技術を決めてしまう

開発会社Aの方が開発会社Bよりも、費用が300万円安くて、リリースも2か月早い！



将来的に、ユーザー数 / 機能数 / データ処理が増加したときに、機能追加や修正がしづらくなっていく

開発設計(技術選定)の正しい入口



よくあるリスクを防ぐには、利用技術による今と将来のトレードオフを理解した上で決めることが重要です



正しい技術選定をするためには・・・？



正しい入口

利用技術による今と将来のトレードオフを理解した上で決める

「今欲しいものを早く安く作れる技術」と「将来の機能追加 / 改善に向いてる技術」は異なるので、「PoC用のプロダクト」が欲しいのか、
「中長期の改善前提のプロダクト」が欲しいのかを整理

開発設計(技術選定)の正しい入口



開発言語による大まかな特徴は以下となります

フロントエンド

- 安くて早い技術



- 将来の拡張性が高い技術



バックエンド

- 安くて早い技術



- 将来の拡張性が高い技術



- 特定技術と相性が高い技術
(AI/フロントエンド/複数技術利用など)



開発設計(DB設計)のよくあるミス



開発設計におけるよくあるミスとして、短期的な開発速度やデータ処理速度を重要視してしまうことがあげられます



短期的な開発速度や
データ処理速度を重要視してしまう

なるべく早くリリースしたいので、リリース時点の
プロダクトが、最低限のデータ処理をできていれ
ばOK!

開発設計(DB設計)のよくあるミス



その結果として、将来的に、新しい種類のデータの取得や連携が難しくなってしまうリスクがあります



短期的な開発速度や
データ処理速度を重要視してしまう

なるべく早くリリースしたいので、リリース時点の
プロダクトが、最低限のデータ処理をできていれ
ばOK!



今に最適化された DB設計 (データテーブルを1つにまとめる、
今の画面構成に最適化したデータテーブルを設計するなど) **により、**
将来的に新しい種類のデータの取得や連携が難しくなる

開発設計(DB設計)の正しい入口



よくあるリスクを防ぐには、将来的なビジネス構想や蓄積して活用したいデータ種類から逆算してDBを設計することが重要です



将来的にも、様々なデータを増やせる DBを設計するには・・・？



正しい入口

DB設計は不可逆、将来的なビジネス構想や蓄積して活用したいデータ種類から逆算して DBを設計する

SaaSビジネスにおける価値の源泉は「データ」なので、
中長期で、どのような業務課題を解決したいのか、
どのようなデータが必要で、どのように活用したいのかを踏まえて設計する

開発設計(アーキテクチャ設計)のよくあるミス



開発設計におけるよくあるミスとして、「システム構成の話は難しいので、エンジニアに任せればいい」と考えて、完全に委ねてしまうことがあげられます



「システム構成の話は難しいので
エンジニアに任せればいい」と考えて、完
全に委ねてしまう

専門的な技術の話はよく分からないので、信用
しているから完全にお任せしたい！

開発設計(アーキテクチャ設計)のよくあるミス



その結果として、中長期で開発コストの増大、バグの大量発生、開発速度が遅延してしまうリスクがあります



「システム構成の話は難しいのでエンジニアに任せればいい」と考えて、完全に委ねてしまう

専門的な技術の話はよく分からないので、信用しているから完全にお任せしたい！



中長期で開発コストの増大、バグの大量発生、開発速度の遅延

開発設計(アーキテクチャ設計)の正しい入口



よくあるリスクを防ぐには、システム構成の詳細は任せてよいが、「ビジネス方針」を必ず伝えることが重要です



中長期で開発コストの増大、バグの大量発生、
開発速度の遅延を防ぐには・・・？

正しい入口

システム構成の詳細は任せてよいが、
「ビジネス方針」を必ず伝える

アーキテクチャ設計は「どう作るか」ではなく「どこが変わりやすいか」を決める設計なので、中長期のビジネス方針の変化を伝えておくことで、変更が発生しやすいものを最初から分けて、コスト / バグ / 開発納期を最小化できる

開発設計(機能設計)のよくあるミス



開発設計におけるよくあるミスとして、機能名で議論して設計を進めてしまうことがあげられます



機能名で議論して設計を進めてしまう

分析機能が欲しい、ダッシュボードが必要、管理画面を作りたいなど

開発設計(機能設計)のよくあるミス



その結果として、必要以上にリッチな機能が出来上がってしまうリスクがあります



機能名で議論して設計を進めてしまう

分析機能が欲しい、ダッシュボードが必要、管理画面を作りたいなど



必要以上にリッチな機能が出来上がってしまうリスクが高い

開発設計(機能設計)の正しい入口



よくあるリスクを防ぐには、何の機能が欲しいかではなく、何を実現したいかで議論することが重要です



必要以上にリッチな機能が
出来上がってしまうリスクを防ぐには・・・？

正しい入口

何の機能が欲しいかではなく、何を実現したいかで議論する

誰のどの業務のどの課題を解決するのか、から議論することで、
不要な機能を減らした最小コストの設計案がエンジニア側から出てきやすい

開発設計(機能設計)の正しい入口



正しい議論の進め方のイメージは以下となります

イメージ①

分析機能が欲しい

NG!

誰が、どの数値を、
どのような意思決定に使う
ための機能が欲しい

OK!

それなら分析画面は不要で、
CSV出力できればいい…?

イメージ②

CSV出力が欲しい

NG!

誰の、どの業務の、
どのデータを、誰に、
どの頻度で渡したい

OK!

その頻度ならCSV出力は不要で、
エンジニアがDLすればいい…?

アジェンダ③

開発実装における落とし穴について

開発実装のよくあるミス



開発実装におけるよくあるミスとして、最初から完璧な機能構成を目指してしまうことがあります



最初から完璧な機能構成を
目指してしまう

CSV出力機能もあった方がいい気がするし、分析機能もあった方がいいし……

開発実装のよくあるミス



その結果として、無駄な機能開発コストが増えてしまうリスクがあります



最初から完璧な機能構成を
目指してしまう

CSV出力機能もあった方がいい気がするし、分
析機能もあった方がいいし……



無駄な機能開発コストが増えてしまう

開発実装の正しい入口



よくあるリスクを防ぐには、自分の考えが変わる前提で進めてみるのが重要です



無駄な機能開発コストを減らすためには・・・？



正しい入口

自分の考えが変わる前提で進めてみる

初期段階では正しそうに見えるが、顧客からの声を聴く中で、開発したい機能はどんどん増えるし、いらない機能にも気づいていく

開発実装の正しい入口



各開発手法の違いは以下となります

アジャイル開発

OK!

小さく作って、使ってもらい、
学びながら改善する開発手法

- ・短いサイクルで開発リリース
- ・仕様変更に強い
- ・ユーザーの利用データやフィードバックを反映しやすい

ウォーターフォール開発

NG!

最初に要件の全体像を固めてから、
開発を進める開発手法

- ・最初に要件を固めて一気に開発
- ・仕様変更に弱く手戻りコストが高い
- ・完成するまでユーザーに使ってもらえない

サービス紹介

お客様のよくある課題

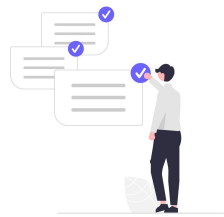


プロダクト開発やシステム開発における、Quality(品質)・Cost(価格)・Delivery(納期)にお悩みはありませんか？

プロダクト開発における課題

1

プロダクト開発の最適な進め方やあるべき姿が分からない



2

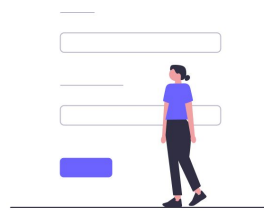
プロダクト開発にかかるコストや納期が大きい



システム開発における課題

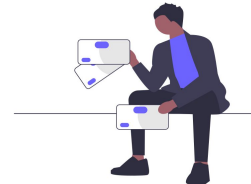
3

システム開発の最適な進め方やあるべき姿が分からない



4

システム開発にかかるコストや納期が大きい

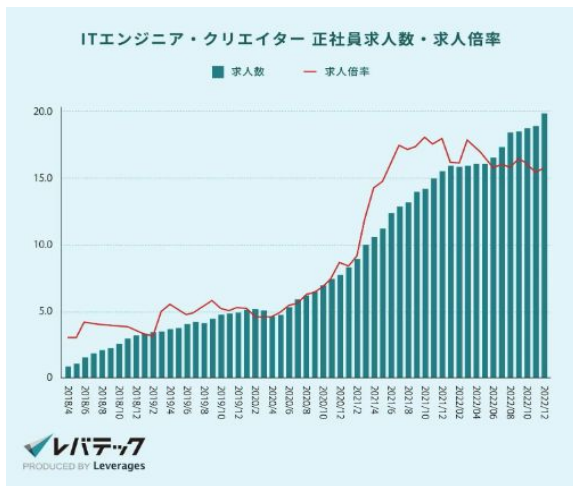


お客様のよくある課題



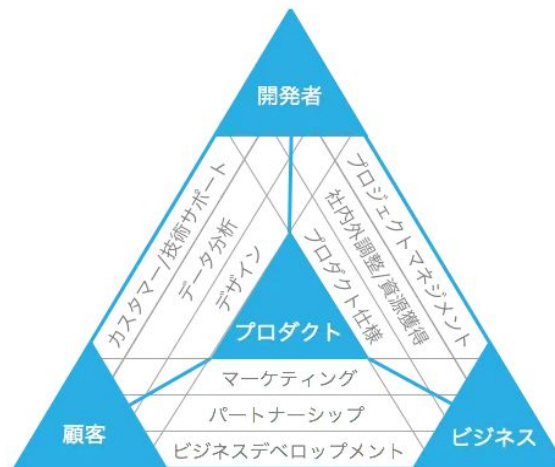
一方で、プロダクト開発やシステム開発のノウハウをもつ人材は採用が難しく、また、自己学習するには必要な知識が多く時間がかかります

開発人材の求人倍率 *1



* 1: ITエンジニア・クリエイターの求人倍率、15.8倍と高止まり続く | レバレッジ株式会社のプレスリリース
<https://prtimes.jp/main/html/rd/p/000000531.000010591.html>

開発に必要とされる知識 *2



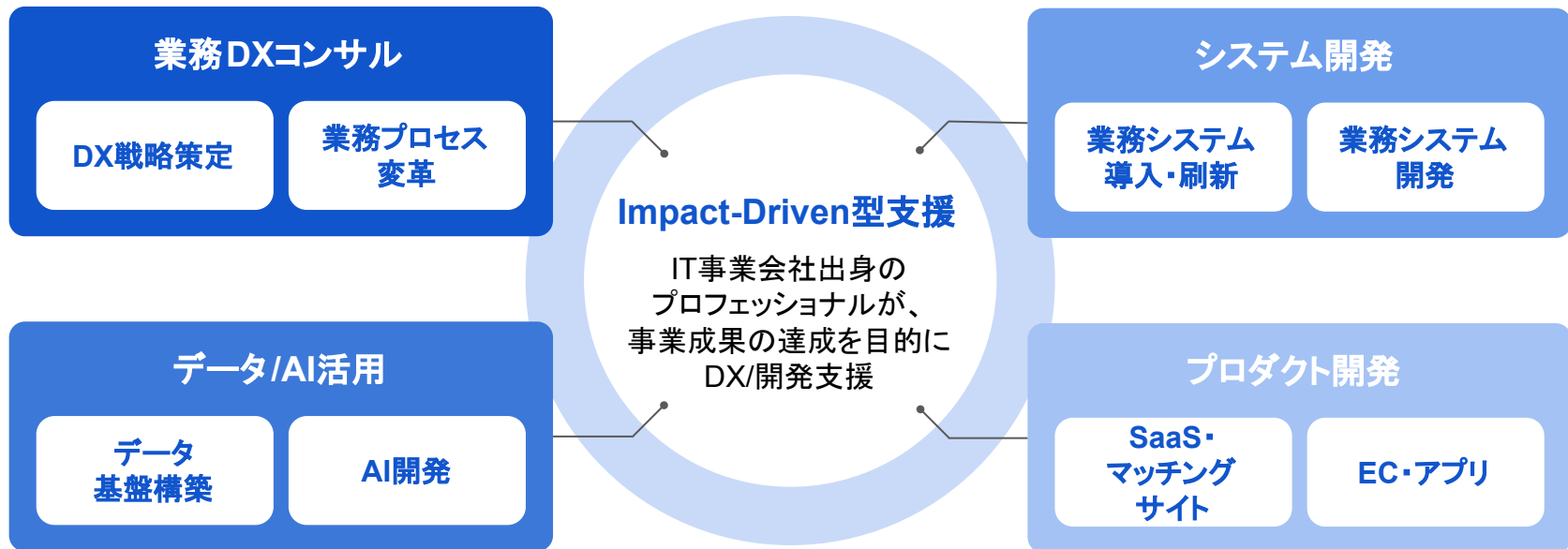
* 2: プロダクトマネジメントトライアングルと各社の PM の職責と JD | by Taka Umada | Medium
<https://tumada.medium.com/product-management-triangle-job-description-d18d1855ef65>

事業内容 ～Impact-Driven型支援～



IT事業会社出身のプロフェッショナルが、「Impact-Driven型支援」を通じて、お客様のDX/開発支援を成功へ導きます

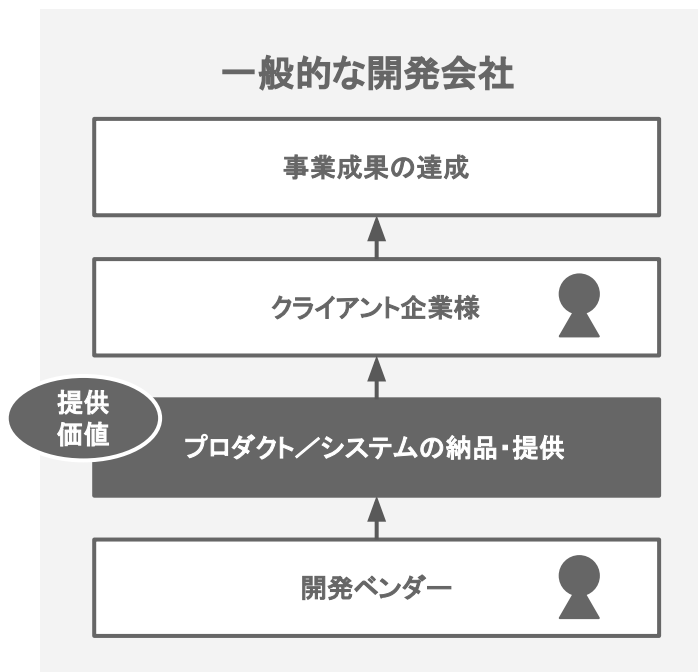
RIPLAが提供するサービス



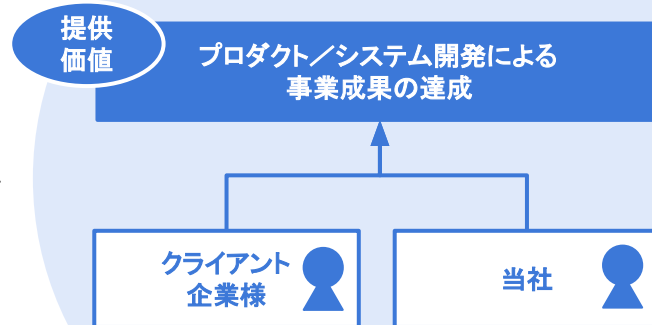
事業内容 ～Impact-Driven型支援～



「Impact-Driven型支援」では、プロダクトやシステムの納品・提供をゴールとせず、クライアント企業様と同じ目線で、事業成果の達成を目的としたDX/開発支援をいたします



RIPLAの "Impact-Driven型支援"



事業内容 ～Impact-Driven型支援～



具体的な支援手法として、ユーザー価値や組織浸透性を意識した要件定義、中長期の拡張性や内製化を見据えたシステム設計により、事業成果を最大化いたします

事業成果を意識した要件定義

- ✓ 事業成果やユーザー価値を意識した製品の機能企画や UI/UX設計
- ✓ 現場業務や組織浸透性を意識したシステムの要件定義

中長期目線で柔軟なシステム設計

- ✓ 中長期の拡張性や運用を見据えたアーキテクチャ設計
- ✓ 将来的な内製化に向けた採用しやすい技術選定

“Impact-Driven型支援”により、事業成果を最大化

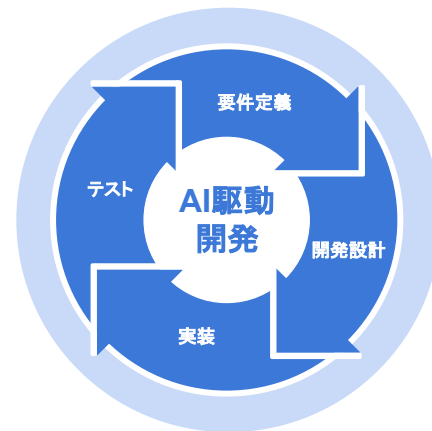
事業内容 ～Boxシリーズ×AI駆動開発～



「Boxシリーズ」による標準機能の高速開発と、「AI駆動開発」による独自機能の柔軟な実装を組み合わせることで、低コスト・短期間で開発を実現いたします

Boxシリーズ

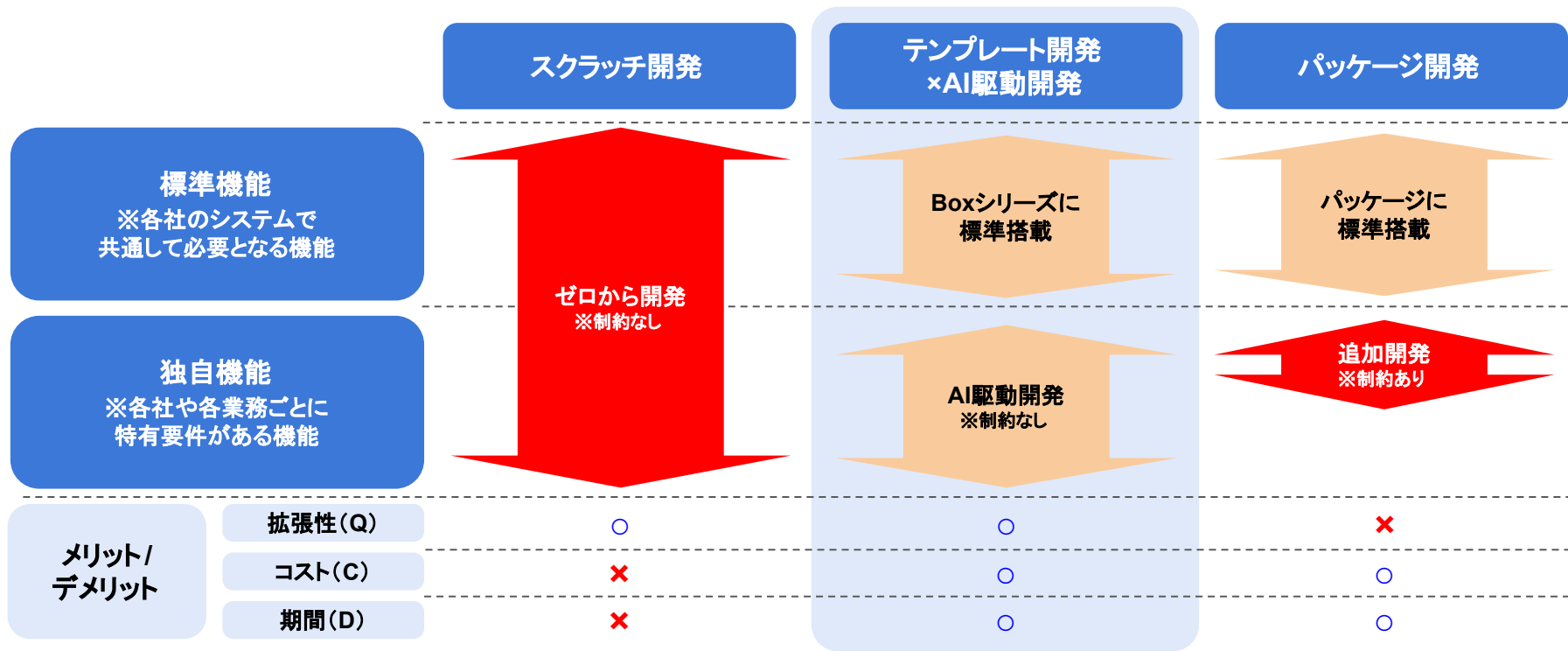
AI駆動開発



事業内容 ～Boxシリーズ×AI駆動開発～



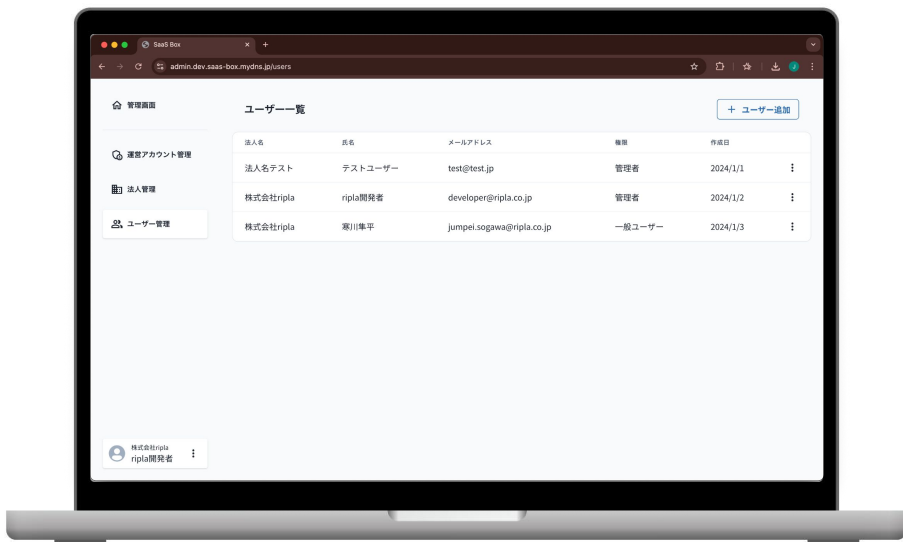
テンプレート開発(=Boxシリーズ)とAI駆動開発を併用することで、Q(拡張性)・C(コスト)・D(納期)の三点両立を実現いたします



事業内容 ～Boxシリーズ～



モダンな技術とUI/UXで各画面の開発・実装をしているため、そのまま使用することが可能です



モダンな技術とUI/UXで
標準機能を搭載

事業内容 ～Boxシリーズ～



Boxシリーズに搭載されている機能例は以下となります(以下一部)

<p>Box名称</p>	 受発注管理 Box	 在庫管理 Box	 配送管理 Box	 業務システム Box	 生成AI Box
<p>機能例</p>	<ul style="list-style-type: none"> ・ログイン機能 ・ユーザー管理/権限管理 ・マスタ管理 (受注/発注/商品) ・受注/発注管理 ・需給予測 ・インフラ構築 	<ul style="list-style-type: none"> ・ログイン機能 ・ユーザー管理/権限管理 ・マスタ管理 (出荷先/入荷元/商品) ・出荷/入荷管理 ・在庫管理 ・インフラ構築 	<ul style="list-style-type: none"> ・ログイン機能 ・ユーザー管理/権限管理 ・マスタ管理 (届け先/商品) ・配送管理 ・配送最適化機能 ・インフラ構築 	<ul style="list-style-type: none"> ・ログイン機能 ・ユーザー管理 ・権限管理 ・ユーザー向け画面 ・管理画面 ・インフラ構築 	<ul style="list-style-type: none"> ・AI チャット機能 ・RAG基盤 ・ベクトルDB基盤 ・検証用画面 ・バッチ処理 ・インフラ構築
<p>Box名称</p>	 SaaS Box	 マッチングサイト Box	 EC Box	 アプリ Box	 LINE Box
<p>機能例</p>	<ul style="list-style-type: none"> ・ログイン機能 ・ユーザー管理/権限管理 ・法人管理 ・ユーザー向け画面 ・管理画面 ・インフラ構築 	<ul style="list-style-type: none"> ・ログイン機能 ・ユーザー管理/権限管理 ・マッチング機能 ・ユーザー向け画面 ・管理画面 ・インフラ構築 	<ul style="list-style-type: none"> ・ログイン機能 ・ユーザー管理/権限管理 ・マスタ管理(顧客/商品) ・受注管理 ・ユーザー向け画面 ・管理画面 ・インフラ構築 	<ul style="list-style-type: none"> ・ログイン機能 ・ユーザー管理/権限管理 ・プッシュ通知機能 ・アプリUI(iOS/Android) ・管理画面 ・インフラ構築 ・リリース申請 	<ul style="list-style-type: none"> ・LINEログイン ・リッチメニュー基盤 ・LINE配信機能 ・公式アカウント情報連携 ・ユーザー管理/権限管理 ・インフラ構築

RIPLAの対応技術



AI・モダンな技術・クラウド型・APIなどの最新技術を活用することで、新規性のある取り組みや、拡張性や柔軟性の高い開発に対応可能です



AIを活用した業務効率化や 新たな価値創出

AI技術を活用することで、業務プロセスの自動化、レコメンド機能、自然言語処理、画像解析などが可能になります。



モダンな技術を活かした 高品質な開発

最新のモダン技術を活用することで、ユーザビリティの高いUI/UX設計、保守性の高いアーキテクチャを実現します。



クラウド型で拡張性のある システム構築

クラウドインフラを活用することで、柔軟かつ拡張性の高いシステムを構築し、初期投資を抑えることも可能です。



API連携による柔軟な 外部サービス連携

外部システムやSaaSとのスムーズな連携を前提としたAPI設計を行うことで、機能追加や業務連携が容易になります。

RIPLAの対応技術



また、お客様のニーズに応じて、幅広い技術を取り扱うことができます

フロントエンド



バックエンド



AI関連



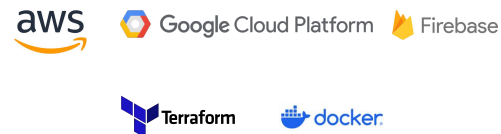
アプリ



DB関連



クラウド技術・その他



これまで多数の業務DXコンサル、データ/AI活用、システム開発、プロダクト開発の支援実績がございます

業務DXコンサル

- ・販売 / 顧客管理システムの刷新
- ・大手建設企業のDX戦略策定
- ・金属商社企業の基幹システム刷新
- ・広告SaaSのマルチプロダクト構想策定
- ・印刷EC新規事業の構想策定など

データ/AI活用

- ・訪問看護のAI FAQシステム
- ・AIデータ分析SaaS
- ・ホテル向けのデータ基盤構築
- ・AI画像解析による商品棚在庫管理
- ・コーチングのAIチャットボットなど

システム開発

- ・物流系WMS / OMS
- ・ガソリンスタンド受発注管理システム
- ・営業ナレッジマネジメントシステム
- ・訪問看護のスケジュール管理システム
- ・ドローン機器管理システムなど

SaaS

- ・研修管理 / 学習管理SaaS
- ・LINEチャットボットSaaS
- ・SFA / CRM SaaS
- ・設備保全管理SaaS
- ・採用管理SaaS など

マッチングサイト・EC

- ・インフルエンサーマッチングサービス
- ・求人検索メディア
- ・美容師マッチングサービス
- ・ビジネスマッチングサービス
- ・オンラインチケット販売EC など

アプリ

- ・献立共有アプリ
- ・旅行ガイドブックアプリ
- ・医療用スマホアプリ
- ・スポーツゲームアプリなど



RIPLA