

アシストソリューション研究会中日本 2022年度分科会発表会

＼DXに乗り遅れるな！／ ノーローコード開発ガイドの 策定とその有用性

ノーコード／ローコード開発と適用方針 ～適用方針に関する研究～ 分科会
チーム・ノコロコ（青木、森山、田中、武田、小久保、宮澤、松尾）



目次

I. 分科会紹介と活動概要

- 分科会紹介
- 活動概要

II. ノーコード/ローコードツールとは

- 市場をみる
- ツールを触ってみる
- ツールを比較する / 課題がわかる
- 研究テーマを決める

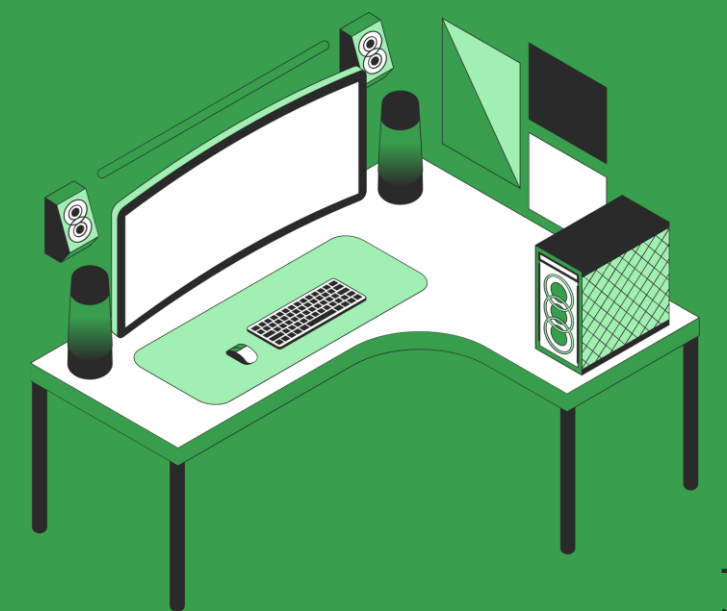
III. スタートガイドとその有用性

- スタートガイドの概要
- ノーコード/ローコード開発の勘所
- ガイドの反響をみる
- アンケート結果を分析する
- 改良ポイントを決める / ガイドを改良する

IV. 研究結果と総括

- まとめ
- 総括
- 展望

I. 分科会紹介と活動概要



分科会紹介




知識・経験が多様なチーム“ノコロコ”メンバー

 リーダー



リコージャパン株式会社
青木 優太

 サブ
リーダー



トランコムITS株式会社
森山 欽司



鈴与システムテクノロジー株式会社
武田 雅美



愛知県経済農業
協同組合連合会
田中 佑治



株式会社メイテツコム
小久保 響子



株式会社アシスト
宮澤 祐平

事務局
窓口



株式会社アシスト
松尾 成昭

知識・経験が多様なチーム“ノコロコ”メンバー

🏆 リーダー



楽々FW使ってます

🏆 サブ
リーダー



高速開発ツール
色々使ってます



楽々FW・WF、
ASTERIA Warp使ってます



PMメインで活動中
ノーローコードは
未経験



スクラッチ開発メイン
ノーローコードは
未経験



運用管理ツール担当
ノーローコードは
未経験

事務局
窓口



前職はPG,今はPM
ノーローコードは
未経験

まずはミッションで想いを1つに

ノーコード/ローコード開発の現状と、各メンバーの知見やこの分科会に対する想いを掛け合わせて、当分科会のミッションを以下としました。

“「本分科会で研究した内容をガイドラインとしてアウトプットし、私達と同じようにノーコード/ローコードツール活用に悩める人達に貢献したい！」



活動概要



私たちの研究テーマ

事務局からのテーマ

開発
／
運用

ノーコード／ローコード 開発と適用方針

～適用方針に関する研究～

東
中
西

コーディングが不要あるいは最小限のコーディングでアプリケーションを開発する「ノーコード／ローコード開発」が日本でも使われ始めています。ノーコード／ローコードが必要とされる背景を探るとともに、その有用性や特性・適用分野などを調査し、野放し状態で開発が進まないよう、適用方針案を策定することで本格的なノーコード／ローコード開発時代の到来に備えます。

- ノーコード／ローコードが必要とされる背景
- ノーコード／ローコード開発が適用できる／できないシステム
- ノーコード／ローコード開発の適用方針の策定

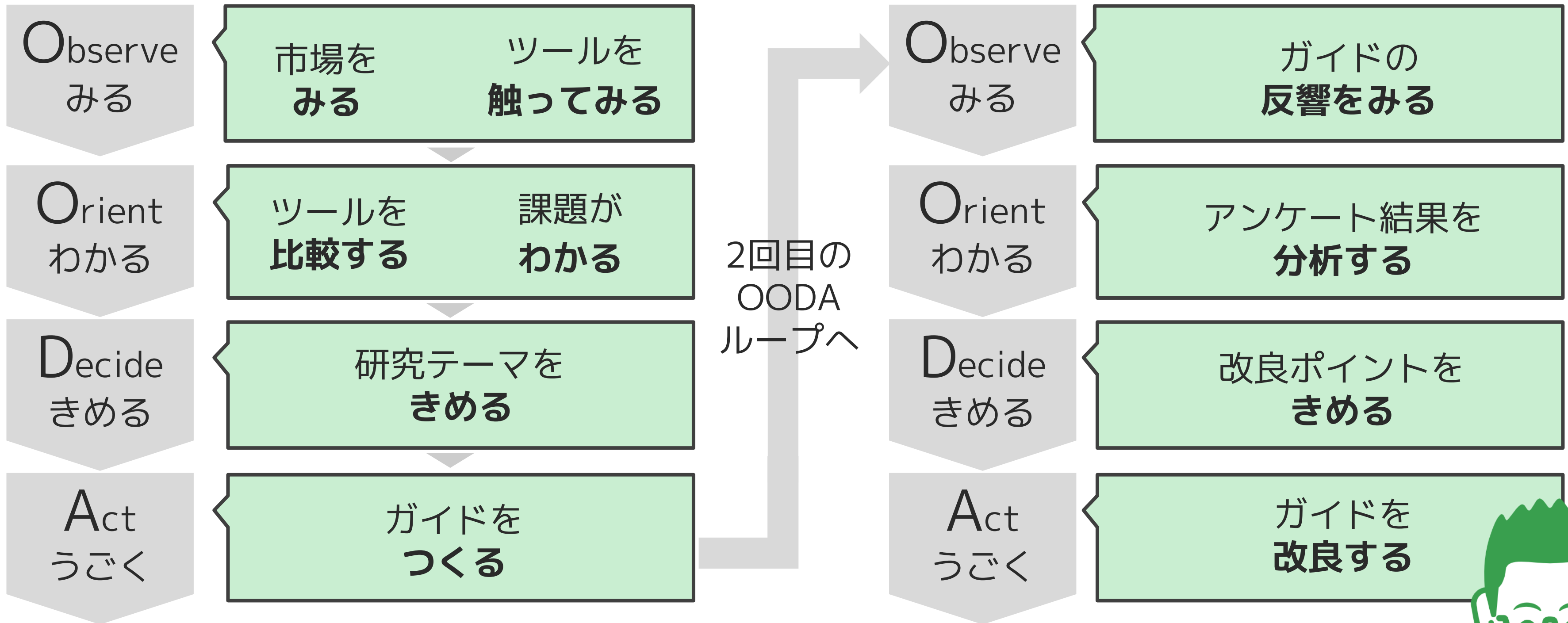
私たちの研究テーマ

「ノーコード／ローコードツールのスタートガイド策定」
ツール選定だけでなく開発・運用も含めた勘所をスタートガイドとして
まとめてリリースし、その有用性を検証する



活動の流れ

私たちは意思決定手法のOODAループを意識して活動を進めました。



「みる」「わかる」を通し、課題を「きめる」
その課題を解決するために「うごく」



数字でみる成果

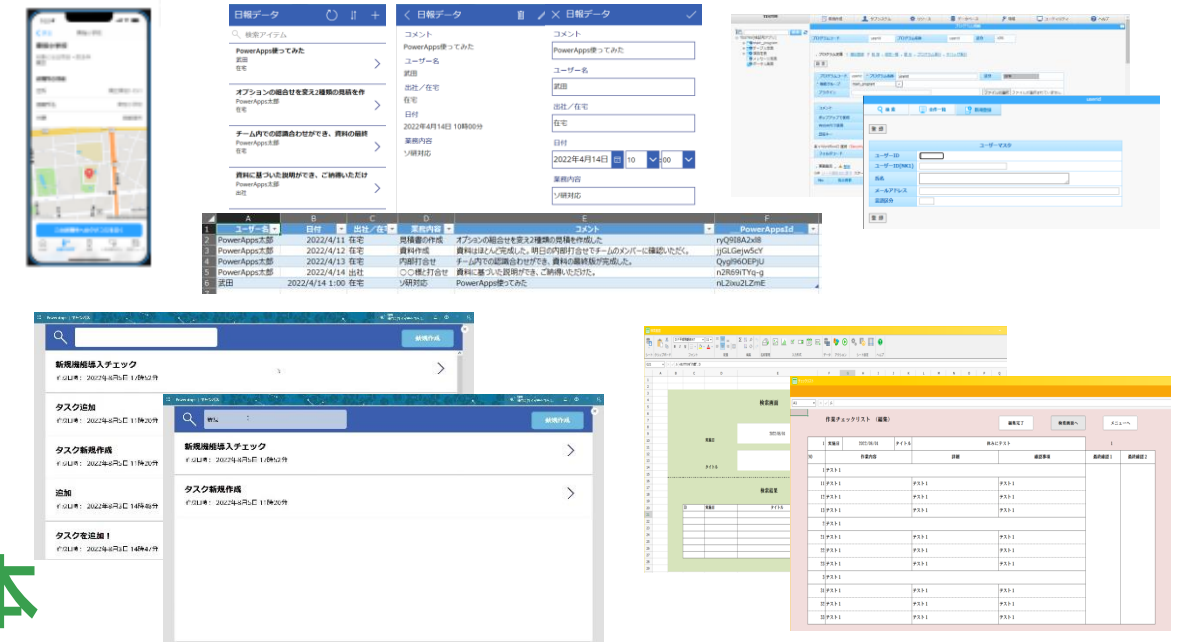
検証したツール

11種



開発したアプリ

11本



ノーコード/ローコード開発スタートガイド

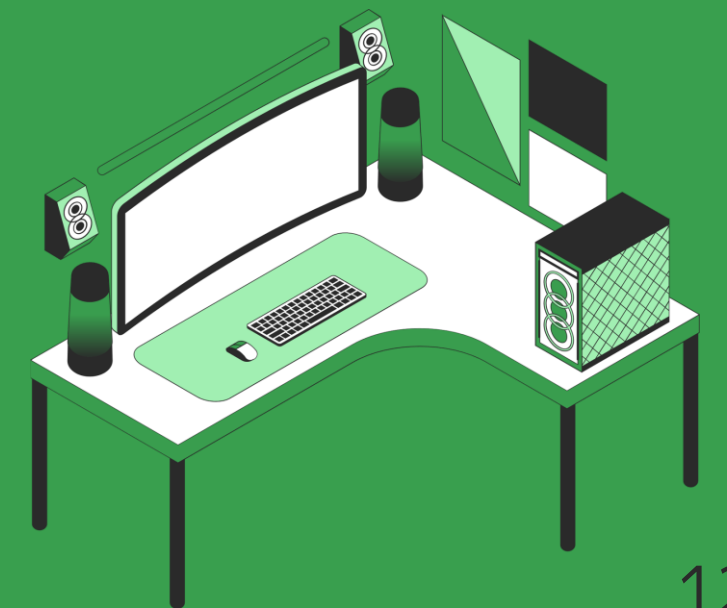
85ページ



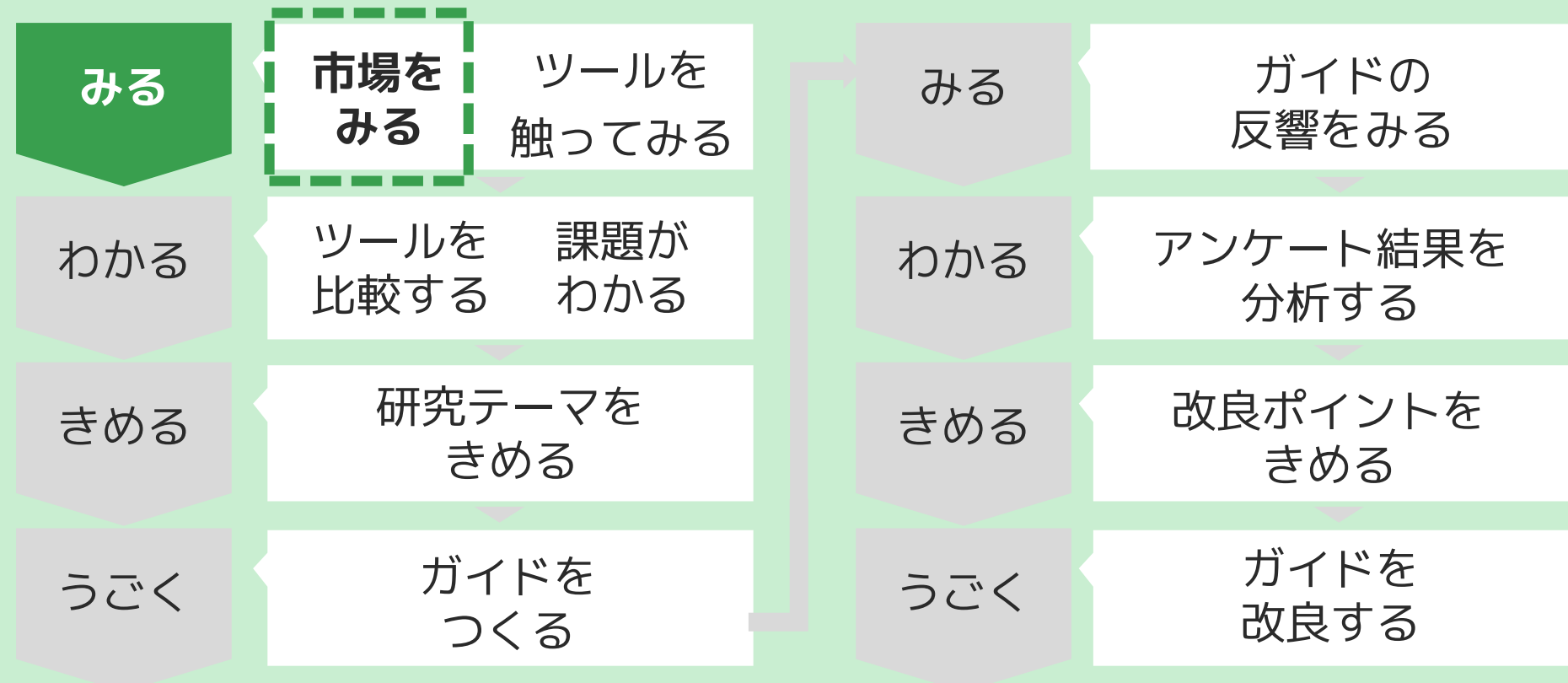
実際にツールに触れてアプリを開発するというリアリティと
多くの人が使えらるものをアウトプットすることにこだわった



II. ノーコード/ローコードツールとは



市場をみる



ノーコード/ローコードの定義

現在、各社が自社のツールを独自にノーコードかローコードに分類していますが…

誰でも使える！

ノーコードツール「XXX」

実際は、作りこみに
プログラミングも必要…

本格的な開発ならコレ！

ローコードツール「XXX」

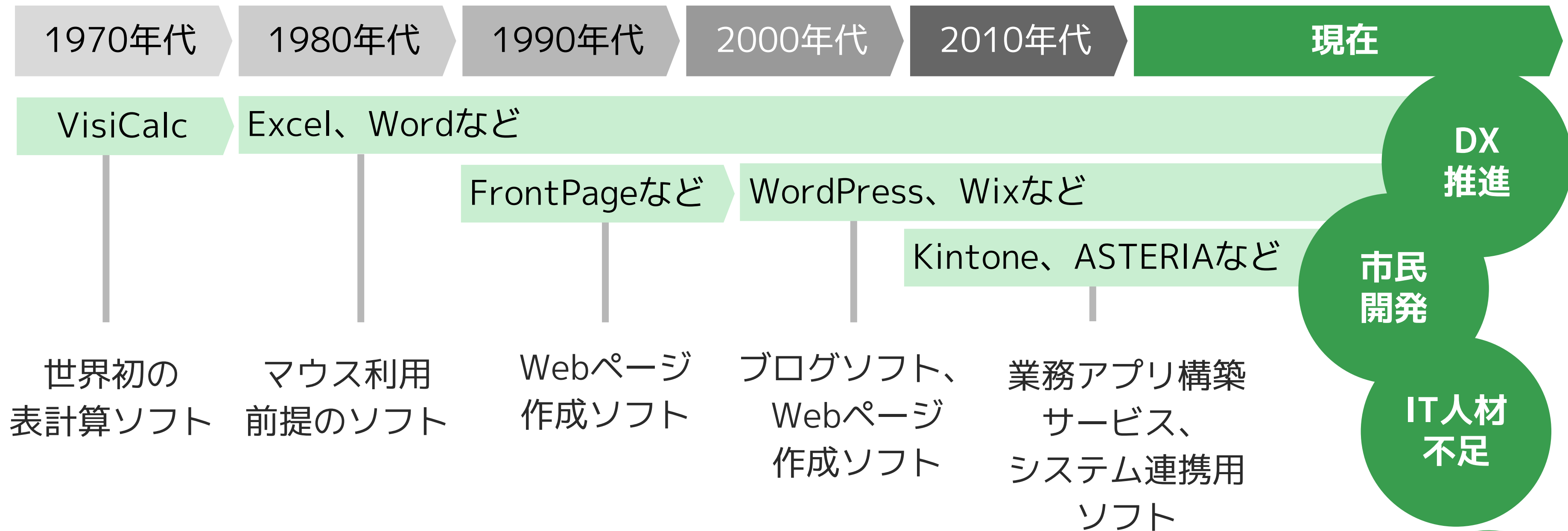
ローコードツールでも
プログラミングしない使い方もある…

そこで私たちは、ツールの分類ではなく開発スタイルとして定義しました。

プログラミングの機能を利用しない開発：ノーコード開発
プログラミングの機能を利用した開発：ローコード開発



歴史と背景



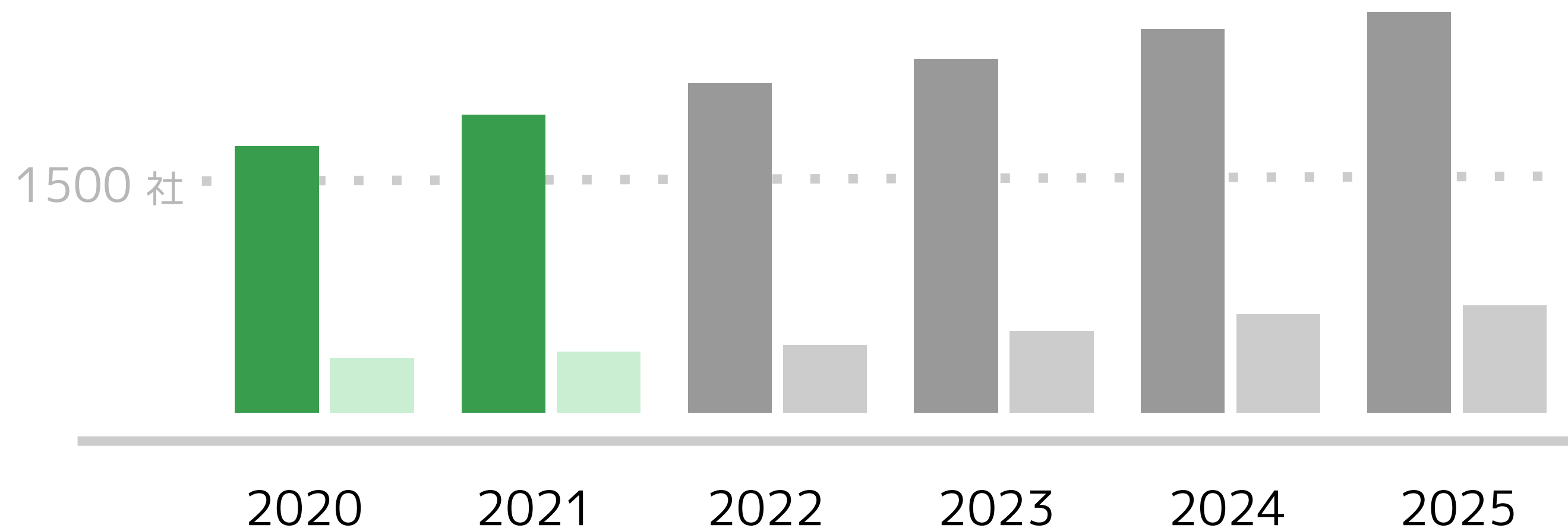
半世紀前からコーディングを削減する努力がされていた
現在、人手不足により非IT担当者による開発のニーズが高まっている



ノーコード/ローコードツールと市場

ローコード開発ツール導入社数の実績と予測は以下の通りです。

- パッケージ型
- SaaS型

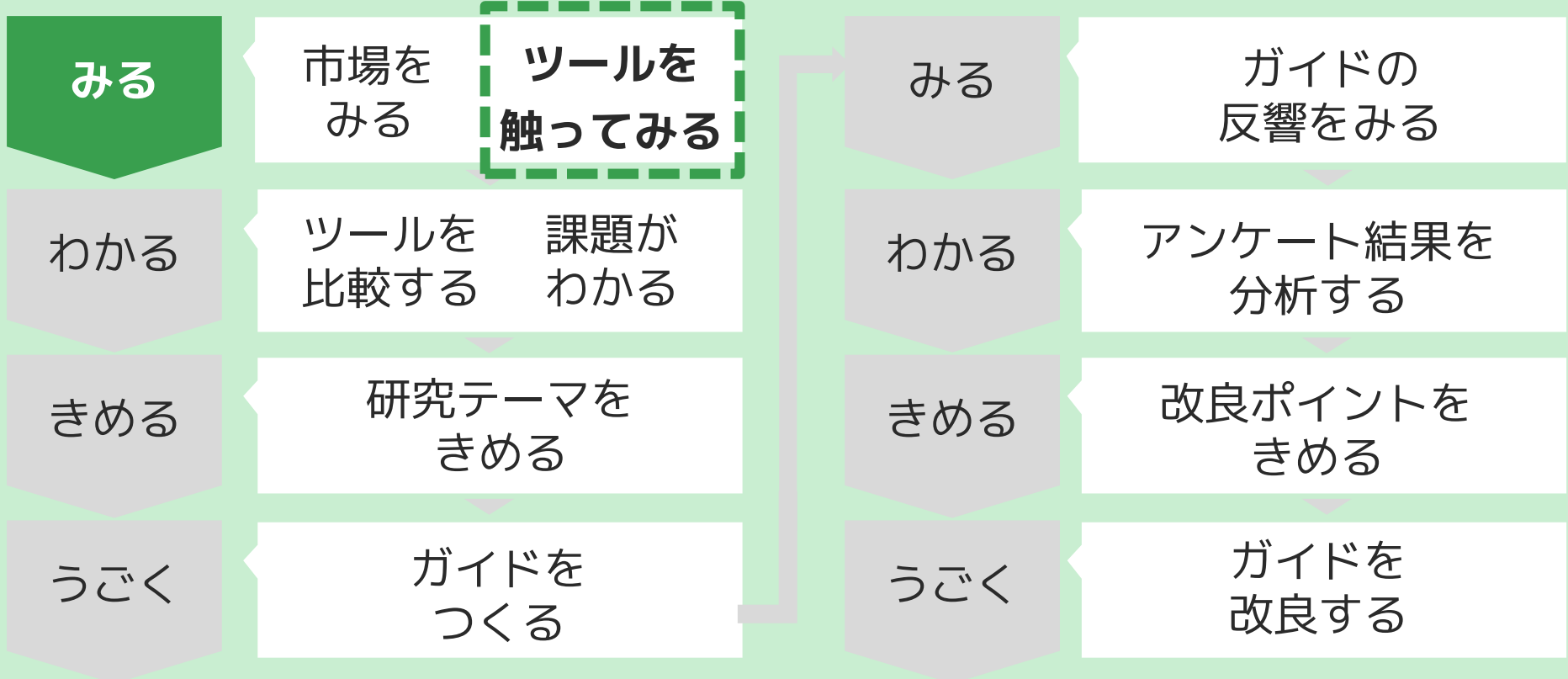


「富士キメラ総研 ソフトウェアビジネス新市場2021年度版ーローコード開発ツール」を参考に分科会にて作成

人手不足・短期開発・工数削減を背景に市場は右肩上がり
昨今話題のキーワードとして認知度も高まっている



ツールを触ってみる



まずはノーコード/ローコードツールを触ってみた

ノーコード/ローコード開発経験のないメンバーもいるため、まずは各自でノーコード/ローコードツールを触ってみました。

ビジネスアプリ

その他業務自動化

This block contains two screenshots of tool pages. The left one is for Microsoft Power Apps, showing its logo and a brief overview of its capabilities for building business applications. The right one is for Framework3, a Japanese no-code tool, featuring its logo and a list of features and use cases.

This block contains two screenshots of tool pages. The left one is for AppSheet, a Google Cloud-based no-code tool for building data-driven applications. The right one is for Workflow11, a Japanese no-code tool for automating business processes and workflows.

8つのツール

This block shows a screenshot of the Asterio warp tool page. It features the logo and a description of the tool, which is designed for creating business applications with a focus on data integration and automation.

This block shows a screenshot of the bubble tool page. It includes the logo and a list of features, highlighting its ability to create web and mobile applications without writing code.

This block shows a screenshot of the Adalo tool page. It features the logo and a description of the tool, which allows users to build mobile and web applications using a drag-and-drop interface.

This block shows a screenshot of the Glide tool page. It includes the logo and a list of features, emphasizing its ease of use for creating mobile applications from spreadsheets and databases.

データ連携

モバイルアプリ

例：Glideによる災害情報アプリの開発

ノーコード 【Glide】を触ってみた

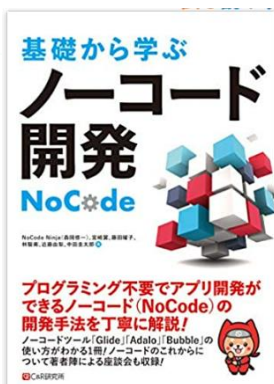
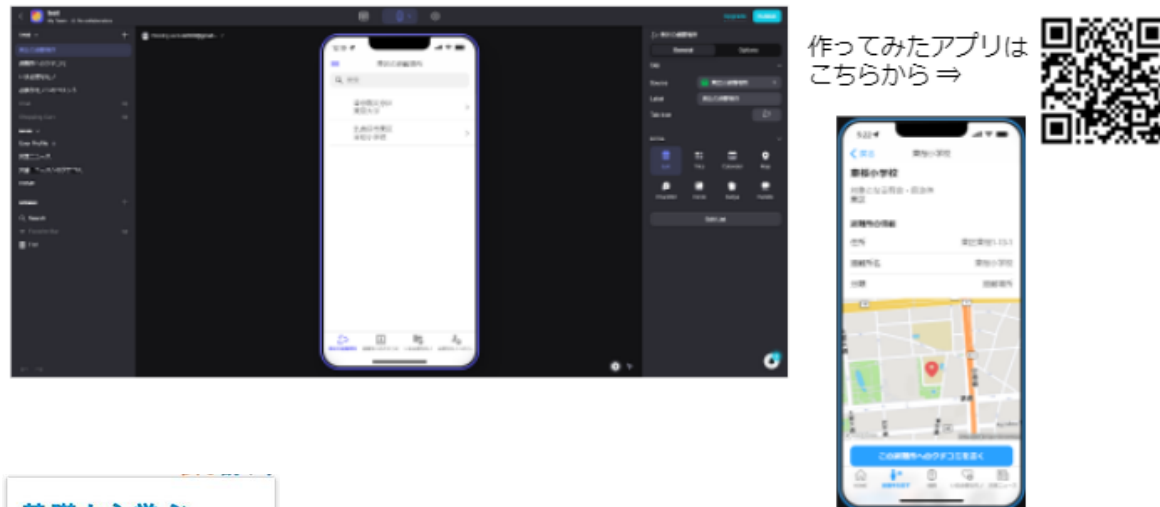
Glide

更新日：2022/4/21
担当：小久保

基礎情報	
開発元 (国)	Glide
製品サイト	https://www.glideapps.com/
提供形態・費用	SaaS ・ オンプレ 無償 ・ 有償
機能・特徴	<ul style="list-style-type: none"> ・ノーコードで、PC・スマホWebサイトを構築できる ・Googleスプレッドシート、Excelなどからアプリを作成できる ・業務向き、というよりはエンドユーザ向き(社内向けのライトなものなら十分な感じもする)

触ってみた評価	
やれること	<ul style="list-style-type: none"> ・データを投稿して収集する ・地図上にピン打つようなマイマップ作成 ・ユーザ登録(ログイン)機能も作成可能
やれないこと	<ul style="list-style-type: none"> ・テキストの装飾が指定できない ・一覧も最低限のバリエーション以外は利用できない ・配置の細かな調整等はできない ・ユーザの種類による情報の出し分けはできない(かもしれない)
使い勝手	<ul style="list-style-type: none"> ・すべて英語のため、翻訳を活用しながらの操作 ・ハンズオンを一度やればコツは比較的つかみやすい
マニュアル、サポート、コミュニティなど	<ul style="list-style-type: none"> ・公式マニュアル、コミュニティあり(残念ながらすべて英語でのやりとりになる) ・インターネット上の情報も多少はあり
想定される活用シーン	<ul style="list-style-type: none"> ・例に作成したような情報を発信したり、収集するコミュニティサイトのスピード構築

画面イメージ/こんなアプリを作ってみた



参考文献『基礎から学ぶノーコード開発』をもとに開発

アプリ動作イメージ



ノーコード/ローコードツールを触ってわかったこと

利点

SaaS型の場合、会員登録するだけで利用開始可能

多くのツールがドラッグ&ドロップで機能を実装可能

パラメータ設定やプログラミング
・API連携で機能追加、拡張可能

掲示板や日記帳、SNSアプリ等はノーコードですぐに構築可能

考慮事項

パッケージ型はインストールに時間がかかり、なかなか開発に辿り着かない

ドラッグ&ドロップは、実装数が多い場合に時間がかかることも

特定機能に特化したツールはカスタマイズが限られる

複雑な業務の実装、デザインの細かい装飾にはカスタマイズが必要

4つの観点

始めやすさ

操作性
学習コスト

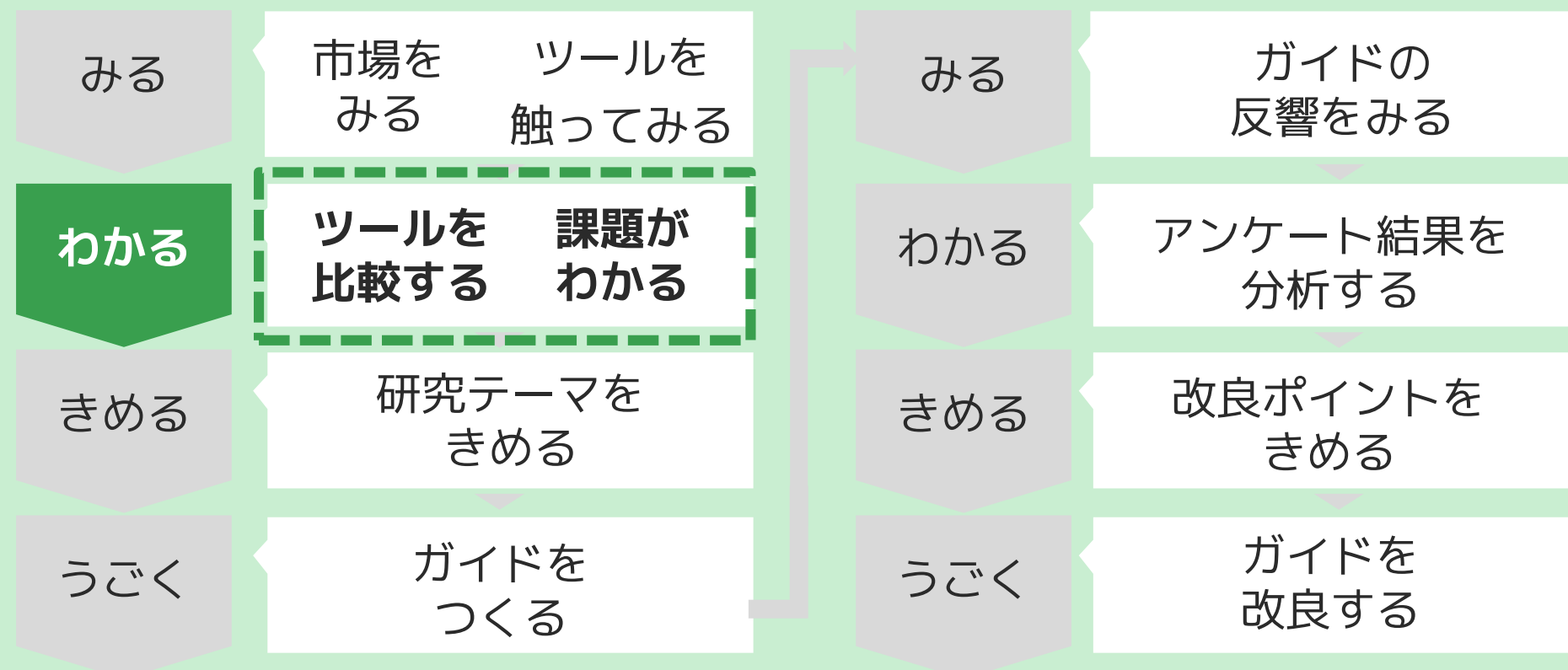
柔軟性

業務適合性

主に4つの観点で違いがあることがわかり、ツールを比較する際の軸に

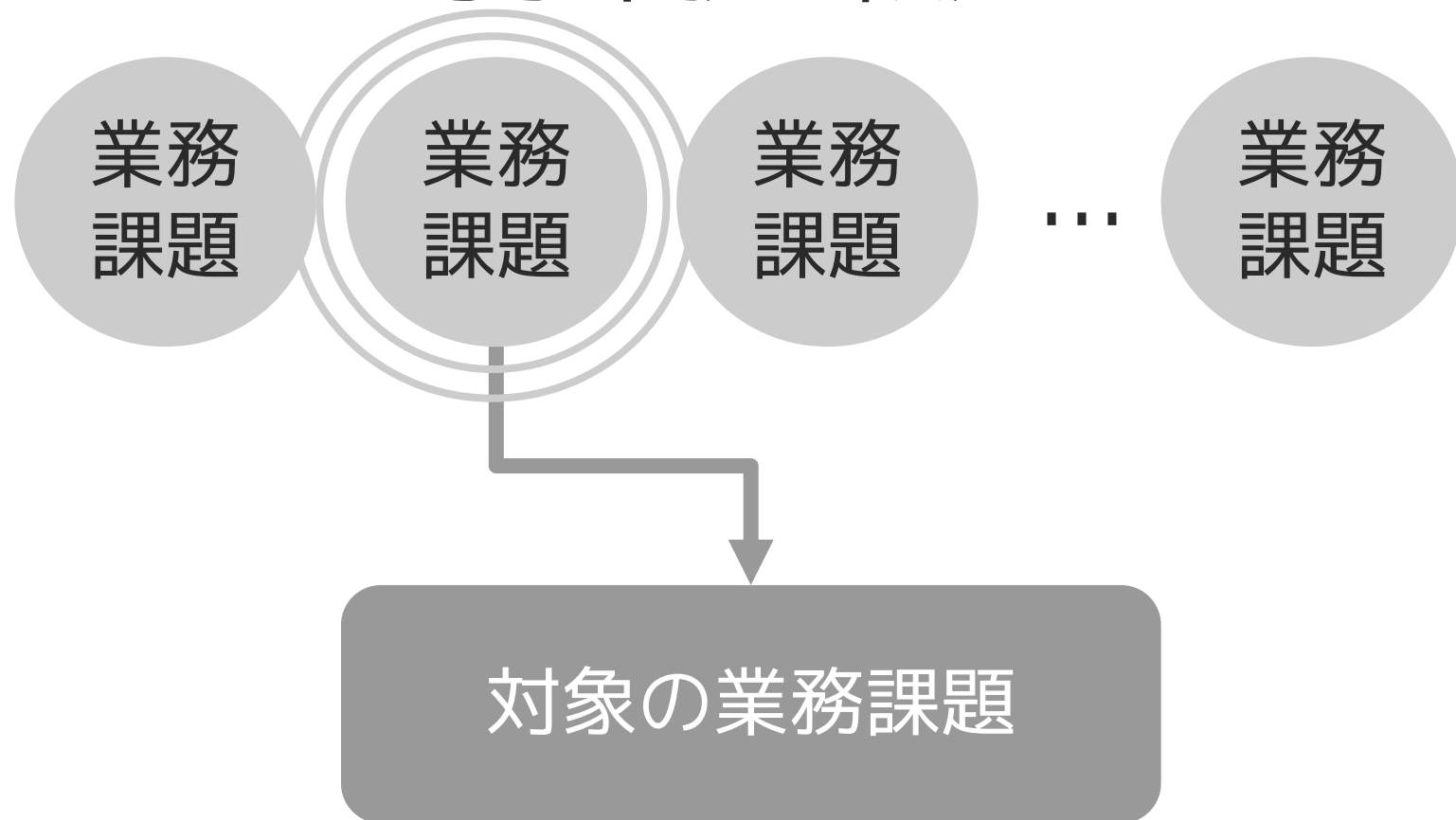


ツールを比較する / 課題がわかる

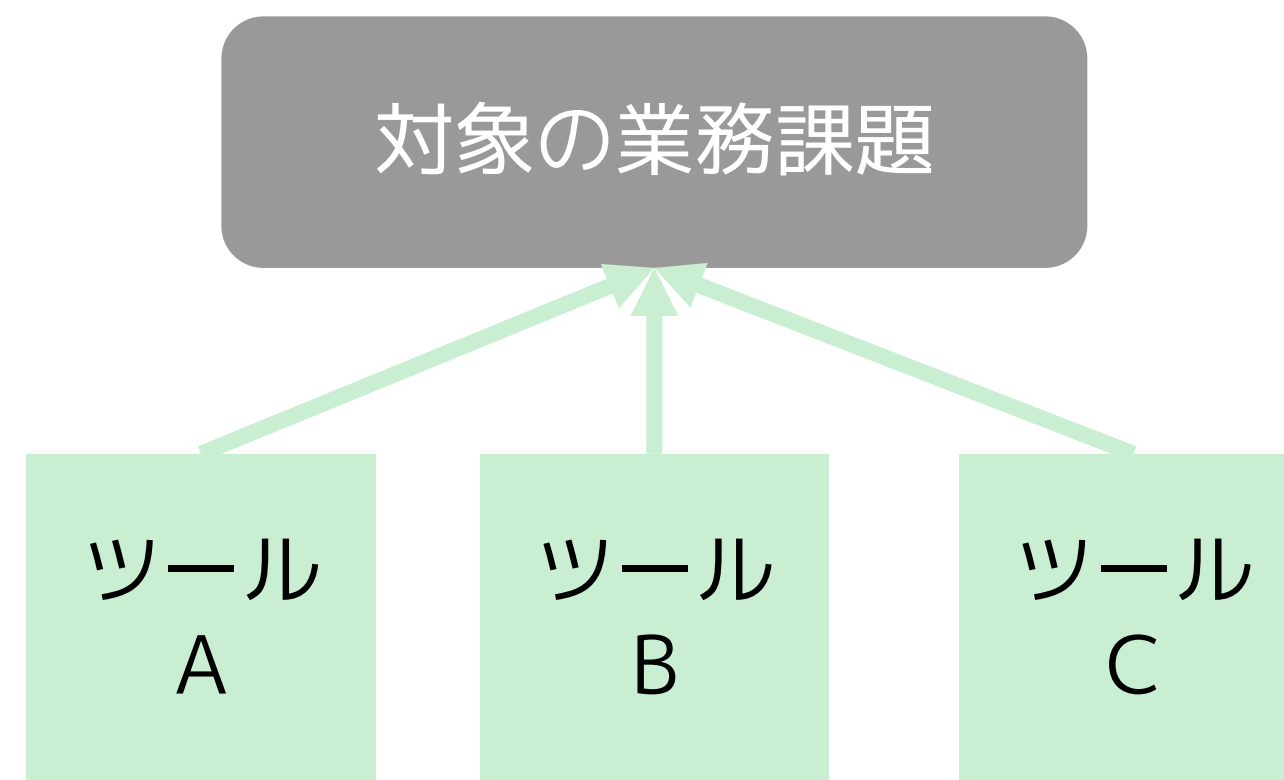


同一テーマにおけるノーコード/ローコードツール比較

自分たちが抱える業務課題から
対象業務を設定



対象とした業務課題について
複数のツールでアプリ開発

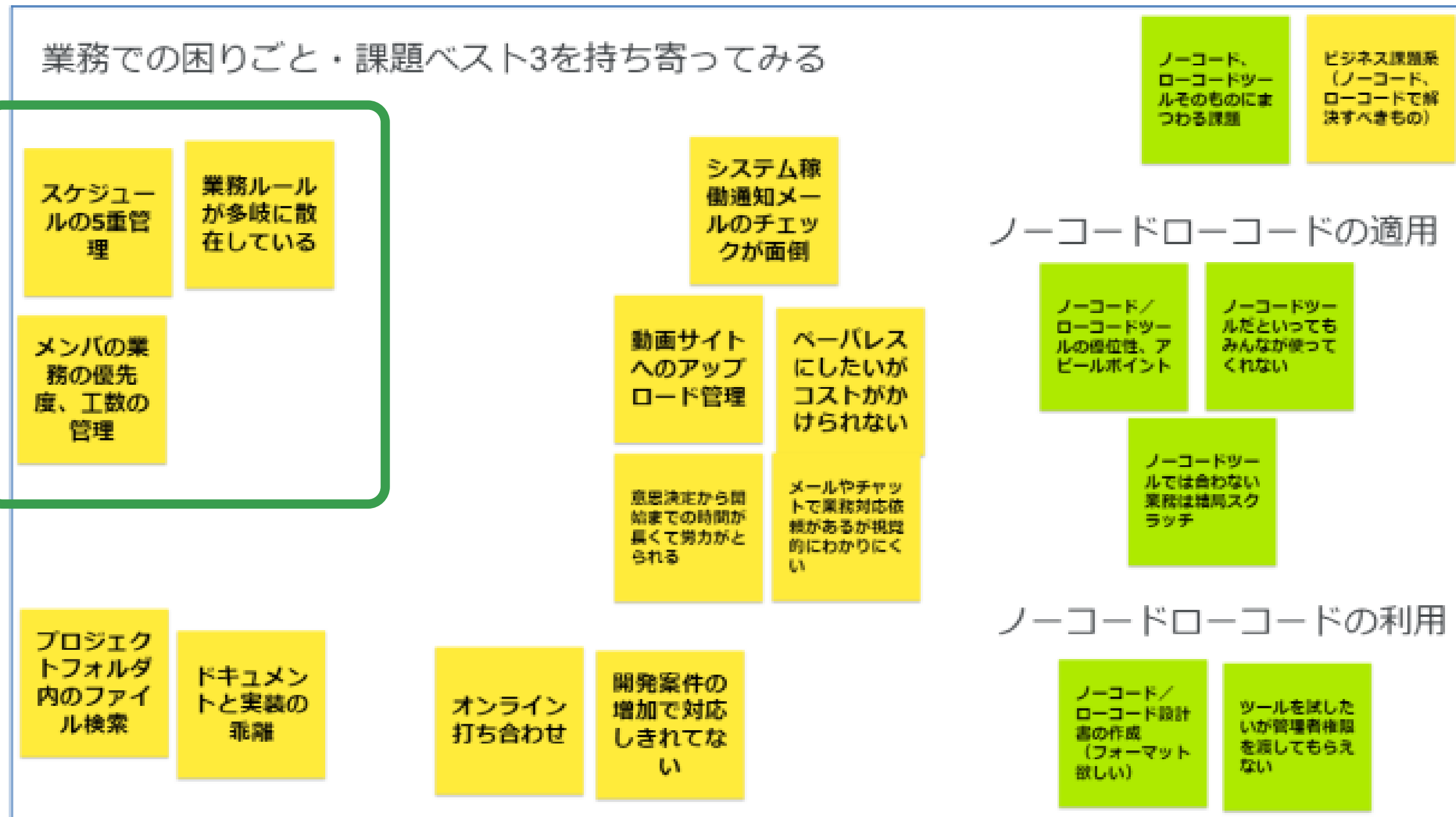


1つのテーマに対し、異なるツールでアプリを開発してみてその結果を比較



チームメンバーが抱える業務課題の洗い出し

タスク管理にまつわる課題

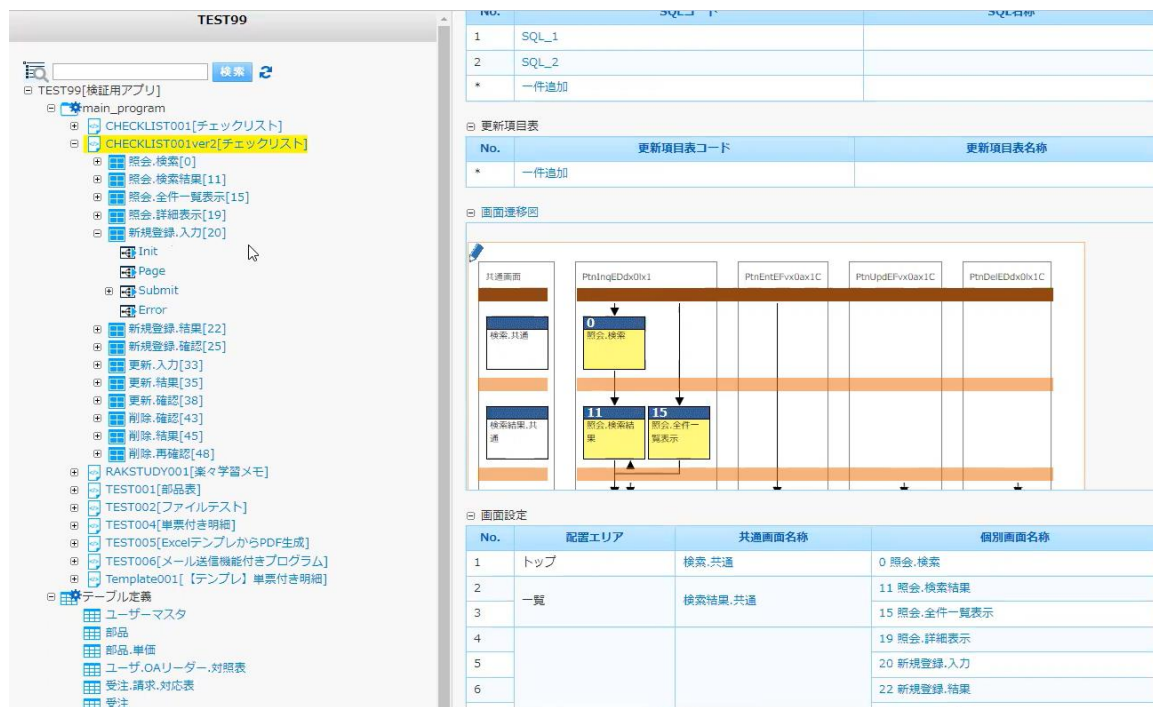


タスク管理のためのチェックリストをアプリ化することを決定



ノーコード/ローコードツールでの開発結果（Aチーム）

楽々Framework3



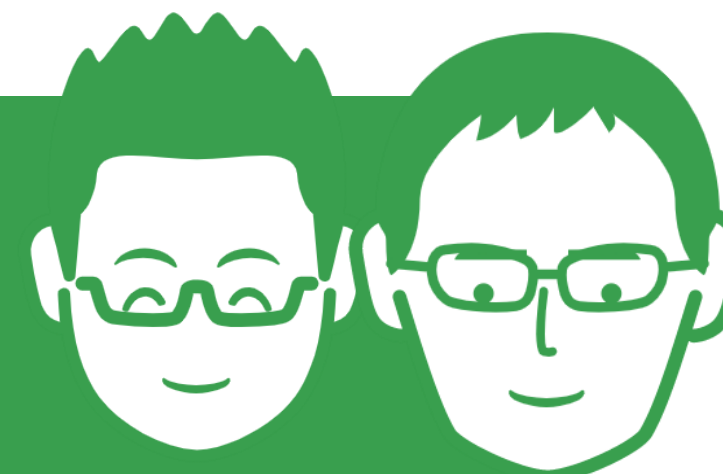
開発したアプリの画面



今回のテーマへの適合度：★☆☆☆☆

約 17 時間で開発

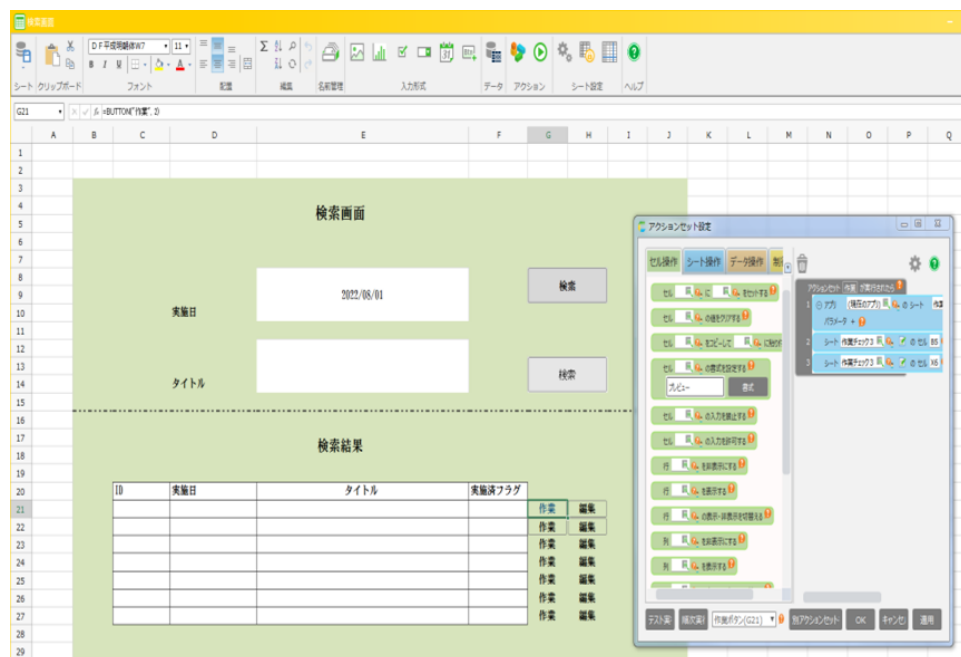
環境構築、カスタマイズに専門的な知識が必要
複数画面に同じ設定を適用できるため、共通画面は作成しやすい
学習、経済コストが高いため、少人数には不向き



ノーコード/ローコードツールでの開発結果（Bチーム）



開発したアプリの画面



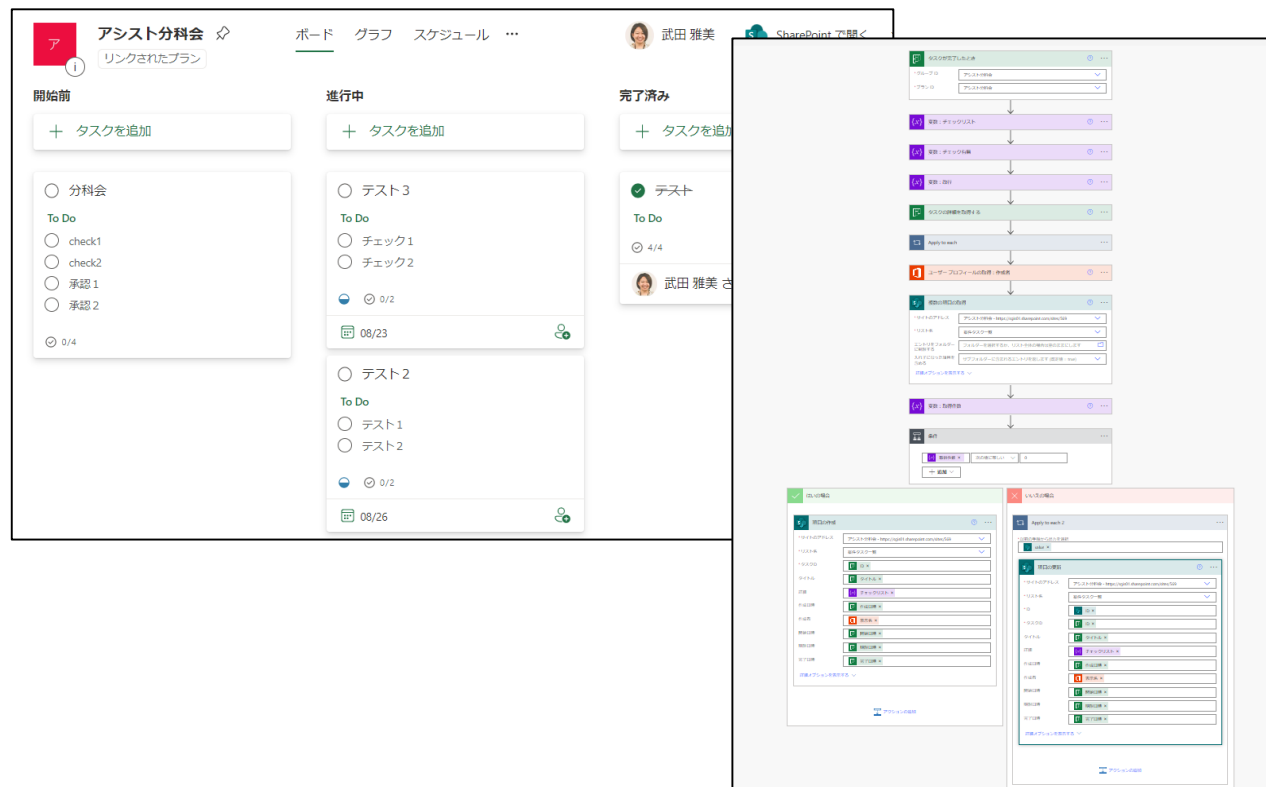
今回のテーマへの適合度：★★★★★

約 10 時間で開発

操作はExcelライクであり、また無料トライアルには操作説明動画が付属しているため、初心者でも始めやすい
複雑な作りこみは難しいが個人や少人数の業務には向いている



ノーコード/ローコードツールでの開発結果（Cチーム）



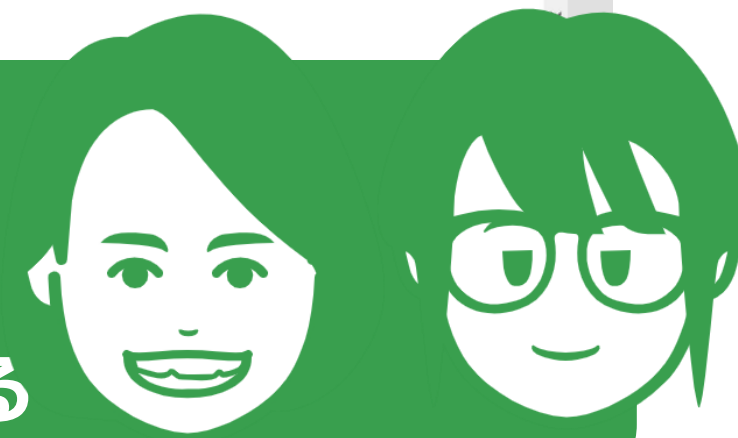
開発したアプリの画面



約 **23** 時間で開発

今回のテーマへの適合度：★★★★☆

サービス連携には一定の知識が必要、機能が多いため
必要な情報を探すスキルも求められる
ブラックボックス化の懸念もあるが少人数での利用に向いている



ノーコード/ローコードツールの比較

少ない工数で開発可能とはいっても、使うためのハードル、業務との相性などはツールによって異なることを再確認しました。

	楽々Framework3	CELF	PowerApps等
始めやすさ	△	○	○
操作性 学習コスト	△	○	△
柔軟性	○	△	○
業務適合性	ビジネスアプリ向け	個人・少人数向け	ビジネスアプリも 個人・少人数も

初心者がこれらを考慮してツールを選定するのは困難…



理解を深めると見えてくる問題点

1

業務課題に**どのツールを
選定すべきか**わからない

2

導入が難しい、実用に
コーディングが必要など
初心者には壁

3

ツールが使えるようになると

野放しの
問題が発生する

「野放し」とは…

- 組織内のツールの乱立
- 開発したアプリの乱立
- 開発後のブラックボックス化

理解を深めると見えてきたのは大きく3点の問題点



解決すべき課題

抽出した
問題点

ツールを選べない

初心者には難しい

野放しになってしまう

解決
すべき
課題

選定の判断材料が必要

開発の知見が必要

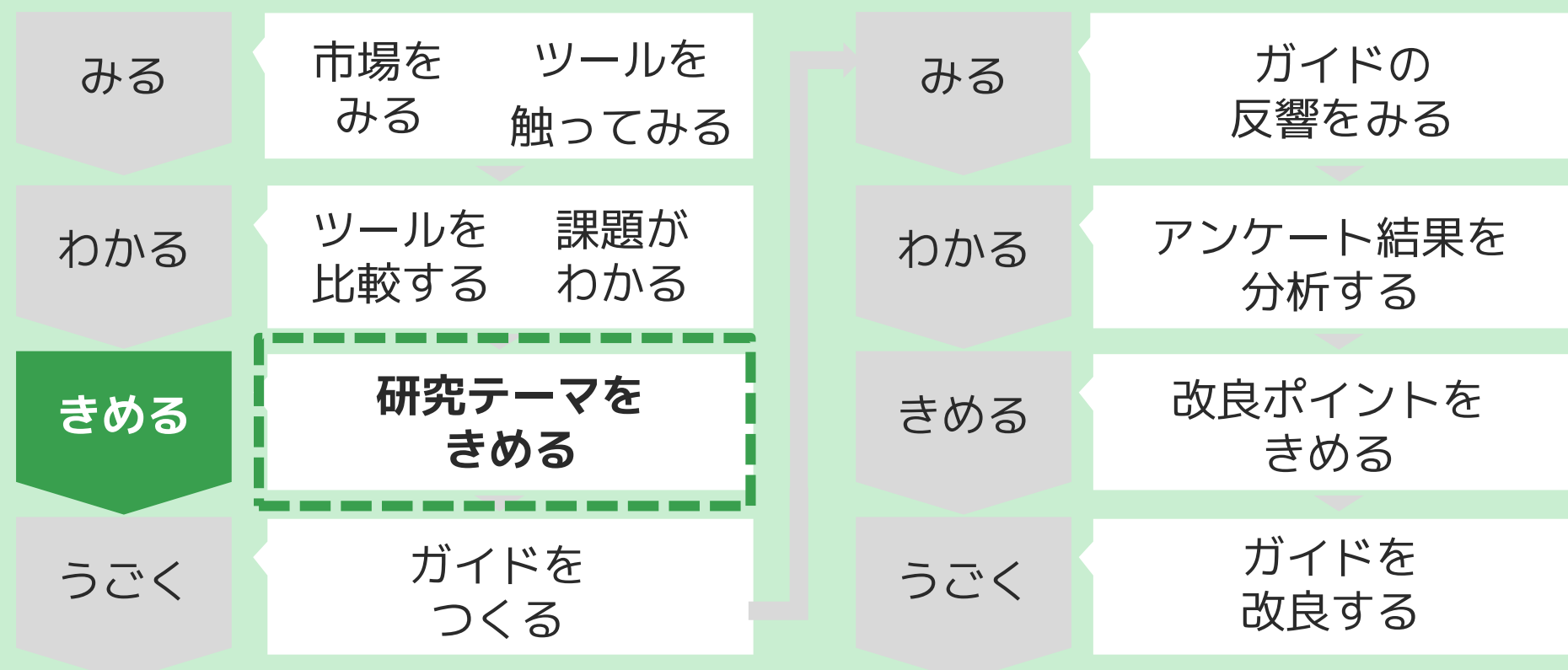
運用ノウハウが必要

これらをふまえ研究テーマの検討へ…

状況に応じて適切なツールを選ぶためのガイドが必要



研究テーマをきめる



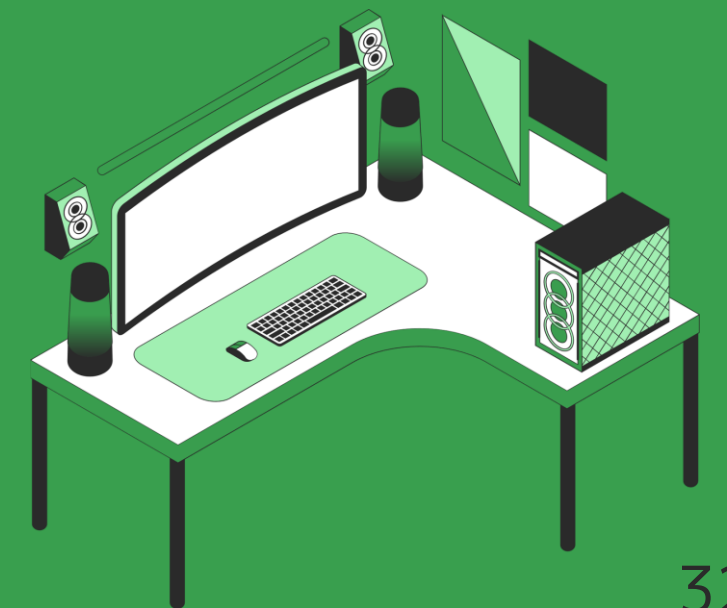
研究テーマの決定

解決すべき課題をふまえ、私たちの研究テーマを以下に決定しました。

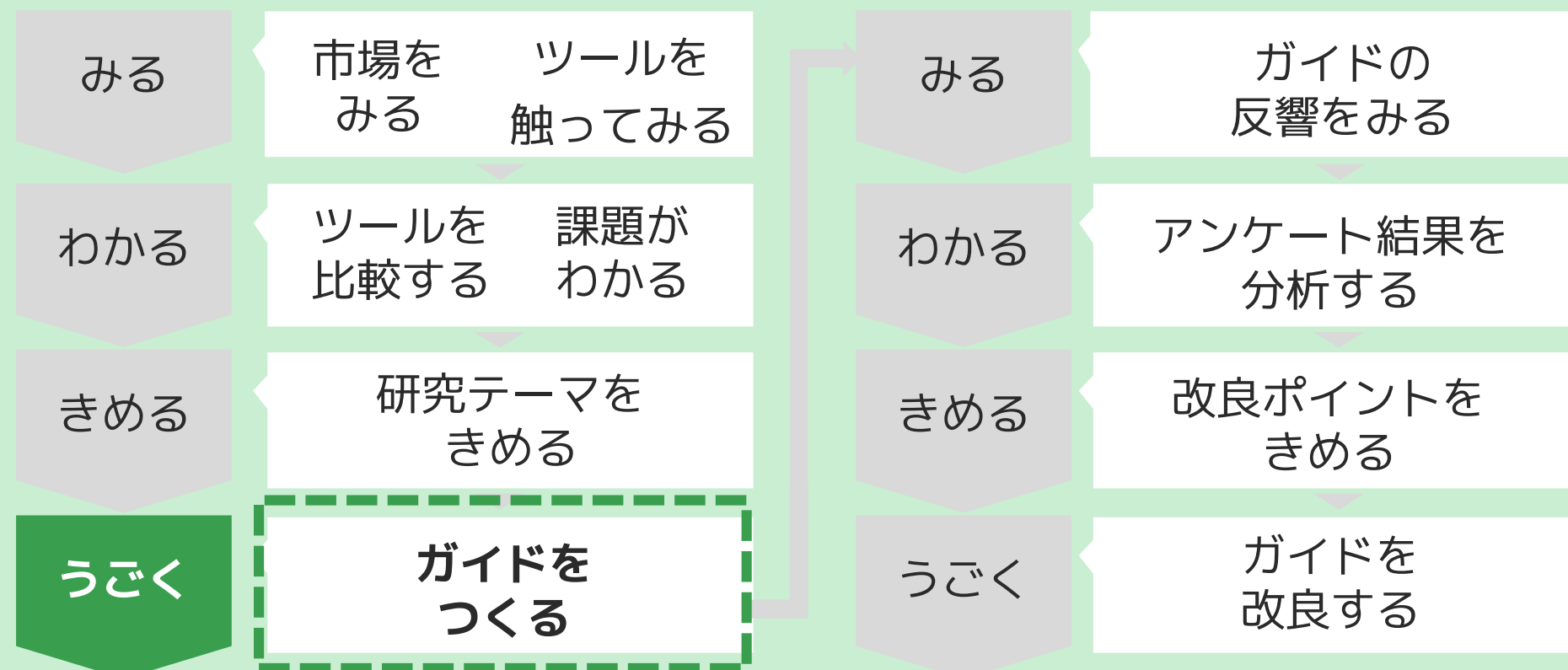
「ノーコード/ローコードツールのスタートガイド策定」
ツール選定だけでなく開発・運用も含めた勘所をスタートガイドとして
まとめてリリースし、その有用性を検証する



III. スタートガイドとその有用性



スタートガイドの概要



まずはスタートガイドをお手元に

スタートガイドー表紙



スタートガイドはこちら。[ここ](#)からダウンロードも可能。



想定するスタートガイドの利用者



スタートガイドで、ツールの選定から開発、運用まで様々なポイントを解説



スタートガイドの記載内容

スタートガイドー「目次」

1. [ガイドラインの概要](#)
 - 1.1. [ノーコード/ローコードの定義](#)
 - 1.2. [歴史と背景](#)
 - 1.3. [ノーコード/ローコードの種類と導入率](#)
 - 1.4. [主な開発手法の特徴](#)
 - 1.5. [ガイドラインの活用目的と活用方法](#)
 - 1.6. [必要なスキル](#)
2. [選定](#)
 - 2.1. [知っておきたいツールの特徴](#)
 - 2.2. [ツール選定の流れ](#)
 - 2.3. [ツールの導入体制](#)
3. [開発](#)
 - 3.1. [事前準備](#)
 - 3.2. [開発](#)
 - 3.3. [テスト](#)
 - 3.4. [リリース](#)
4. [運用/展開](#)
 - 4.1. [管理方法](#)
 - 4.2. [システム改善のポイント](#)

項目のリンクをクリックすれば該当のページへ移動可能



スタートガイドの活用方法

各担当者が直面する悩み・課題にあわせて、該当する章を参照することができます。

スタートガイドー「ガイドラインの活用目的と活用方法」

- 利用を始めたいが何から手をつけたらよいか？ → 『[2.1 知っておきたいツールの特徴](#)』へ
- どんなツールを選定したらよいか？ → 『[2.2 ツール選定の流れ](#)』へ
- 開発に着手する前に必要なことはあるのか？ → 『[3.1 事前準備](#)』へ
- 作りっぱなしを防止したい！ → 『[4.1 管理方法](#)』へ

システムに精通していない担当者でも、
ノーコード/ローコードツールを活用し業務改善ができる



スタートガイドの期待される効果

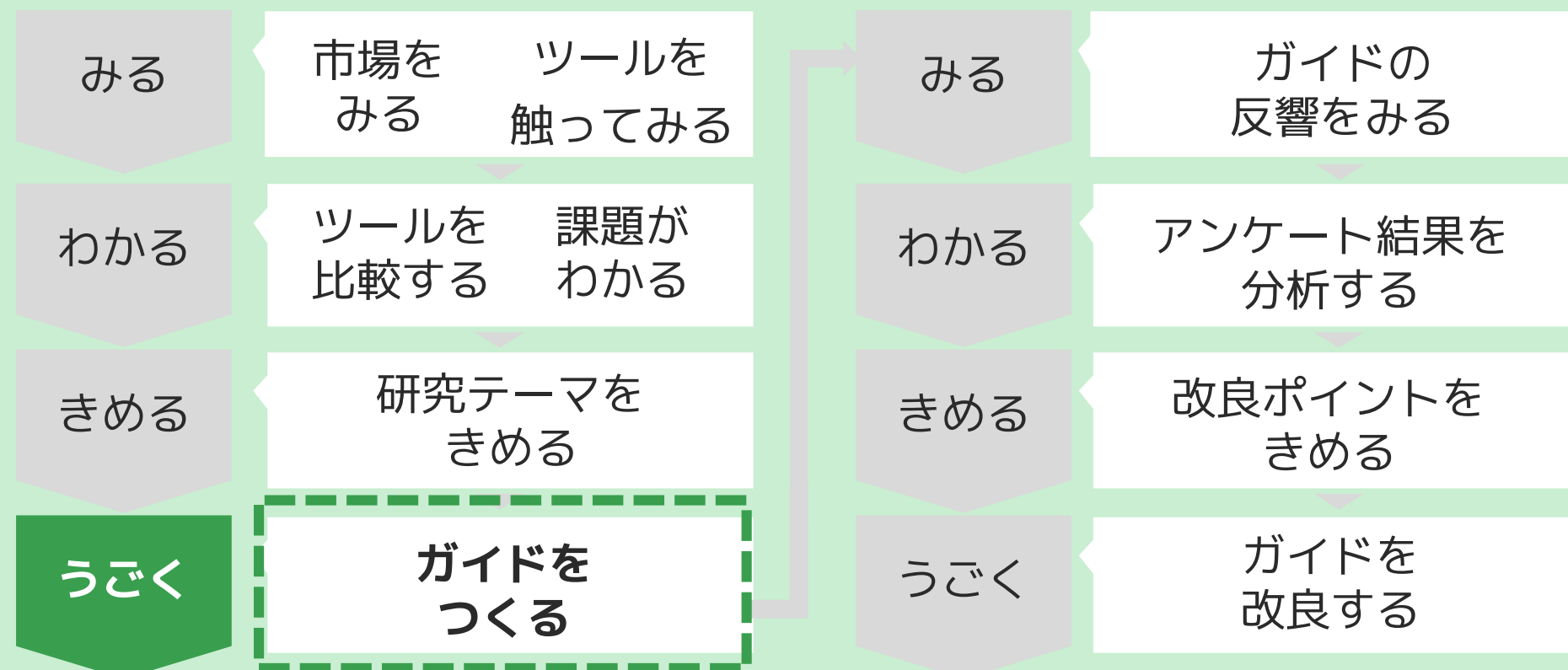
初心者がノーコード/ローコードツールを**選定**できる

初心者がノーコード/ローコードツールで**開発**できる

初心者が**運用**を知ることによって**野放しを防止**できる



ノーコード/ローコード開発の勘所



ノーコード/ローコード開発の勘所 登場人物

ここからはこの2人の掛け合いとともに、スタートガイドに盛り込んだノーコード/ローコード開発の勘所を「①選定」「②開発」「③運用」の3つのシーンでご紹介します。



名前：ユータ (27)

説明：悩める社員
経験浅め



名前：ユージ (35)

説明：スタートガイド所有
ユータの先輩

経験の浅いユータ君でも、ノーコードツールの開発や運用をできるようにする！



ノーコード／ローコードツールを活用してみたいが
何からはじめよう？ツールをどう選ぼう？



ユータ

DXのプロジェクトで「ノーコードツールを活用しろ！」
と言われたけど、どうしたらいいですか？

ユータ君！どこから手をつけていいか分からないんだね。
これをまず読んでみたらどうかな。

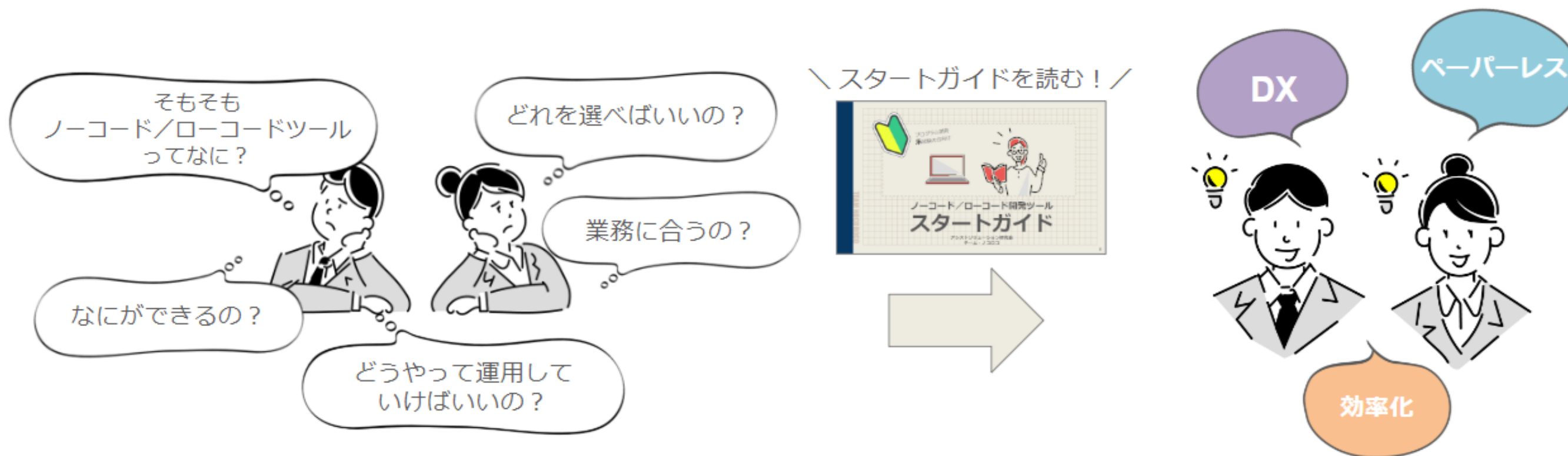


ユージ



はじめに

プログラミングの必要のないノーコードツール、プログラミングの必要が少し出てくるローコードツールはユーザー部門が自分たちでシステム開発を行うことができる夢のようなツールですが、事前に知っておくべきことやおさえておくべきポイントがいくつもあります。



このスタートガイドでは、ツールの選定から開発、運用まで様々なポイントを解説しています。
また、専門的なIT知識を持たない担当者（組織）が、小規模の業務改善用のシステムを開発できるようになるよう

まさに僕のような人のための「スタートガイド」ですね！



ユータ

目的

選定

開発

運用

2.2 ツール選定の流れ

様々なツールが存在する中で、業務課題の解決が可能なツールを選択するのは非常に大変です。
ここでは、ツールを選定するまでの流れについて説明します

1

業務課題の整理

ツールのことは一旦脇に置いて、まずは現状の業務課題を把握することから始めることを推奨します。

2

ツール適用範囲の検討

解決したい課題が何かを把握したところで、課題を解決するために業務のどの部分にツールを適用すればいいのかを検討します。

3

ツールの選定

ツールを利用したい業務を絞り込んだところで、ツールの選定に入ります。

選定フロー

導入事例へのリンクを含む選定フローを用意しました。

まずは対象となる業務課題をしっかりと捉え、
ツールによって解決すべき範囲がどこなのかを検討しよう。

概要

選定

開発

運用



2.2.2 ツール適用範囲の検討



業務課題を解決するために、業務のどの部分にツールを適用するかを検討します。
ただし、ツールにより対応可能なことが異なるため、ガチガチに適用範囲を決めることは推奨しません。
判断基準例を以下に記載します。



向いている業務

- 定型業務である
- 単純作業だが、時間が掛かっている
- 紙やExcelで運用している
- 難易度が低いが、業務改善効果が得やすい
- 複数システムへの重複入力が発生している
- 本社と現場など遠隔地での情報共有に課題がある
- システム間のデータ連携

向いていない業務

- 実現したい内容が複雑
- 求められる機能が多い
- 堅牢なセキュリティが求められる



ツールにより適用範囲が変わる場合があるよ。



スタートガイドに記載してあることのどこまでをやったらよいかかわからず、諦めてしまうのではないか？



ユータ

いざ開発とはいっても、ここでもやることがたくさんあるんですか！？



ユージ

ガイドに記載していることのすべてをやる必要はないね。
全体の流れは捉えつつポイントはおさえよう。

3.2 開発

ノーコード

ローコード

開発では以下の作業が必要となります。

- 1 要件定義**
必要な機能を列挙します。
- 2 設計概要**
各種開発手法別に設計書を用意します。
- 3 命名ルール**
開発の際の様々な命名ルールをあらかじめ作っておきましょう。
- 4 データベースの設定**
データを保存しておく場所の設定をします。
- 5 テスト**
作成した機能が正常に動作するか確認します。

あ、いきなりつくるんじゃないのか・・・
まずはこの全体の流れをイメージしておく必要がありますね。



ユータ

3.3.2 テストの実行

ノーコードローコード

正常系・異常系

	観点	サンプル条件	サンプル結果
正常系	想定された値が入ってきた時、想定した結果になること	項目「メールアドレス」に「hoge@example.com」を入力する	正常に登録されること
	想定された操作がされた時、想定した処理が実行されること	存在するアカウント情報「ID」と「パスワード」を入力して、ログインボタンを押下する	正常にログインできること
異常系	想定されていない値が入ってきた時、想定した結果になること	項目「メールアドレス」に「山田太郎」を入力する	登録されず、エラーメッセージが表示されること
	想定されていない操作がされた時、想定した処理が実行されること	存在しないアカウント情報「ID」と「パスワード」を入力して、ログインボタンを押下する	ログインはできず、エラーメッセージが表示されること

目的選定開発運用

ユータ

正常なデータ以外もテストしないといけないってことか・・・
テストの項目で漏れてしまうところでした。

システム開発 = ゴールとなり、その後の対応が疎かに . . .



ユータ

ようやく開発してリリースしました！
これでプロジェクトは終了ですね。ありがとうございました。



ユージ

いやいや、ユータ君。これからも運用保守はしないといけないよ。
スタートガイドにも書いてあるでしょ？

4.1 管理方法

この章では、ノーコード/ローコード開発を安全に進めるために、また、ガバナンスを機能させながら内製の効果をより得ることができるように、管理が必要となる事項について記載していきます。

- 1 開発時の推奨事項
- 2 保守体制
- 3 障害発生時の対応
- 4 ライセンス管理
- 5 セキュリティ
- 6 バージョンアップ
- 7 バックアップ



ユータ

障害発生時の対応やライセンス管理、セキュリティ！？
確かにトラブルが起こってからでは遅いですもんね。

目的

選定

開発

運用

4.1.3 障害発生時の対応(1)

ノーコード

ローコード

障害とは

ツール自体やツールを導入している環境、通信環境などに問題が生じ、正常に動作しなくなる状態のことです。また、その原因となった問題や不具合のことを指します。

【障害発生時に起こることの例】

- データの消失／破壊
- 処理能力の低下
- 外部との通信ができなくなる
- 一部の機能の損失
- システムの完全停止

【障害発生原因の例】

- 操作ミス
- ツールの設定ミス
- プログラミングの誤り(バグ)
- 想定を超える負荷がかかった
- 機器の故障／破損
- 外部からの攻撃
- コンピューターウイルス

SaaS型では、ツールの提供元で発生した障害が原因の場合があります。自身では対処のしようがなく、情報を待つしかないこともあります。



70

4.1.3 障害発生時の対応(2)

ノーコード

ローコード

障害が発生した時

以下のことを並行して速やかに進める必要があります。

- 1 関係者への周知／連絡**
早急に影響を受ける部門や利用者に連絡をしましょう。
- 2 代替手段への切り替え**
業務を止めないことが最優先となります。
運用でカバーできる場合(※)は、プログラムや設定の修正は行わず、一旦運用方法によるカバーで対応します。
※特定の操作でエラーが発生し、その操作が回避できる場合など
- 3 原因究明／復旧作業**
業務への影響度に応じて対策を検討し、暫定的または恒久的な復旧を行います。
すぐに原因が分かり、修正に時間が掛からない場合は、プログラムや設定の修正で対応します。
すぐに復旧することが難しい場合は、暫定的な復旧を行います。暫定的な復旧を行った場合は、プログラム修正などを行った後、利用時間外や利用者の少ない時間帯に改めてリリースを行い、恒久的な復旧を行います。

71

障害発生時の一般的な対応フローはおさえておこう。





4.2.3 利用促進

ノーコード

ローコード

ツールの利活用を促進するため、社内でコミュニティや意見交換の場を設けるといいでしょう。業務改善を実現した事例の共有／発信を積極的に行いましょう。



他部門への展開時に気を付けること

契約の見直し

利用者数：追加利用者分のライセンスが足りない場合
追加で契約する必要があります。

性能契約：利用者増加に伴い負荷が増加する場合
対応できる性能に変更する必要があります。

問合せ先

サポートへの問合せ回数に制限がある場合があります。
一部の利用者が使い切ることがないよう監視／制限をする必要があります。
サポートへの問合せ窓口が一本となるよう体制を整えましょう。

利用を許可する範囲の検討

すべてのメンバーがすべての機能を利用できるようにすることは、セキュリティリスクにつながります。
システムの利用範囲に対して、制限をかける必要があります。

アクセス権限

部外秘の情報など、他部門に見せられないデータがある場合は、アクセス権を設定しましょう。
組織／ユーザー単位で、閲覧／編集可能な範囲を設定する必要があります。

自分だけでなく他のメンバーや他部門にも伝えていこうね。

目的

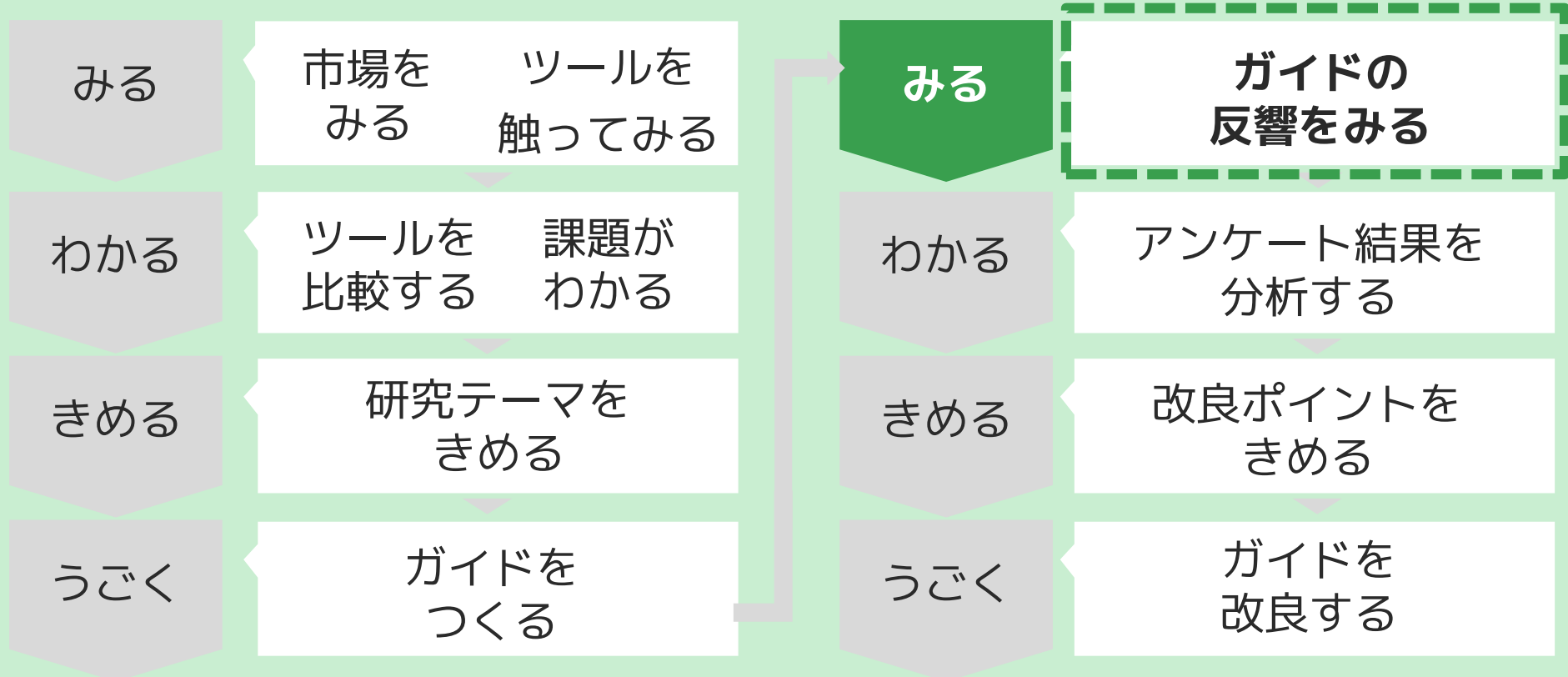
選定

開発

運用



ガイドの反響をみる



有用性検証の概要

目的

1. 作成したスタートガイドの有用性を確認
2. フィードバックからスタートガイドをブラッシュアップ

概要

主な確認事項

実施概要

- 対象者にスタートガイドを読んでもらいアンケート形式で評価

回答者の属性

- 業種
- 経験
- ノーコード／ローコードの認知度

対象

- システム開発未経験者中心
- メンバー各自の社内や、取引先など

内容

- スタートガイドの内容の理解度
- 活用への期待度の評価

目標有効件数

- 30件

アンケート回答者

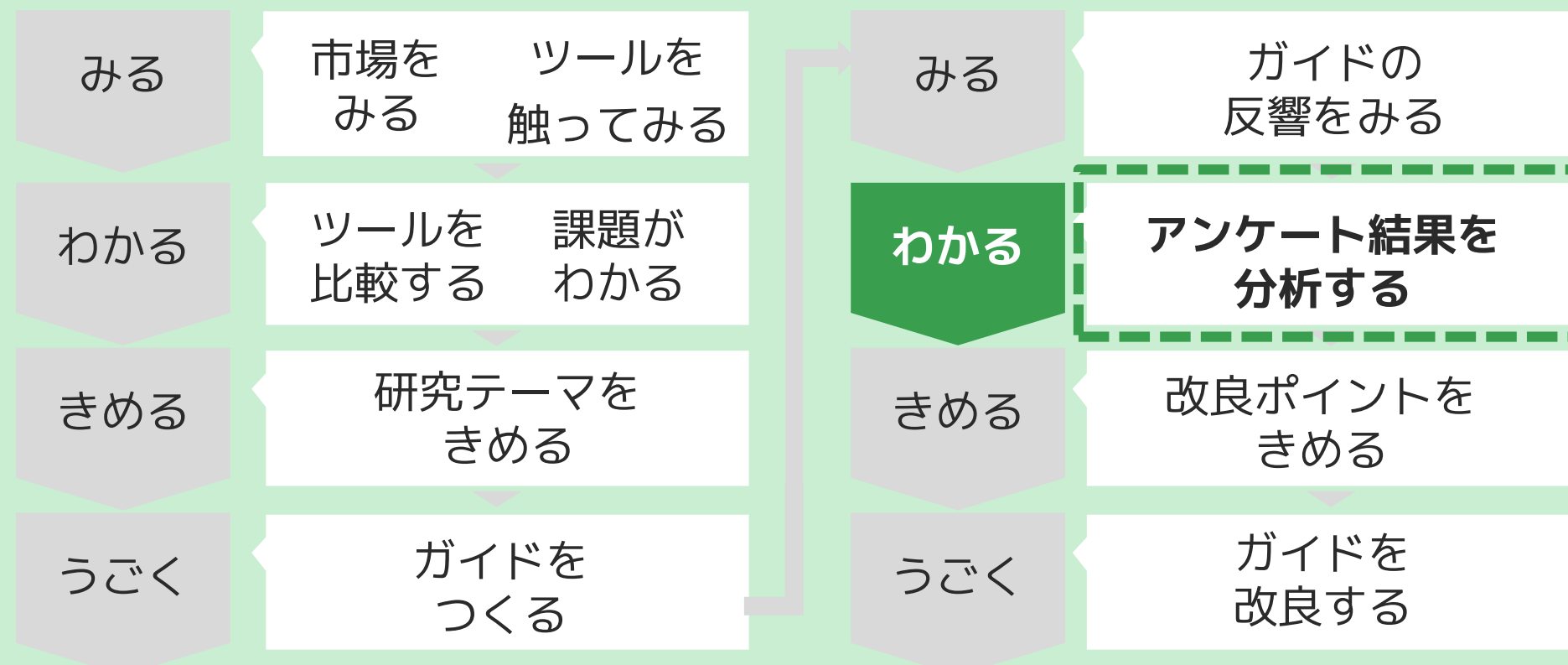
＼感謝／

約80ページのガイド確認とアンケートへのご協力ありがとうございました！

45名の方へ依頼し 35名もの方から回答いただきました

社名	対象者	依頼人数
リコージャパン株式会社	営業部門、IT部門 など	5名
トランコムITS株式会社	総務部門、IT部門 など	5名
鈴与システムテクノロジー株式会社	IT部門だが開発経験無し など	6名
愛知県経済農業協同組合連合会	営業部門 など	10名
株式会社メイテツコム	IT部門、常駐先（某放送局） など	8名
株式会社アシスト	広報部門、新事業部門、営業支援部門、製品技術 など	5名
アシスト取引先	某自動車メーカー、製造業、物流業	6名

アンケート結果を分析する

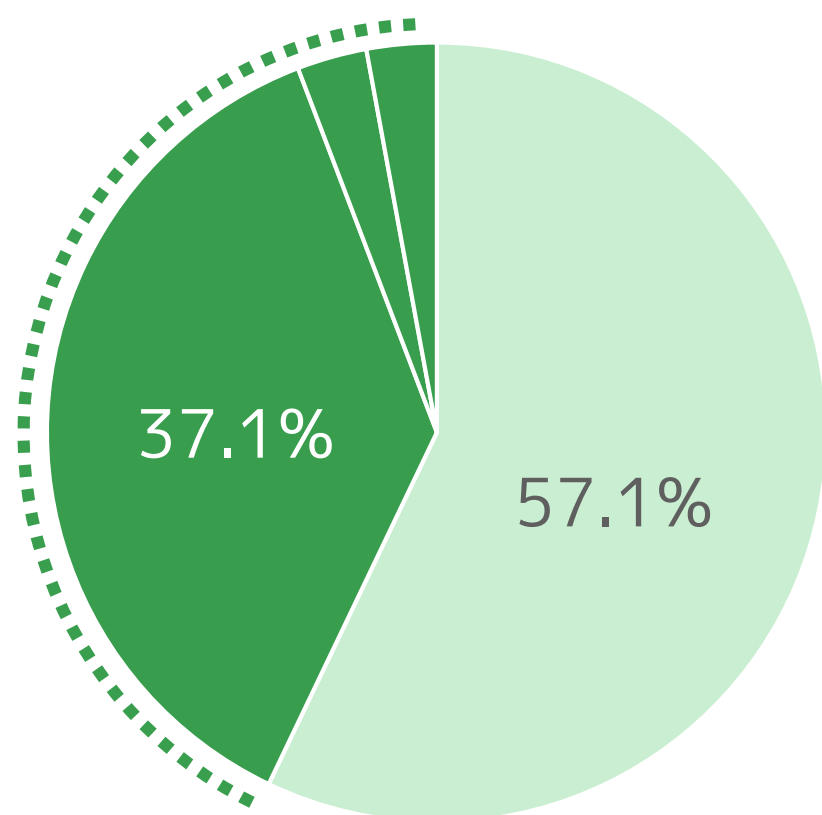


アンケート結果（属性）①

(1) 業種はどちらですか（35件の回答）

- Sierやシステム開発会社、システム子会社などIT系
- 製造業や流通業、金融業などの非IT系
- 放送業
- メディア

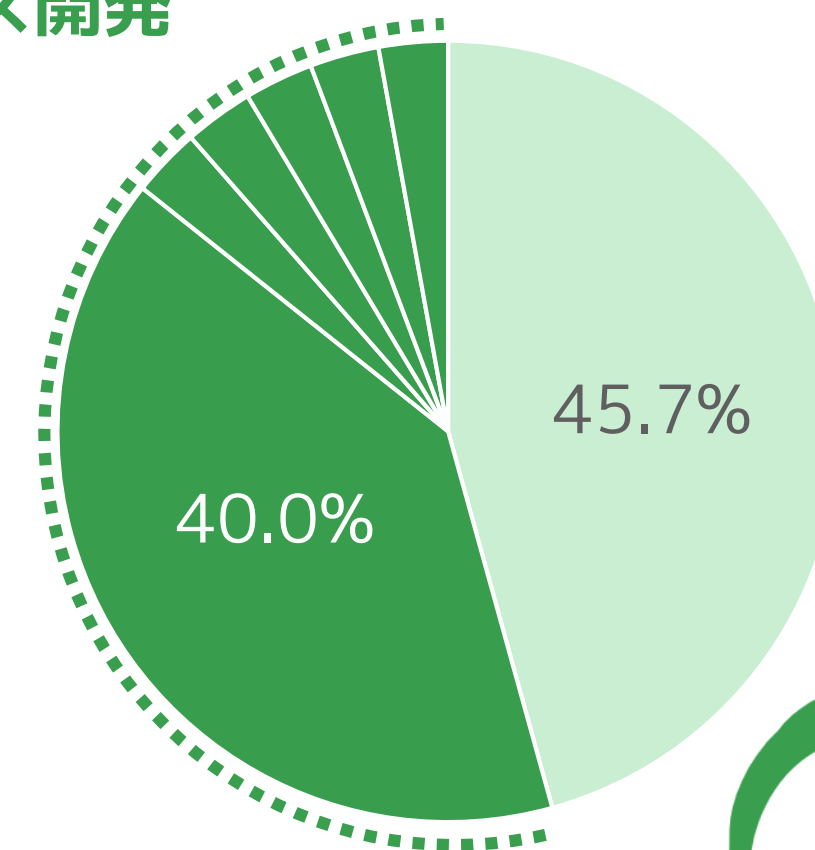
42.9%



(2) 所属部門はどちらですか（35件の回答）

- 情報システム部などIT部門
- 営業部門や人事部門など非IT系部門
- デジタルメディア運用とソリューション開発
- イベント事業
- 新商材・サービス開発
- 生産技術部門
- 製品開発部門

54.3%



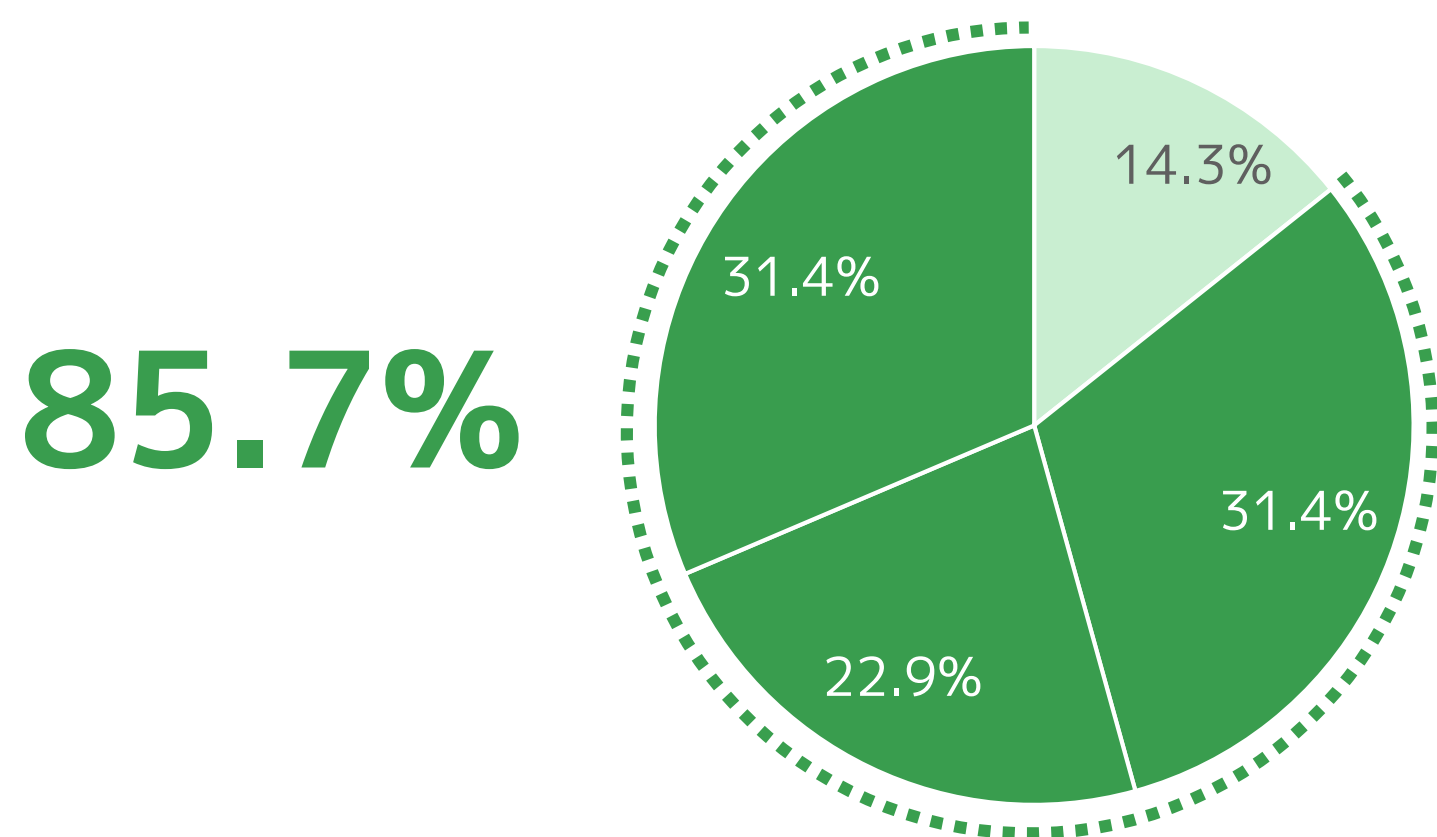
回答者のうち半数近くが、システム開発と関係性が低い



アンケート結果（属性）①

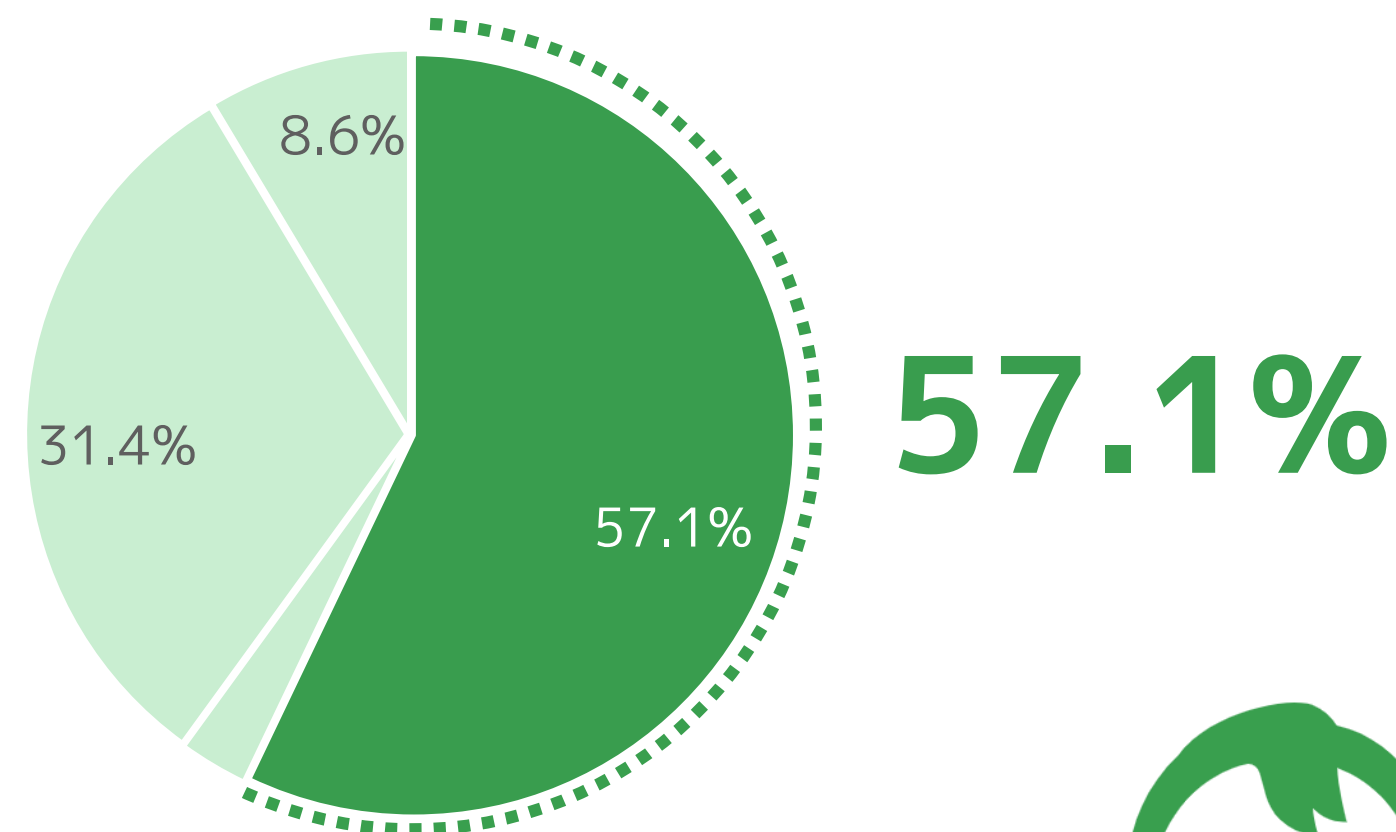
(3) ノーコード/ローコードツールを知っていますか（35件の回答）

- まったく知らない
- どういったものかは知っている
- それらツールを見たことがある。触ったことがある
- それらツールを日常的に見ている。触っている



(4) ノーコード/ローコードツールでの開発をしたことがありますか（35件の回答）

- まったく開発をしたことがない
- 開発をしようとしたが挫折した
- 数回、開発をしたことがある
- 日常的に開発をしている

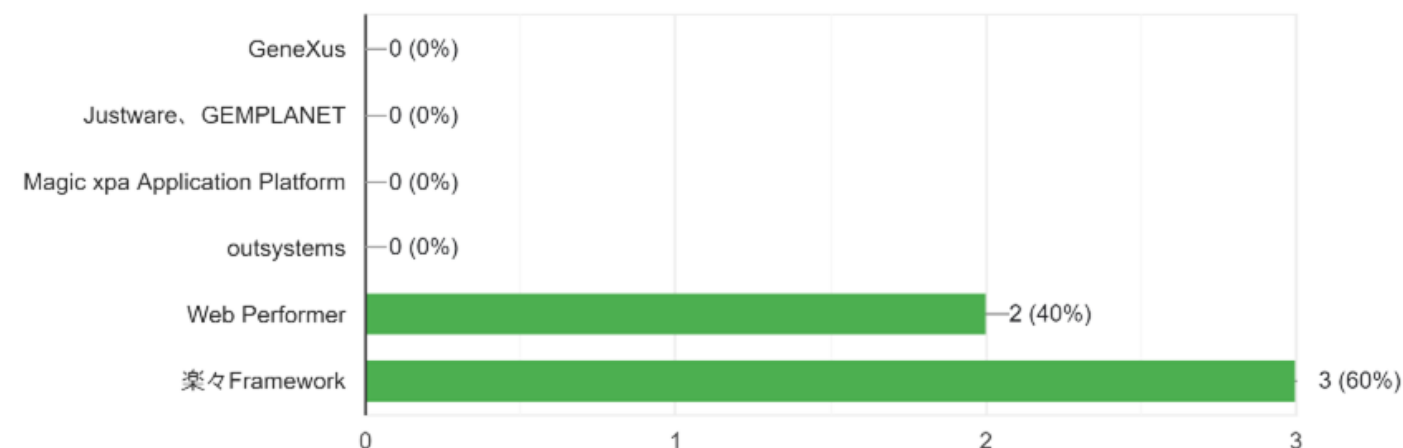


ノーコード/ローコードツールの認知度は8割以上、しかし開発未経験が約6割



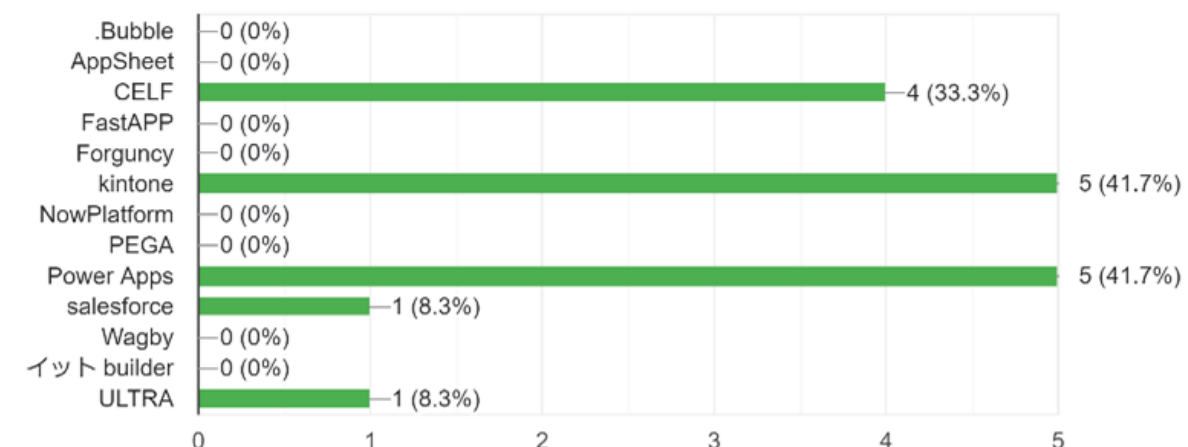
アンケート結果（属性）②

(1) 使用経験のあるビジネスアプリ開発ツール
（市場シェア上位）
（複数選択可）



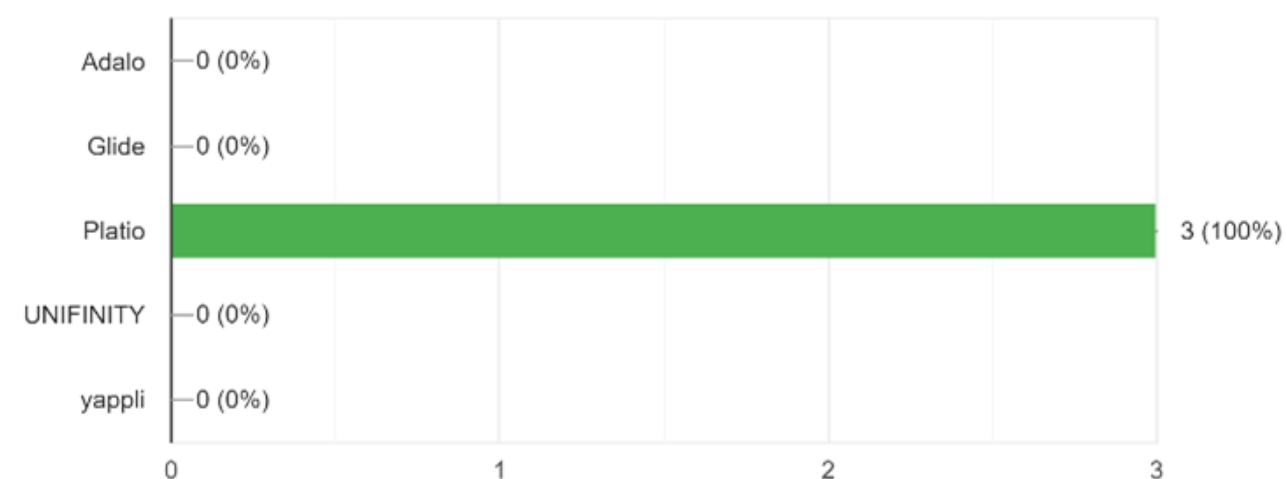
5人

(2) 使用経験のあるビジネスアプリ開発ツール
（その他）
（複数選択可）



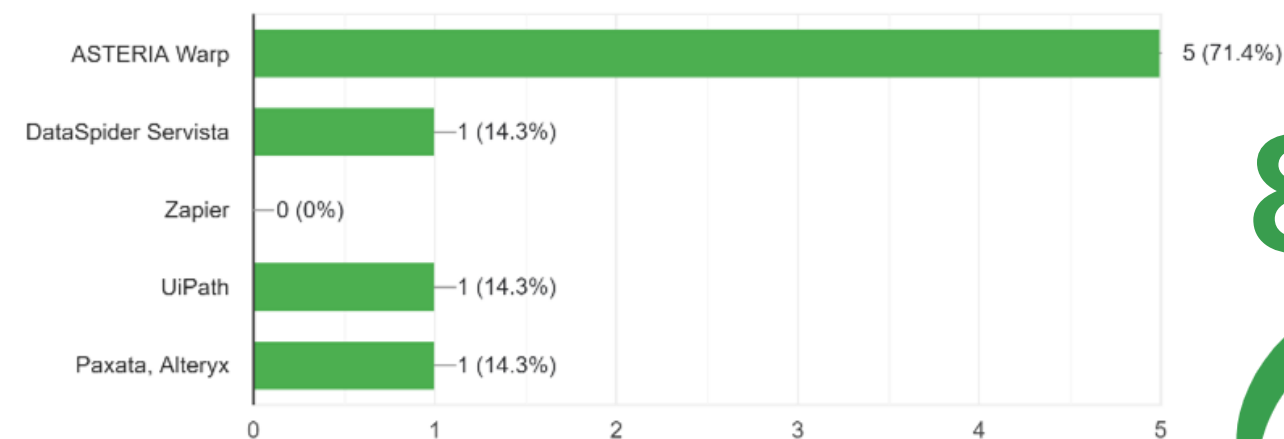
16人

(3) 使用経験のあるモバイルアプリ開発ツール
（複数選択可）



3人

(4) 使用経験のある業務自動化ツール
（複数選択可）



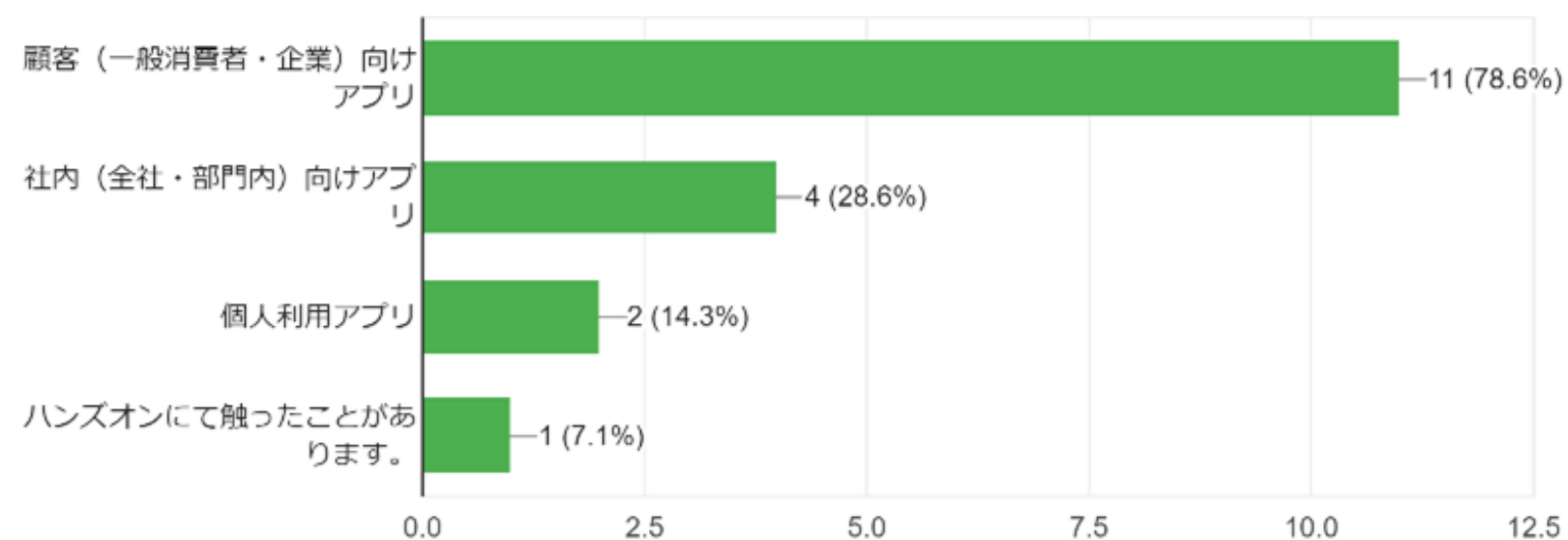
8人

ノーコード/ローコードツールを使用したことがある回答者は半数にも満たない

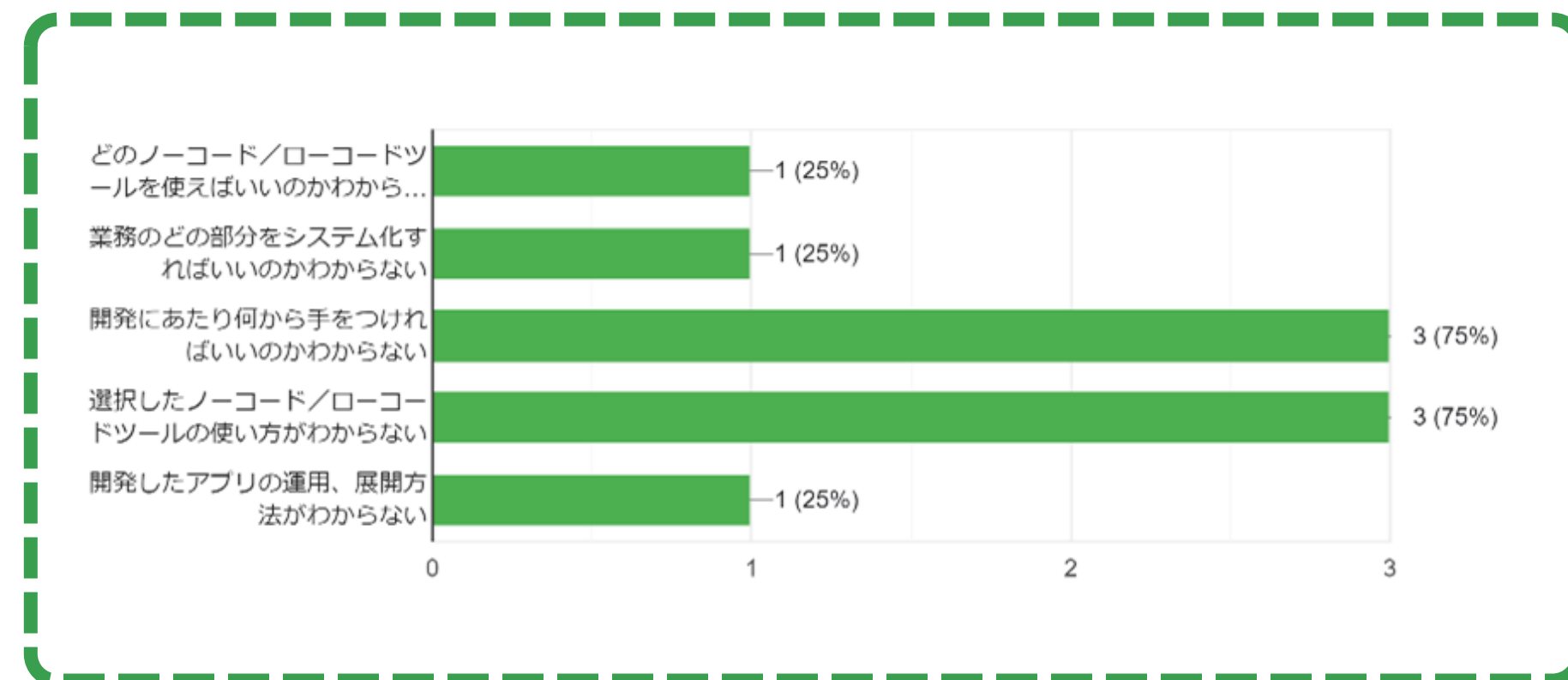


アンケート結果（属性）③

(1) どんなアプリを開発したか（経験者のみ）
（複数選択可）



(2) 開発途中で挫折した理由（経験者のみ）
（複数選択可）

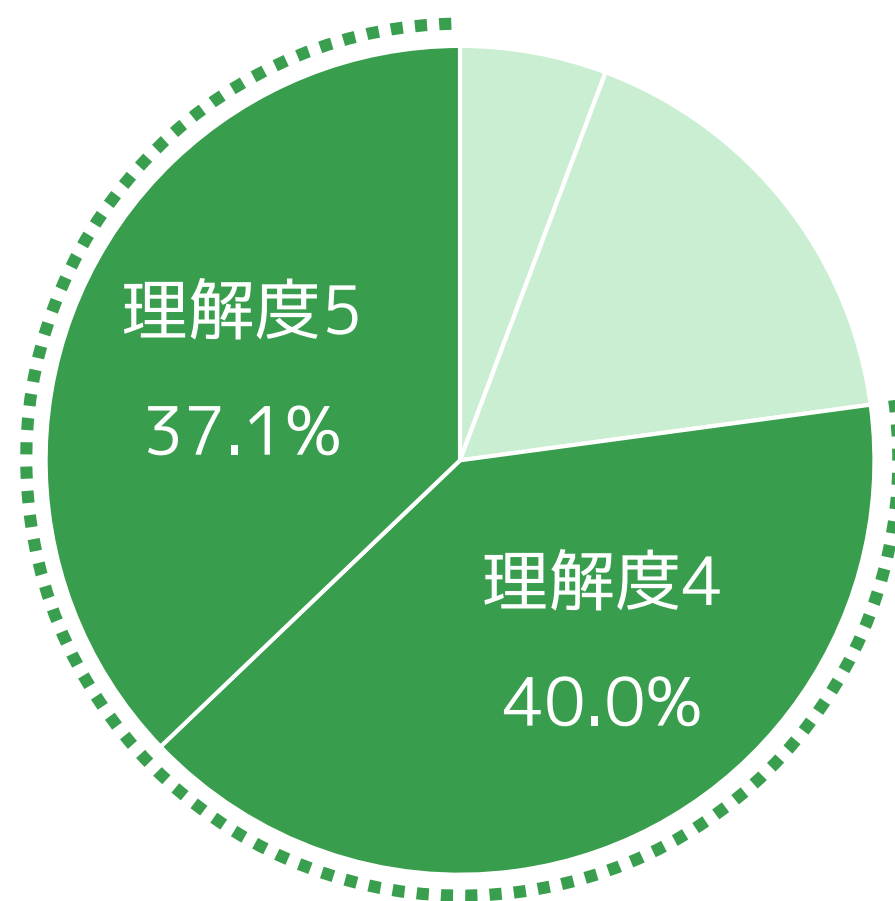


選定や開発、運用それぞれで迷い挫折している



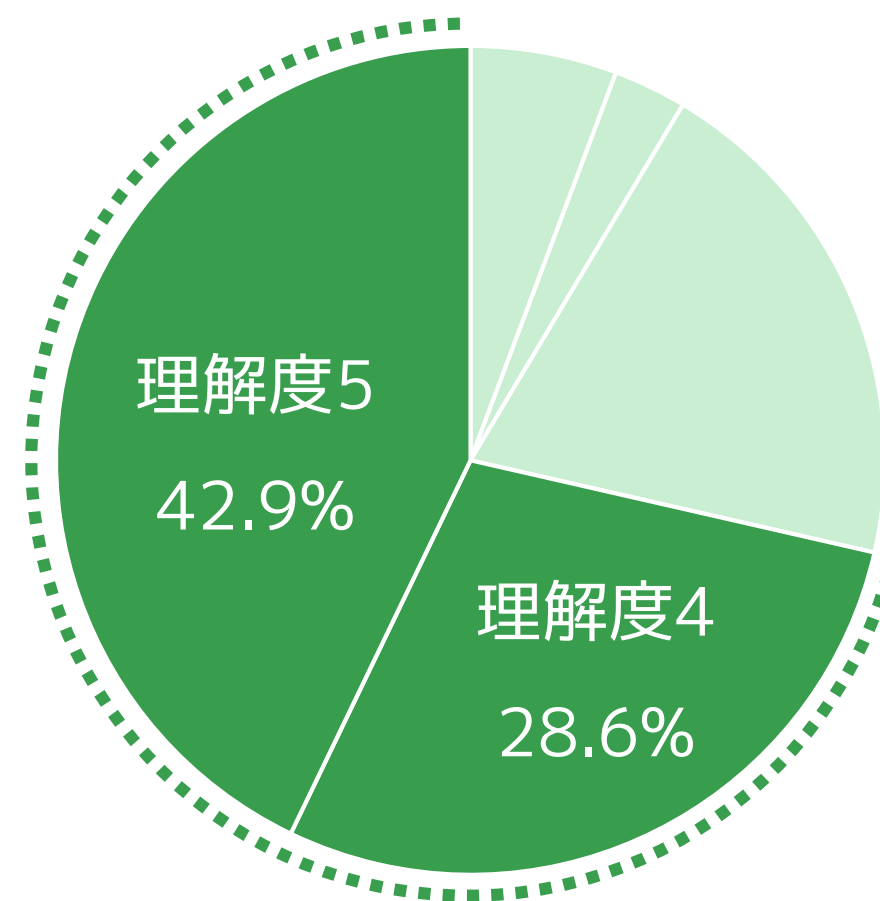
アンケート結果（内容は理解できましたか？）

「選定」章の理解度
(5段階)



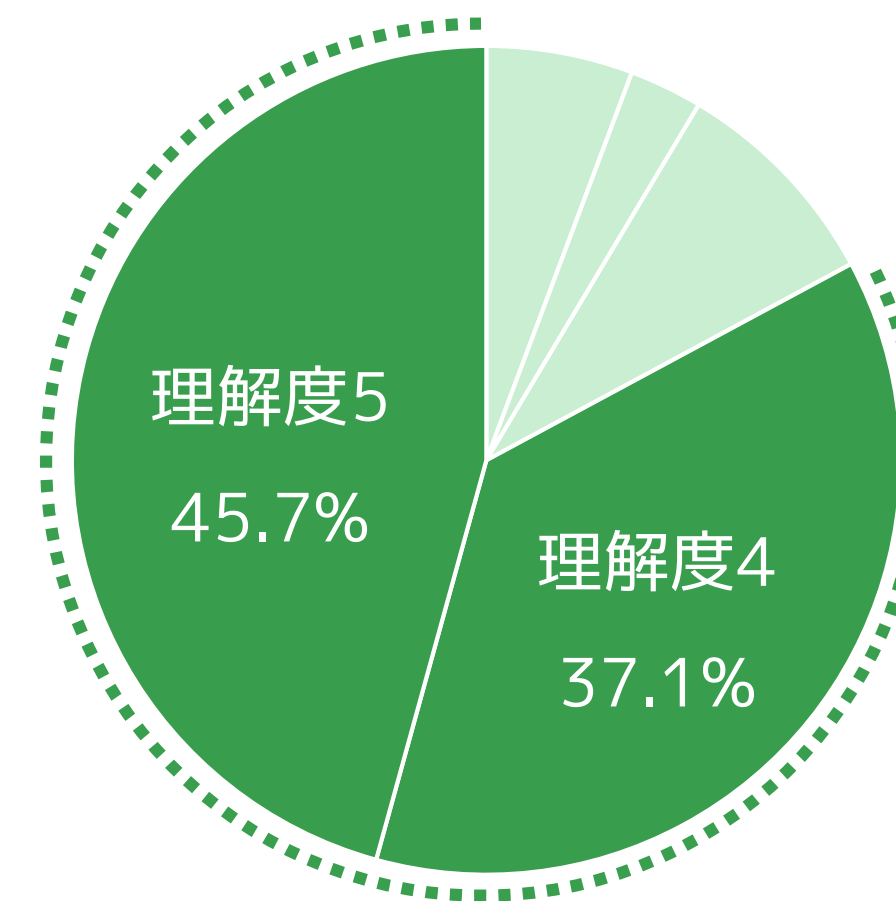
77.1%

「開発」章の理解度
(5段階)



71.5%

「運用・展開」章の理解度
(5段階)



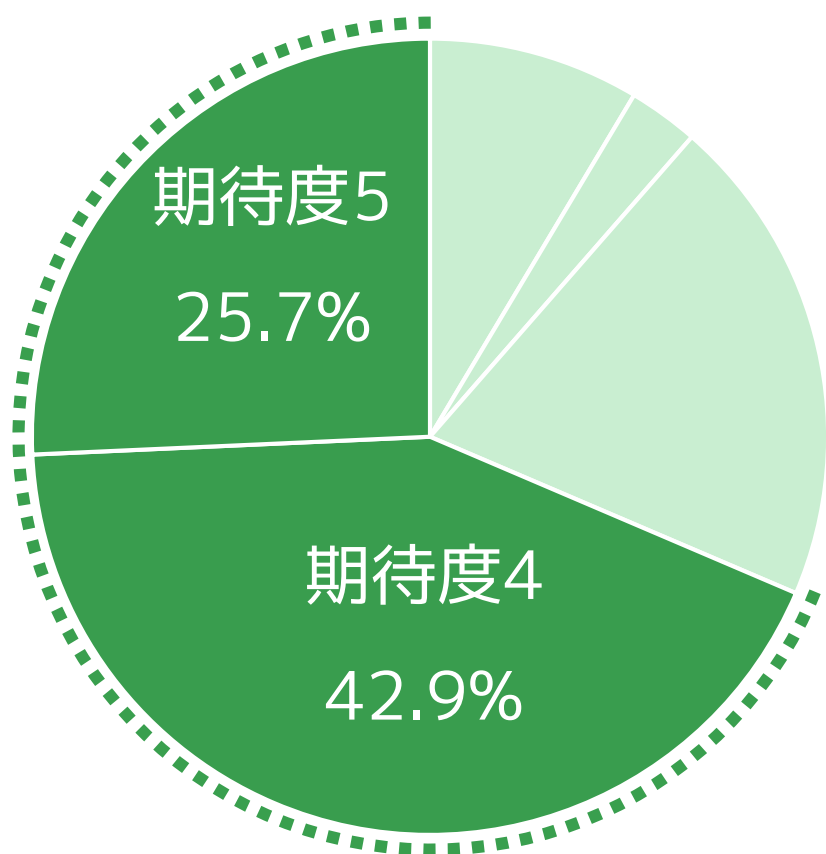
82.8%

内容の理解度は非常に高く、いずれの章でも7割以上の方が理解できたと回答



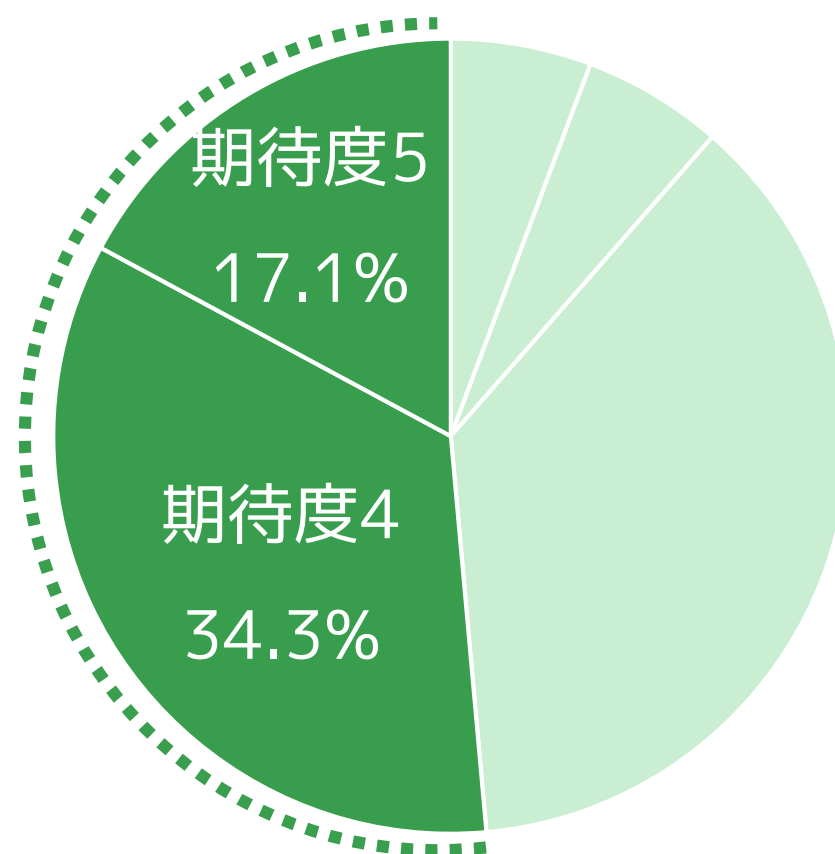
アンケート結果（活用できそうですか？）

「選定」章の活用期待度
（5段階）



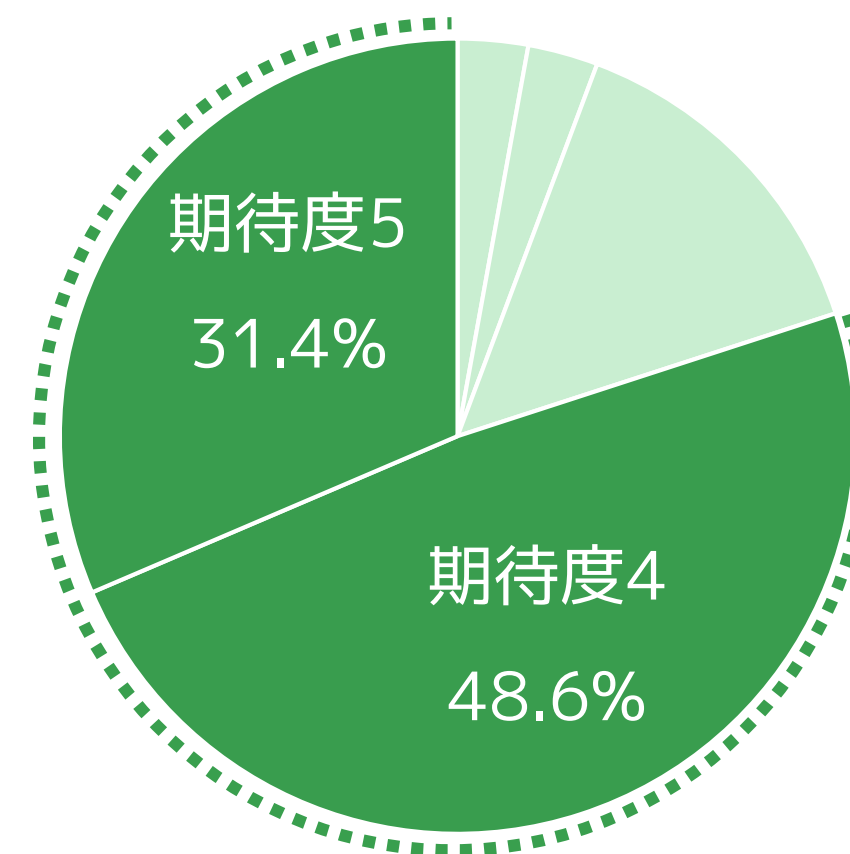
68.6%

「開発」章の活用期待度
（5段階）



51.4%

「運用・展開」章の活用期待度
（5段階）

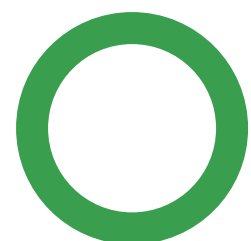


80.0%

「開発」章は半数以上、その他も3分の2以上の方が活用できそうと回答



回答者からのコメント



高評価、ポジティブなコメント

- サンプルの記載もありわかりやすかった。
- 業務整理は重要でその部分のプロセスが記載されていて参考になった。
- キーとなる工程が整理されており、システム開発で検討するポイントを知る機会となった。



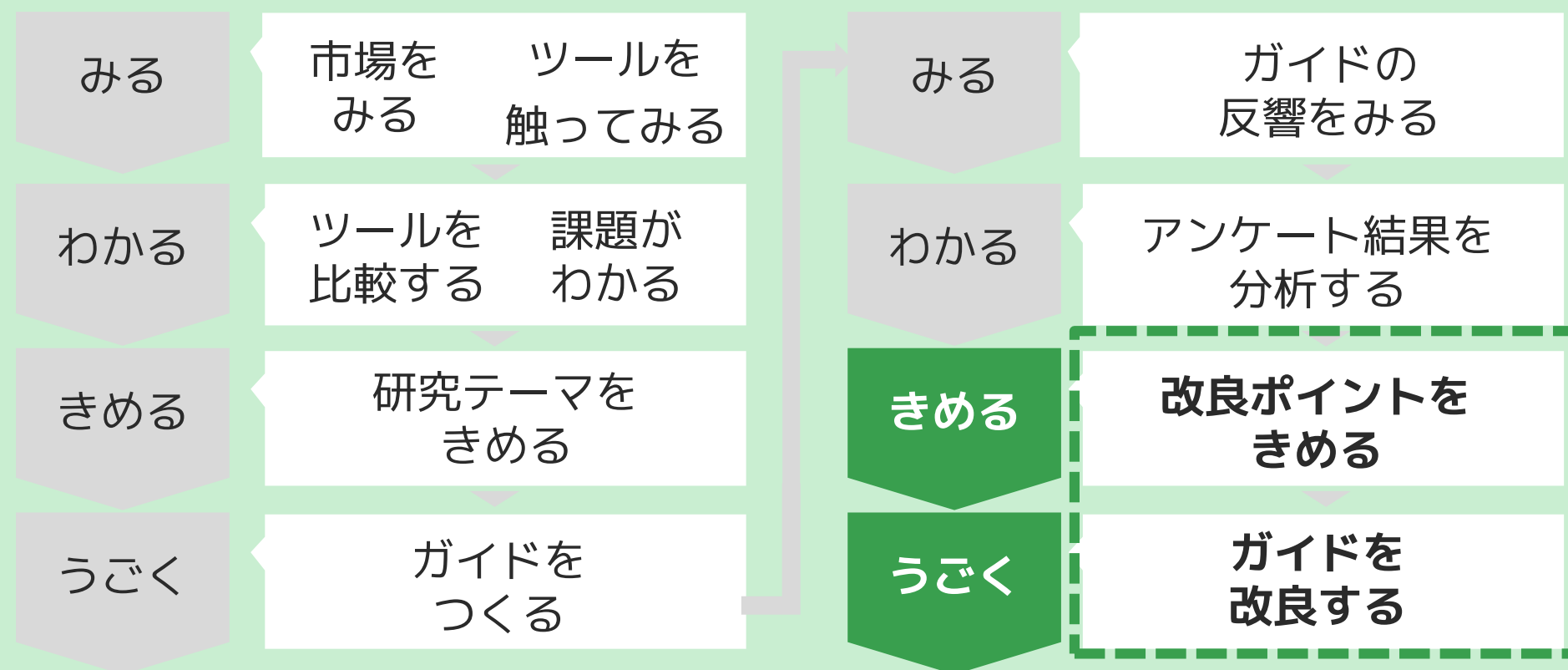
ご意見、ご指摘いただいたポイント

- ローコードまで考えると必要だと思うが、ノーコードをやりたい人には内容が重たい。
- 準備ややることが多く、「やることが変わっただけで忙しさは変わらないのではないか？」という気持ちになりそう。
- 理解できたが、どんなことができるのかといったところまで想像がつかない。

内容の網羅性は評価が高い。一方、内容が重たくなりハードルが高い印象

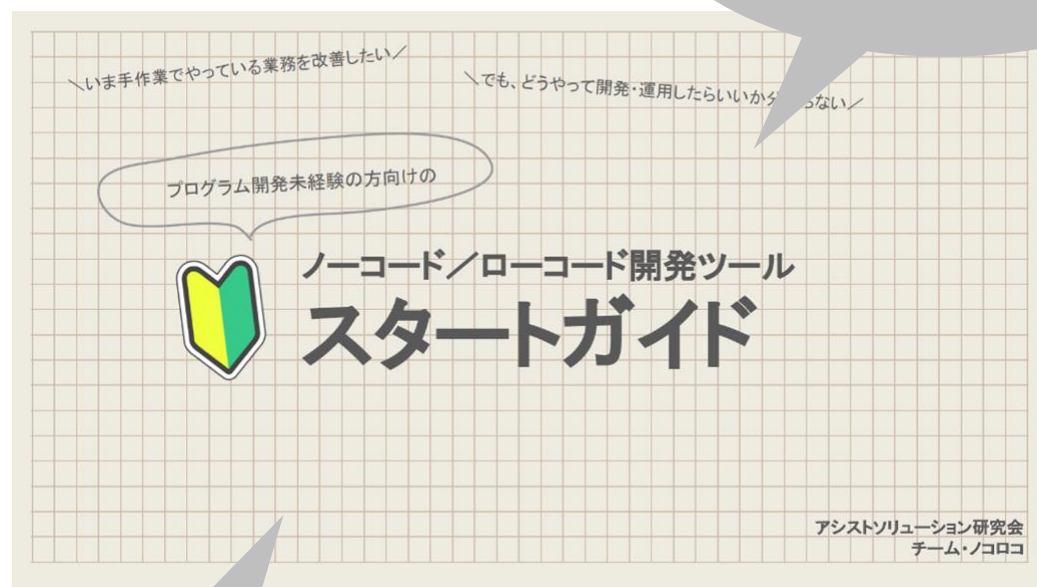


改良ポイントをきめる / ガイドを改良する



アンケート結果からの改良

(初版)



やることが多い...

内容が重たい...

想像がつかない...

(1.1版、1.2版)



1
ノーコードでも見て欲しいポイントを分かりやすく

2
難しい内容だからこそ表現を柔らかく

3
具体的な利用イメージがわくように

60項目の改良

初心者でもよりわかりやすくなるよう、大きく3つの観点で60項目を改良



改良ポイント1：ノーコードでも見て欲しいポイントを分かりやすく

ガイドラインの見方

The image shows a document page titled "ガイドラインの見方" (How to view the guidelines). The page content includes a section "3.1 事前準備" (3.1 Preparation) with three numbered points. A callout box on the right explains the labels: "ノーコード" (No-code) and "ローコード" (Low-code). The "ノーコード" label is for content applicable to all, while the "ローコード" label is for more difficult content for low-code users. A second callout points to a "コラム" (Column) section, explaining that detailed explanations are provided in separate pages before the main content.

3.1 事前準備

開発するためには事前に様々なことを確認しておく必要があります。ここでは主な項目について記載します。

- 1 開発する前に押さえるべきポイント
開発のルールについて確認をすることがあります。
開発範囲や業務運用の変化など、開発の詳細ではなく業務全体の観点で問題ないかを確認してください。
- 2 ノーコード/ローコード開発に関する全般を管理する部門（担当者）が確認する事項
開発を管理する部門（担当者）は、開発工程をスムーズにする役割があります。
そのため、重複する機能を持つシステムの有無、担当者ごとの役割などを確認する必要があります。
- 3 開発手法の選択
開発手法についてあらかじめ決めておく必要があります。「作りながら仕様を決めていく方法」と「設計書などをすべて作成してから開発する方法」があるため、進め方を関係者で共有し開発に着手してください。

ノーコード
ローコード

ノーコード ノーコードにも当てはまる、全ての方に読んでいただきたい内容になっています。

ローコード ローコード

ローコード ローコード向けの少し難しい内容です。システム開発について詳しく知りたい方は読んでみてください。

ローコード ローコード

コラム 業務整理のためのフレームワーク

業務整理にお役立ちなフレームワークをご紹介します。
迷ったり、行き詰ったりした時には参考にしてみてください。

アンケート
業務内容についてアンケートを行います。
ポイント：質問は簡潔にし、回答者の負担を減らすと回答を促しやすいです。

インタビュー
業務内容のインタビューを行います。
ポイント：管理職だけでなく担当者も合わせて実施すると良いです。

プレインストーメンク
課題を共有し意見を交換します。
ポイント：考えすぎず、思いつづものを早く書き出していくことです。

グルーピング（MECE）
課題の共通部分を抽出して、グルーピングします。
ポイント：抜け漏れがない（MECE）状態にしましょう。

ロジックツリー
課題を上位概念と下位概念に分解します。
本質的な問題が「どこに」「なぜ」存在するか、分かるようにします。
ポイント：課題間の関係性・因果関係を注視しましょう。

ロードマップ
将来達成に必要と思われる事項を時系列で書き出します。
ポイント：具体的なアクションにつなげるため、目標を設定しましょう。

アクションプラン
行動の計画を、アクションプラン表に記入します。
ポイント：「誰が」「いつ」「どこで」「どのように」実施するの4つを記載しましょう。

各章の詳細な説明ページの前に、内容を簡潔にまとめたページを用意しました。

スライド右上に「ノーコード」「ローコード」のラベルを記載して、ローコード向けのやや難しい内容は読み飛ばせるように改良



改良ポイント2：難しい内容だからこそ表現をやわらかく



3.2.4 データベースの設定(1)

データベースとは、システムにおいてデータを貯めておくためのツールです。ツールによって、決められたデータベースしか使用できないもの、データベースの選択が必要なものがあります。

この章では、「リレーショナルデータベース(RDB)(※)」と呼ばれるデータベースについて説明します。

※Excelの表のような形をとったデータベース。行と列で構成される。
【代表的なRDB】PostgreSQL, MySQL, OracleDBなど

データベースの設定は、後から変更するのが難しい(もしくは不可能な場合も)です。また、テーブル構成によって、処理効率に大きな影響を与える場合もあります。そのため、設計段階で慎重に検討する必要があります。

テーブル構成

- 正規化
一つの要素から他の要素が決定される場合、別のテーブルに分ける必要があります。これを正規化といいます。第3正規形、またはあえて正規化しない非正規形にすることが多いです。
非正規形に向いているのは、Webのアクセスログなどのログ系に多いです。
→ 詳しくは『コラム「正規化」とは』へ
- 複数テーブルでのデータの紐づけ(外部キーの活用)
外部キーとして別テーブルの値を利用すると、データ探索の処理効率が上がったり、データが存在しない場合にエラーとして検知することができます。

40



3.2.4 データベースの設定(1)

データベースとは、システムにおいてデータを貯めておくためのツールです。ツールによって、決められたデータベースしか使用できないもの、データベースの選択が必要なものがあります。

本スタートガイドでは「リレーショナルデータベース(RDB)」を前提に説明をすすめていきます。

リレーショナルデータベース(RDB)とは

Excelの表のような形をとったデータベースで、行と列で構成されます。1つの表をテーブルと言います。また、複数のテーブル同士の関係を定義することで、複雑なデータの組み合わせも扱うことができます。高度な検索を行うことができるのが特徴です。一方で、データを扱う場合は、件数が多いほど処理に時間がかかるため、Webのアクセスログなどの大量にあるデータの扱いには向きません。

【代表的なRDB】
PostgreSQL, MySQL, SQL Server, OracleDBなど

48



イラスト多め、文章少なめ、難しい表現や専門用語はなるべく使わず、使う場合は注釈入れるなど改良

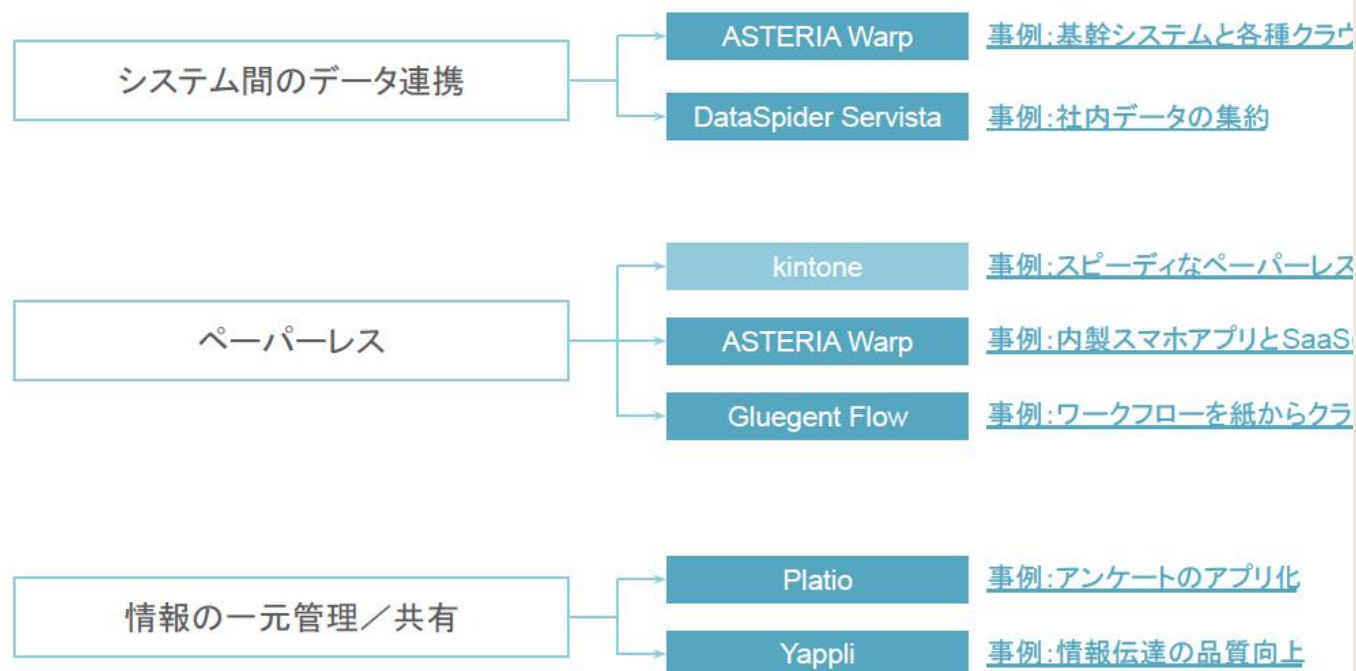


改良ポイント3：具体的な利用イメージがわくように

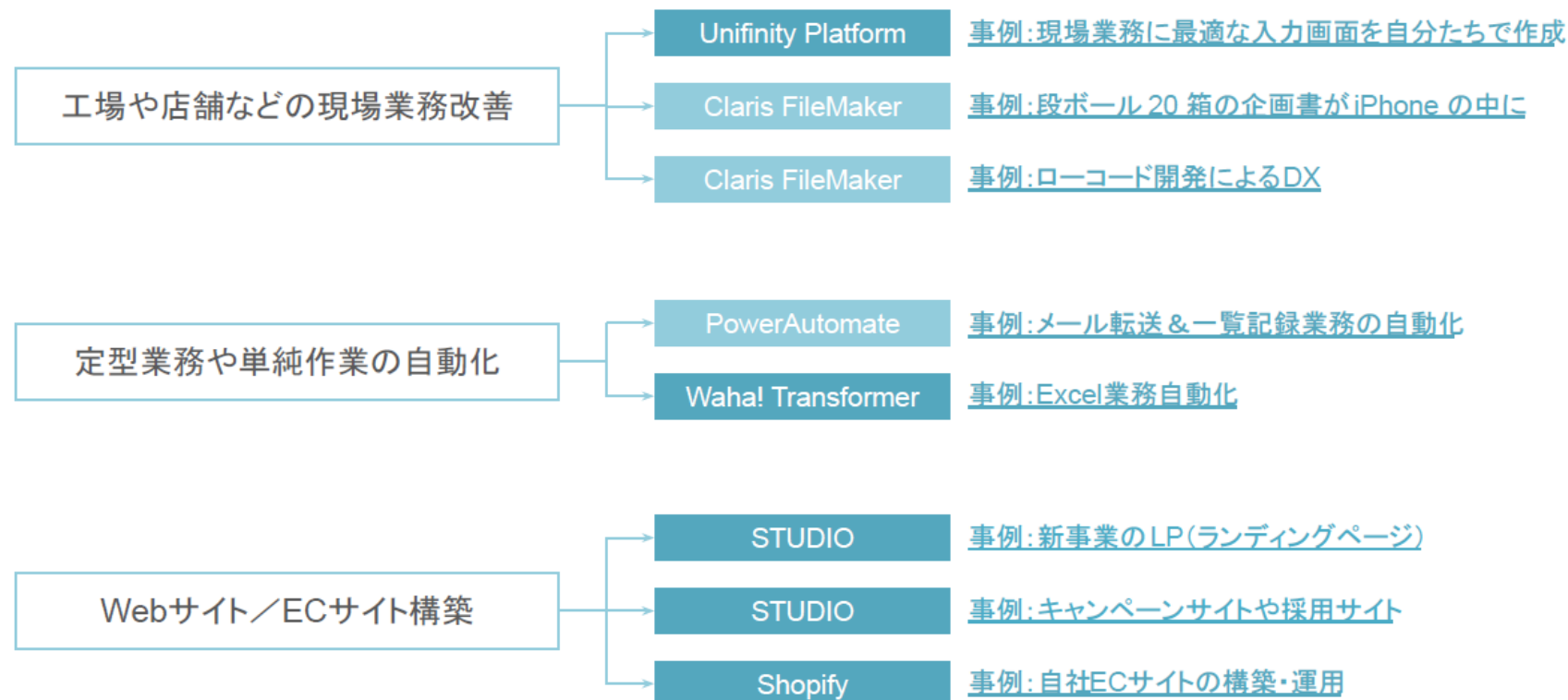


2.2.3 ツール選定フロー(事例)

改善したい内容別の導入事例を紹介します。業務選定～ツール選定の参考にしてみてください



2.2.3 ツール選定フロー(事例)



概要

選定

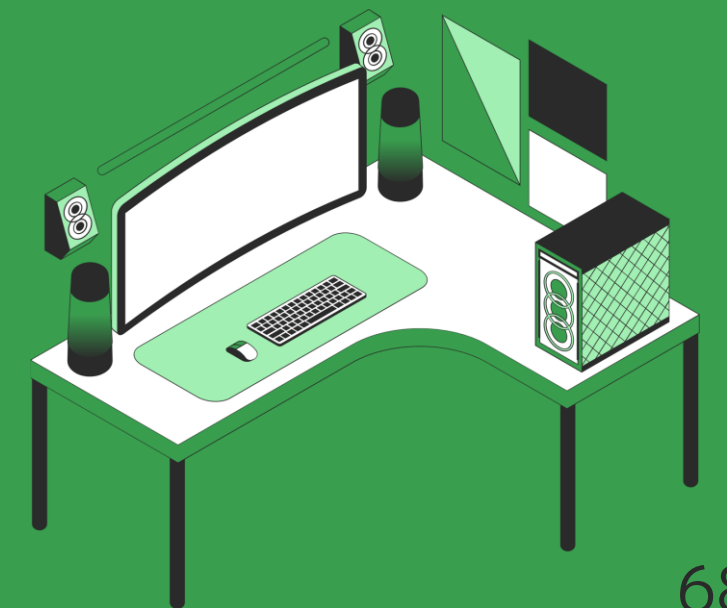
開発

運用

実際のツール名や事例情報も掲載して、利用イメージがわくように改良



IV. 総括と展望



まとめ



まとめ

「みる」「わかる」で課題設定し、研究テーマを「きめる」、そして「うごく」という流れで活動しました。



まとめ

二巡目のOODAループで実際にガイドをリリースしてその有用性を検証しました。

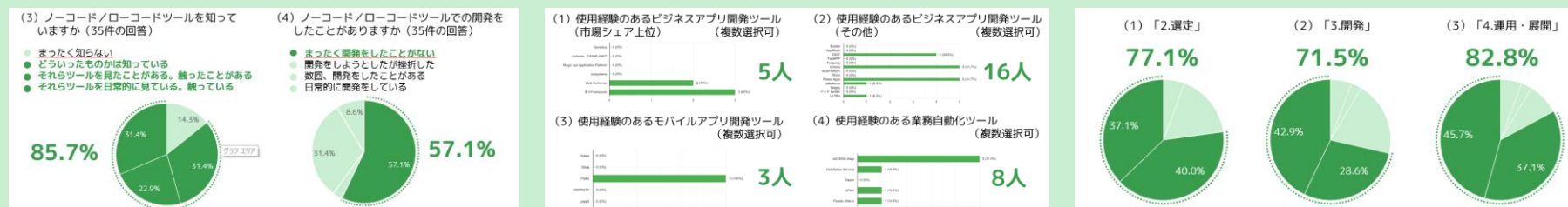
Observe
みる

ガイドの
反響をみる



Orient
わかる

アンケート結果を
分析する



Decide
きめる

改良ポイントを
きめる

60項目の改良ポイント

Act
うごく

ガイドを
改良する



改良版リリース

スタートガイドは、初心者の「選定」と「開発」の手助けとなり
また「野放し」に対しても一定の道標を示すことができた



総括



総括

メンバーの多様性を活かした議論

メンバーの立場や開発経験、知識のばらつきがあったからこそ、どこまで記載すべきか、どういった用語にすべきかの議論ができ、結果として読み手にとってわかりやすいガイドができました。

OODAループを経てリアリティのあるガイドを作成

OODAループ（みる、わかる、きめる、うごく）をベースに、自分たちで実際に開発を行ったことは、スタートガイドにおいてリアリティを埋め込むことができたと考えています。

仮説の立証

スタートガイドが有用であると評価いただいたことで、ガイドによって初心者であってもノーコード/ローコード開発ができるだろう、という仮説を立証できたと考えています。

メンバー所感



青木

この**活動で得た教訓は今後PJを進めていく上で活かせるものが多く**、非常にありがたい経験をさせていただいたなと感じております。

今回作成したスタートアップガイドは、**ノーコード/ローコード開発に十分活用できる**と思います。



武田

ガイドの表現方法や内容の精査には苦勞しましたが**最終的にはメンバーの思いが一致した**良い活動ができたと思います。

ミッションへの思いをメンバー間で共有でき、気持ちを途切れさせず活動できたのが良かったと思います。



森山



田中

メンバー所感



小久保

メンバーの皆さんと時にぶつかりながらも活動し、**最後まで悩みぬいただけあって満足のいくものが出来た**と思います。

どのようなポイントを押さえ、開発を進めていけば良いのか？
「本当に活用できる、嬉しいシステム」の実現にガイドを活用いただけると幸甚です。



宮澤



松尾

ガイドを作って終わりではなく**分科会から外へ出て声を集めその有用性を確かめる**ところまでこだわることができたこのチーム最高です。

展望



そして、スタートガイドの今

スタートガイドを策定しその有用性も検証しましたが、検証に協力いただいた皆さんの声をふまえると、もっと良いものしたいとの思いが芽生え…

実はその後もスタートガイドの改訂を続けました。

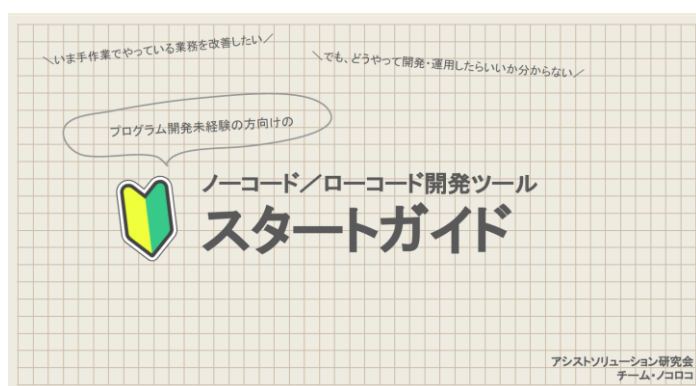
2022年11月

12月中旬

12月末

2023年2月中旬

3月(現在)



有用性
検証
実施



ソ研
事務局
提出



中日本
発表会



初版(1.0版)

改訂版(1.1版)

改訂版(1.2版)

改訂版(1.3版)

【実例】ガイドを使ってWEBアプリを仕上げた個人経営アパレル店長さん

分科会メンバーの知人でExcel・Wordは使えるが開発は未経験の店長さんが、自社商品の販売サイト開発に苦勞されていたので、スタートガイドをご提供しました。



当初は画面操作を数パターン試した
のみでリリースしてしまった…

開発の基本を知り、しっかり
テストも実施して無事再リリース

開発経験のない方を、このスタートガイドでお手伝いできた！



やまない私たちの想い

加えて、当初掲げた私たちのミッションの実現に向けて、以下も進めつつあります。

スタートガイド
ウェブ公開

所属企業で
スタートガイド
勉強会

スタートガイド
説明動画
社内公開

“ 本分科会で研究した内容をガイドラインとしてアウトプットし、私達と同じようにノーコード/ローコードツール活用に悩める人達に貢献したい！

皆さんのDX推進に
このスタートガイドをぜひ活用ください！

ご清聴ありがとうございました

チーム・ノコロコ

