

2024年度 アシストソリューション研究会
ノーコード/ローコード時代のシステム開発分科会

現場発信のIT時代！

ノーコード・ローコード・ヒーロー

への挑戦



メンバー紹介

リーダー



村上開明堂
野仲 泰志

サブリーダー



村田機械
山下 達也



NDSソリューション
沢野 立明



アイカ工業
藤本 心



愛三工業
橋本 隆史



ジャペル
野々田 健吾



日東工業
田中 佑樹



菱友技研
村山 竜太



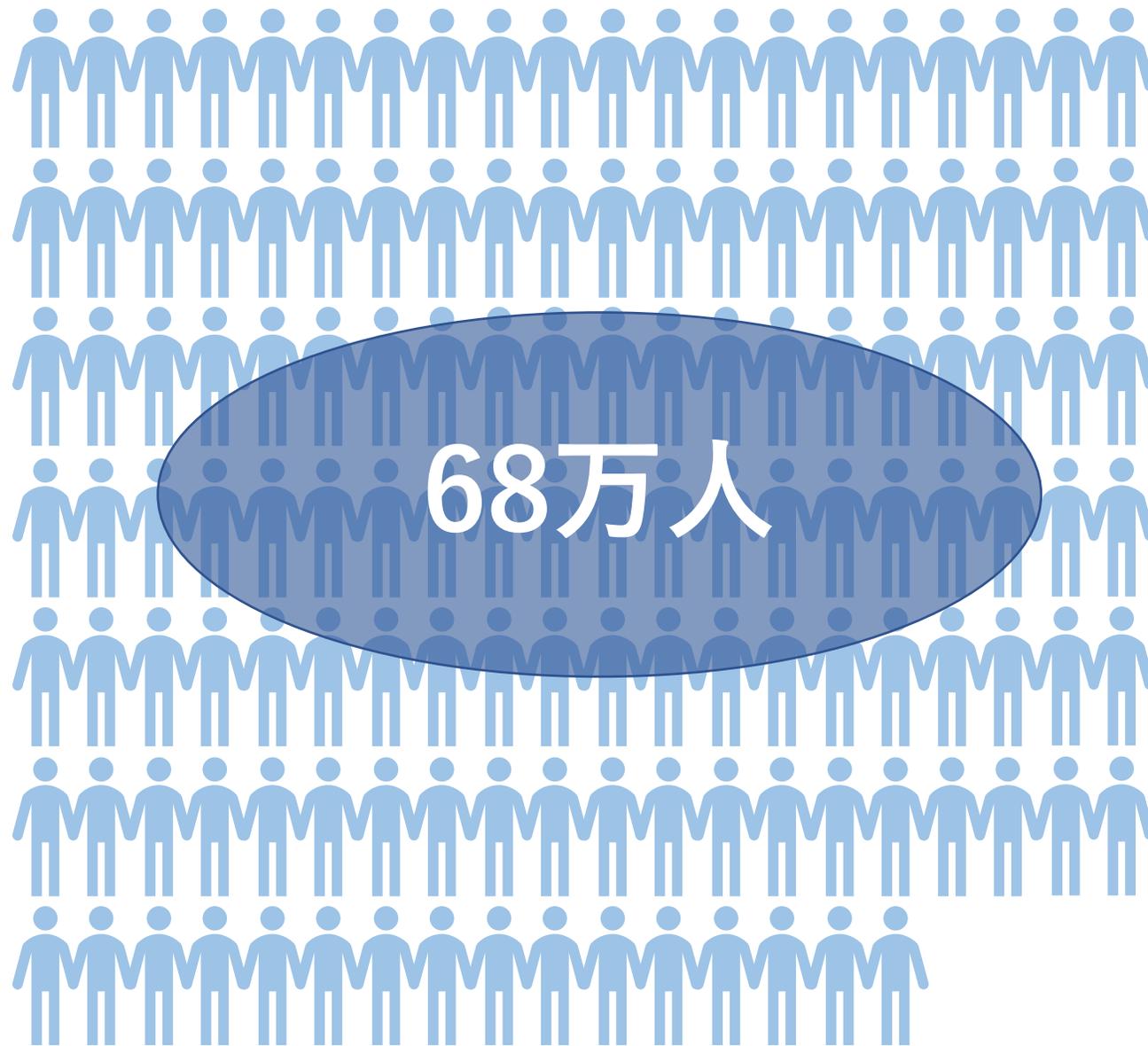
TIS
尾関 朋子



アシスト
河村 祐希



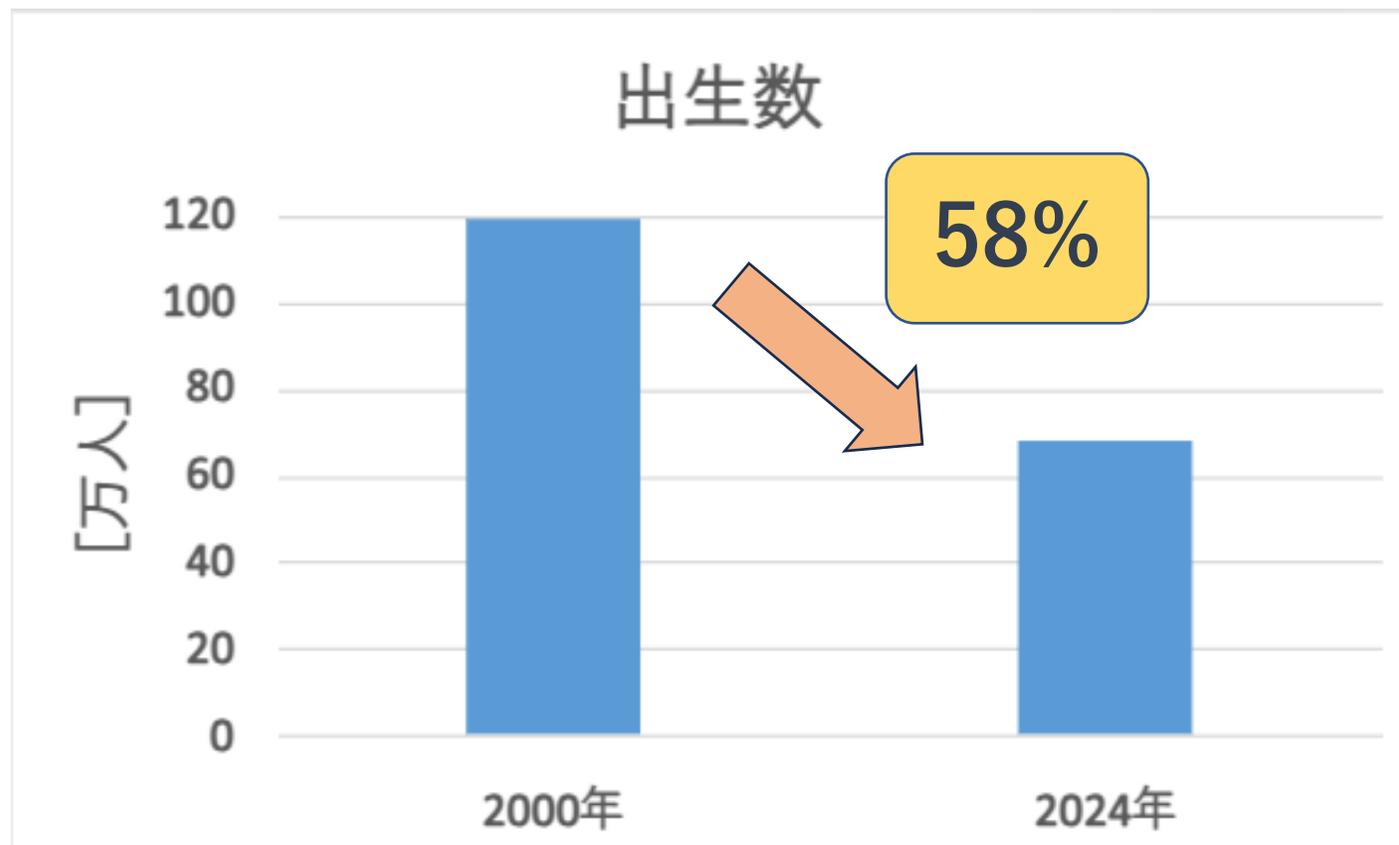
アシスト
飯田 恵里香



68万人

出生数は減り続けている

初の68万人台となる見込み

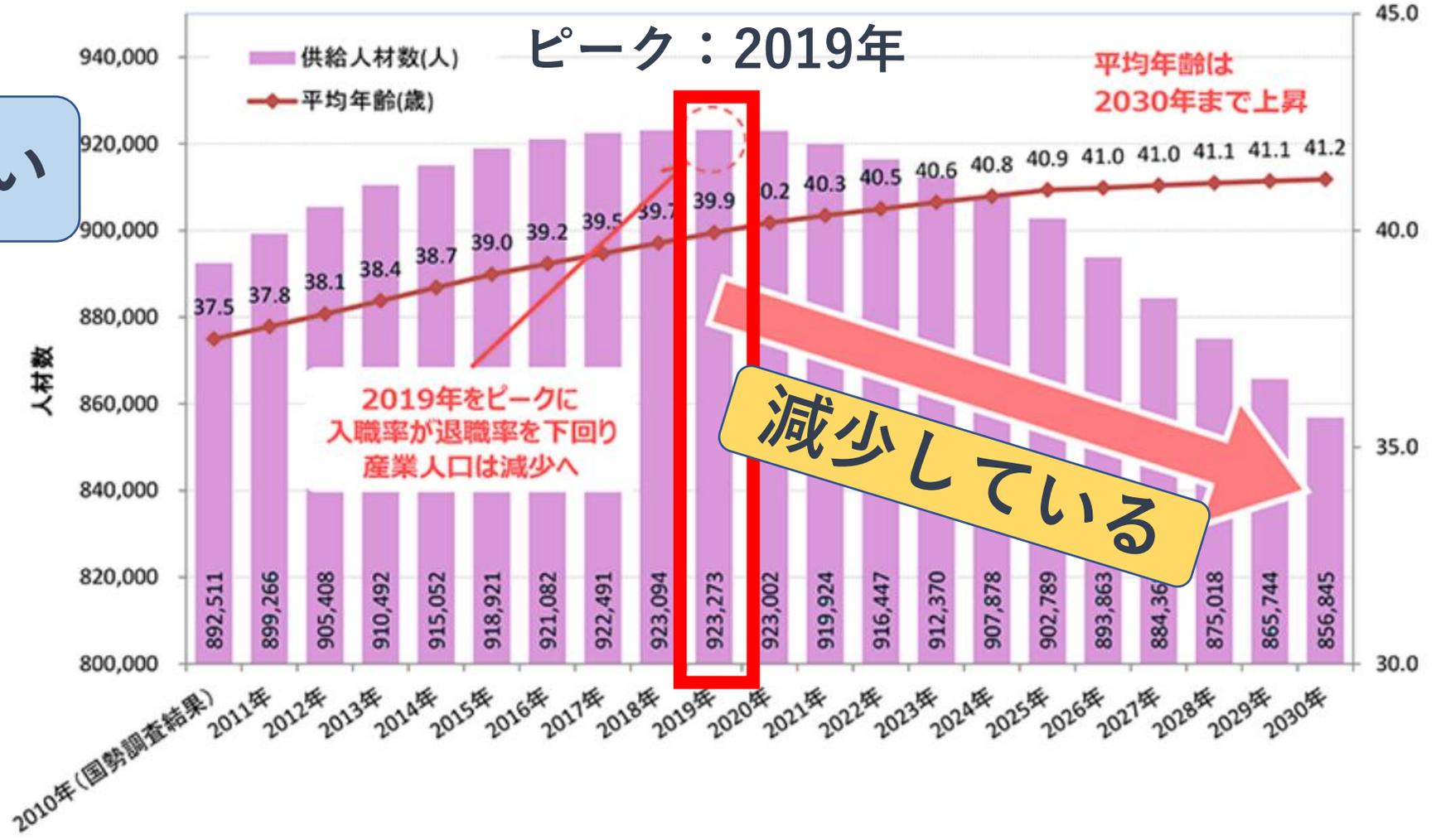


出典：「人口動態調査」より抜粋(厚生労働省)

労働人口減少 ⇒ IT人材不足

IT人材の供給動向の予測と平均年齢の推移

人材が足りな～い



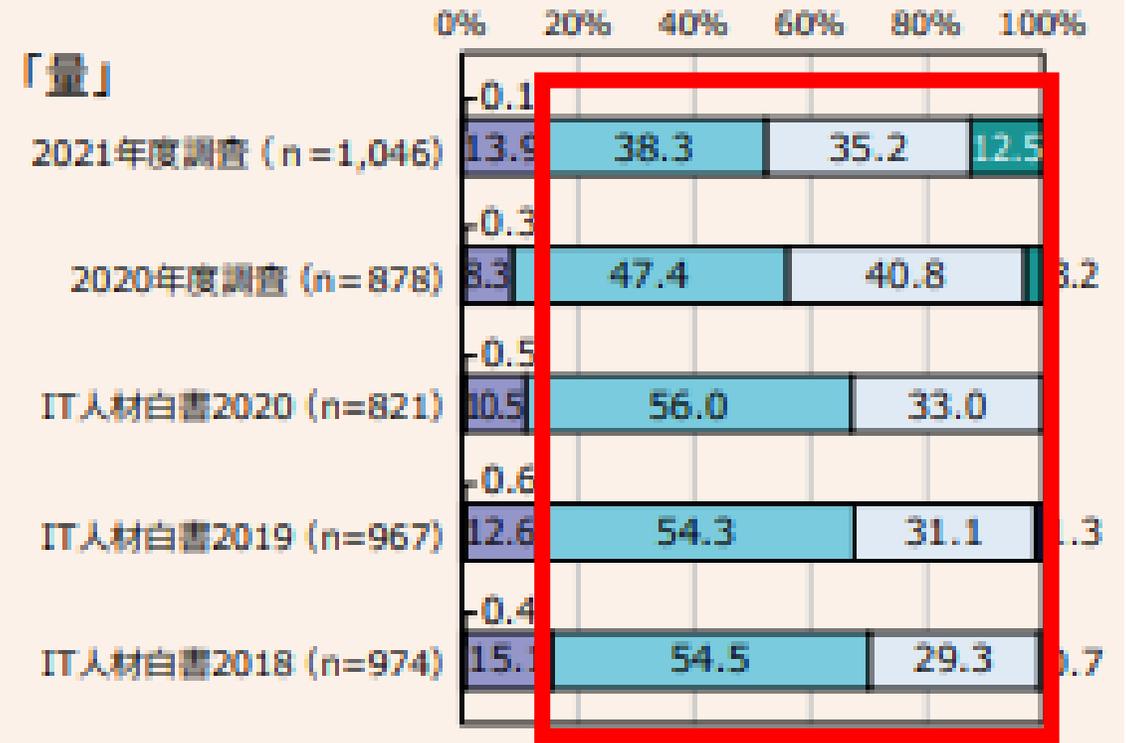
出典：「IT人材の最新動向と将来推計に関する調査結果 報告書概要版」(経済産業省)

多くの企業でIT人材不足

業務量が増えているが
人材は増えない…



人材の過不足について



- やや過剰である (別表は過剰に回答した企業)
- 特に過不足は無い
- やや不足している
- 大幅に不足している
- 分からない
- 無回答

多くの企業が
不足と回答



IT人材を補充したい！

業務

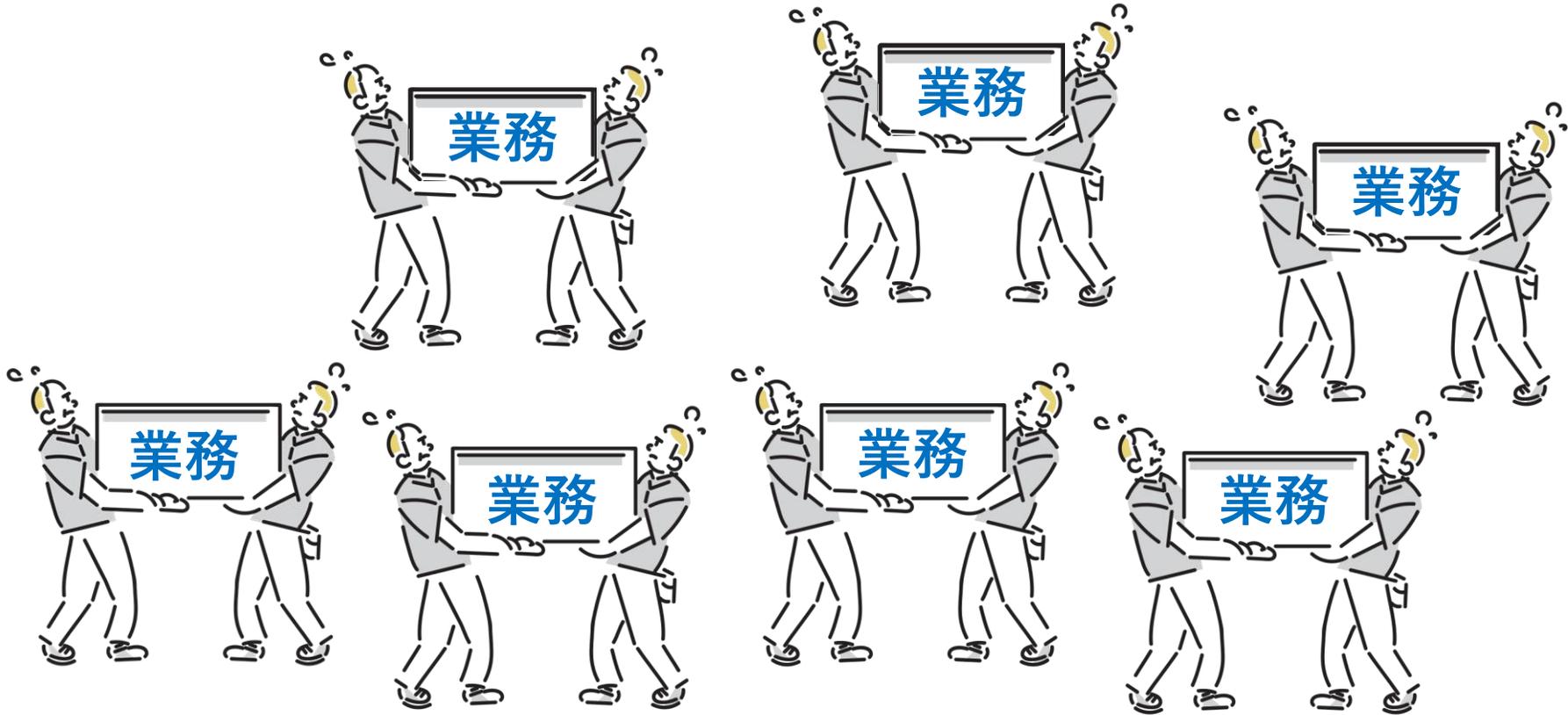
業務

業務





獲得競争が激しい、採用コスト高騰





人を増やさずに
どうやって対応していけばいいの？

業務
業務

業務
業務

業務
業務

業務
業務

業務

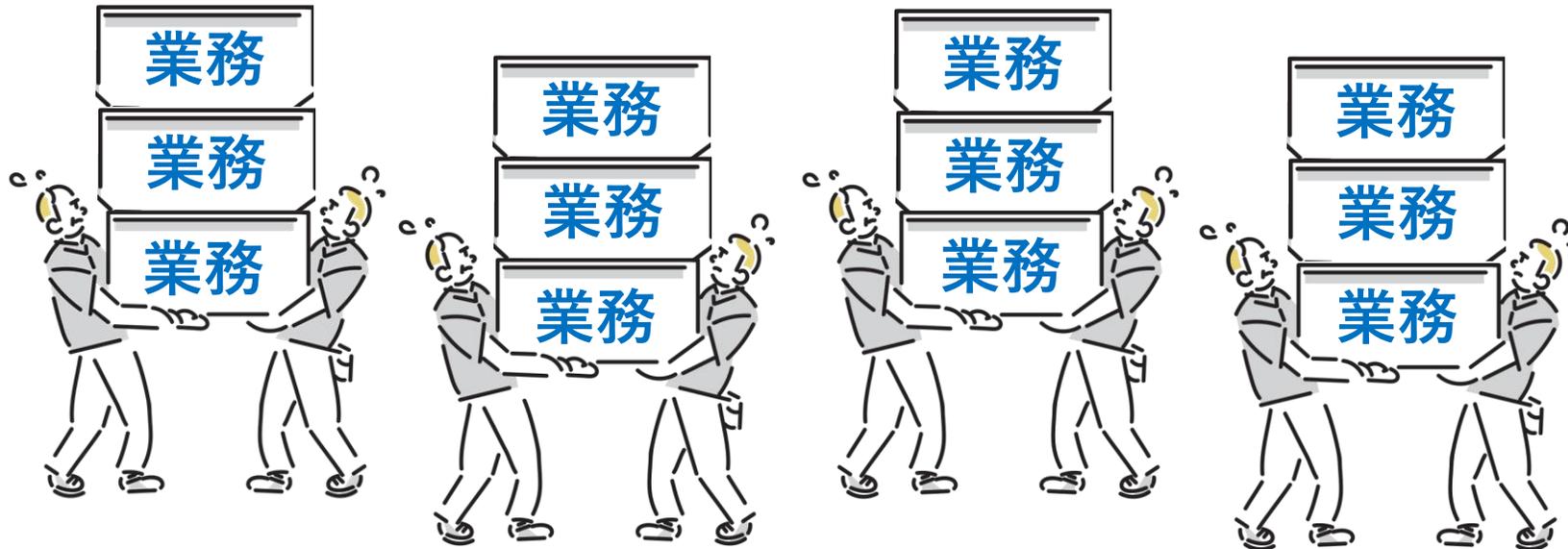
業務

業務



IT人材不足に対応するには

全社一丸となってITツールを
活用する時代がやってきている！



全社一丸となったITツール活用

IT業務の整理

業務

業務

業務

業務

全社一丸となったITツール活用

適材適所に振分け

基幹システム

セキュリティ対策



IT部門

データ抽出解析ツール

回覧承認システム



ノーコード/ローコード活用！

非IT部門

目次

ノーコード/ローコードの必要性

✓不足するIT人材

○ノーコード/ローコードツールという打ち手

当分科会の研究と提案

ノーコード/ローコード活用チャートの提案

ノーコード/ローコード活用チャートの有効性

まとめ

ノーコード/ローコードとは

LOW-CODE/NO-CODE PLATFORM ECOSYSTEM



プログラミング言語の知識がなくても、
アプリやウェブサイトを作成できる



視覚的なインターフェースで作成するため、
開発期間を短縮することができる



専門知識が不要なため、
学習コストを削減することができる

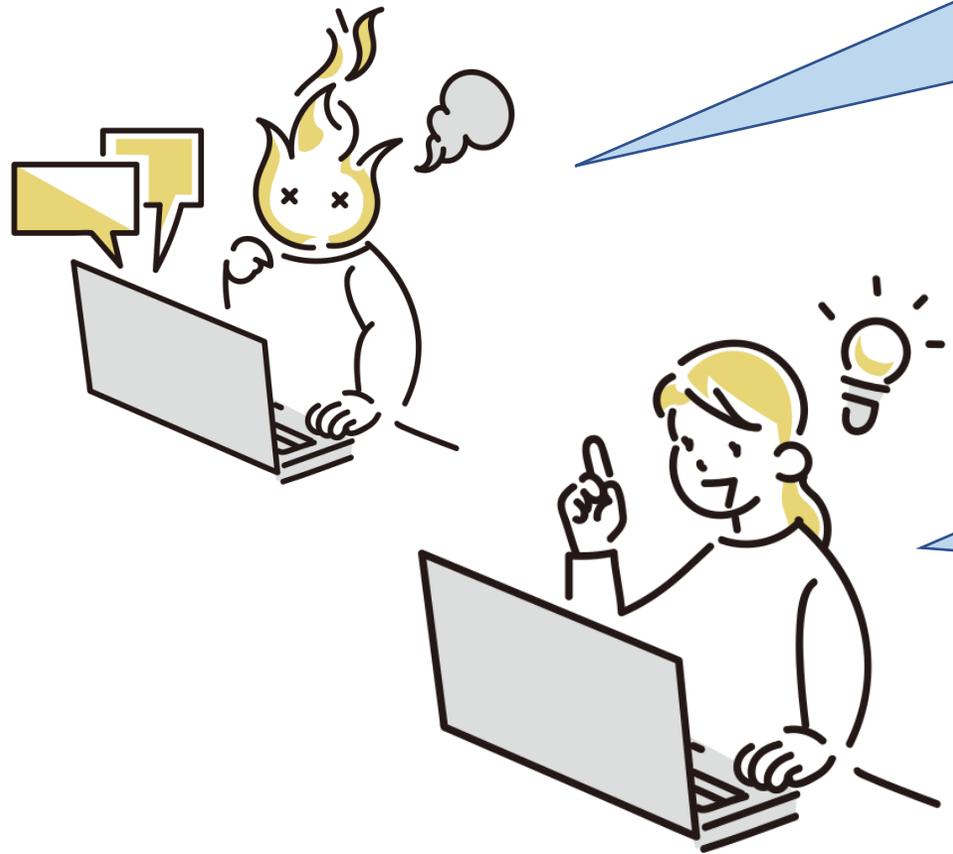


ノーコード/ローコードとは

ノーコードとローコードにも違いはあるが…
要するに

専門的知識の無い非IT部門でもできる
アプリケーション開発ツール

ノーコード/ローコードの活用



非IT部門からの依頼が沢山…
首が回らなくなる…

非IT部門に
ノーコード/ローコードツールで
開発をして貰おう！

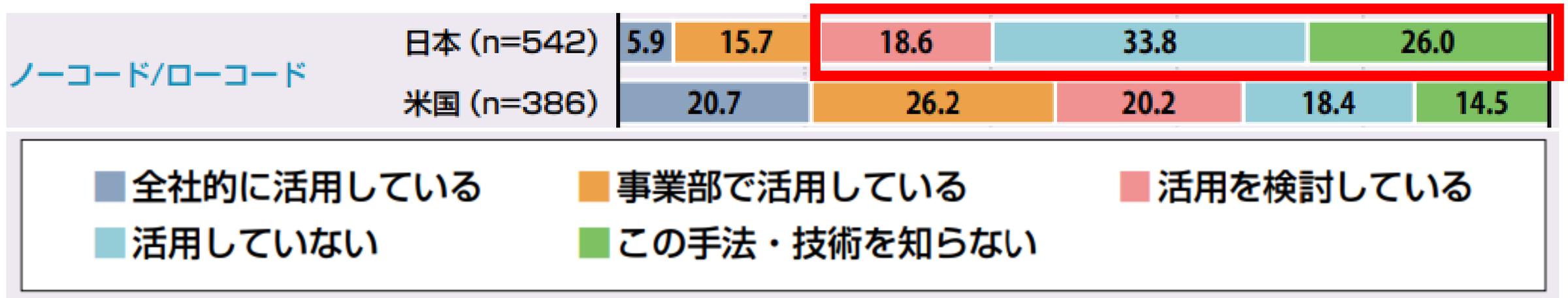
ノーコード/ローコードの活用



IT部門の業務減少！
非IT部門アプリの納期短縮！
Win-Win！

非IT部門のスキル向上！
問い合わせも減るかも！？

活用できていない…



約**8割**の企業で
活用できていない！

活用の課題(非IT部門)



開発知識が足りない…



何から学べばいいかわからない…



一緒にやってくれる人がいない…

活用の課題(IT部門)



知らないシステム(野良システム)の
問い合わせが来た…



結局サポートで
業務が増えるのでは…



教育する時間がない…

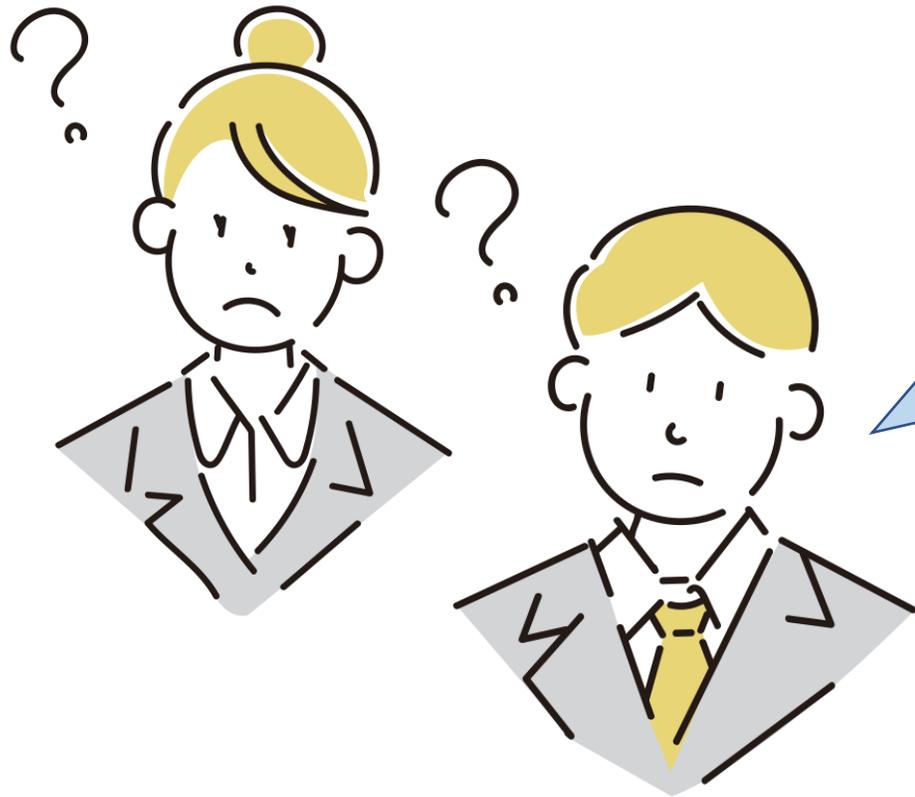
現状の課題(まとめ)

最適な勉強/教育方法が分からない

チームでの開発体制が整っていない

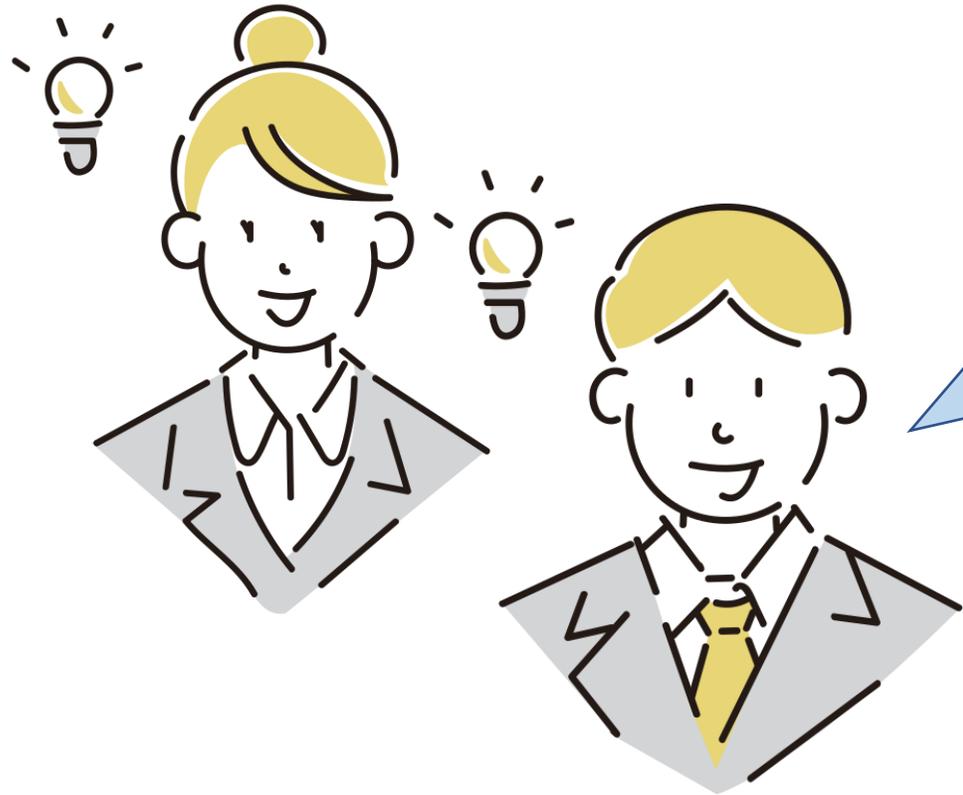
運用管理が考えられていない

課題に対して…



どうすれば
ノーコード/ローコードツールは
活用できるの？

課題に対して…



現状の課題を解決してくれる
ヒーローのような存在に
現れてほしい…



我々の手で
ヒーローを生み出そう

目次

ノーコード/ローコードの必要性

- ✓ 不足するIT人材
- ✓ ノーコード/ローコードツールという打ち手

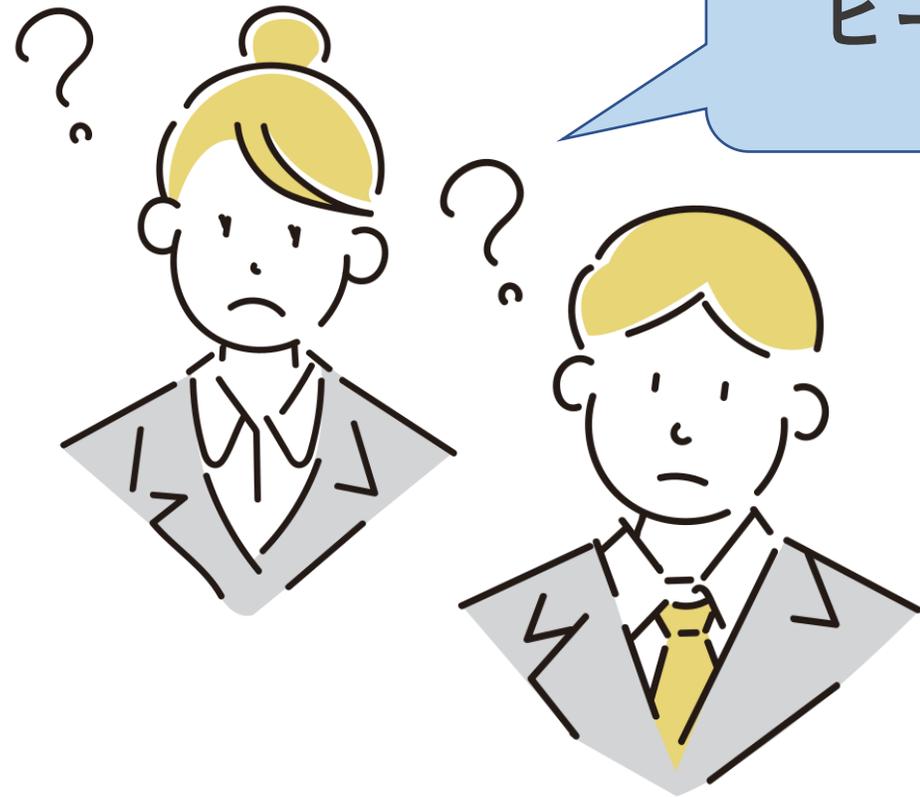
当分科会の研究と提案

○ ノーコード/ローコード活用チャートの提案

ノーコード/ローコード活用チャートの有効性

まとめ

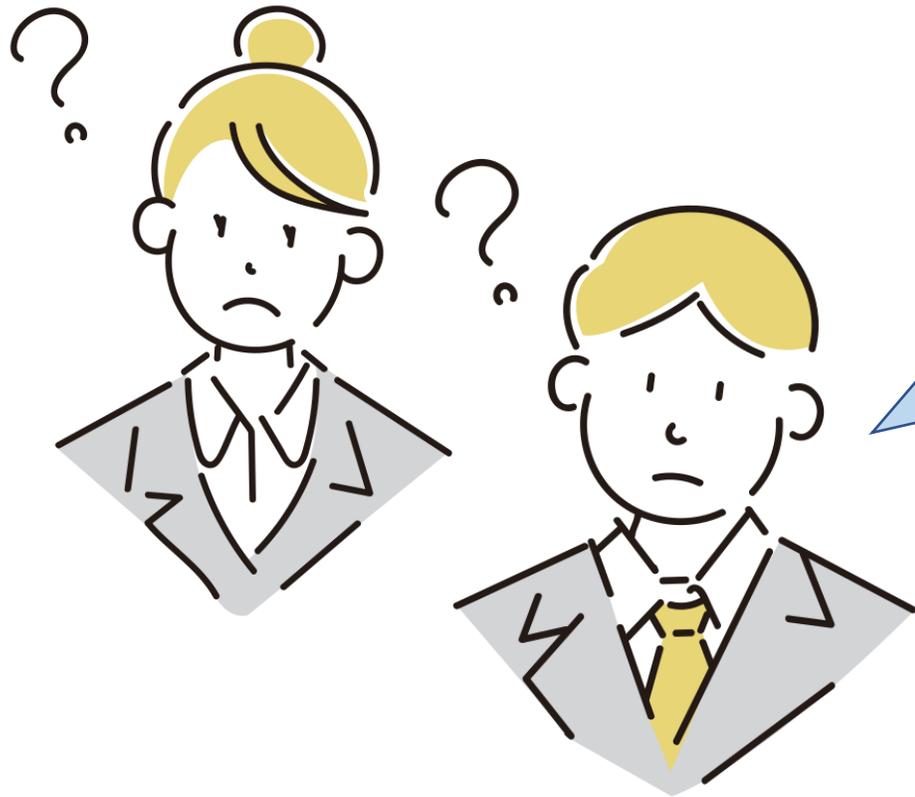
ヒーローを生み出すには？



ヒーローを生み出すにはどうしたら？

IT部門に余裕はないけど…

ヒーローを生み出すには？



非IT部門が率先して読んで
率先して学んでくれて
ヒーローになってくれるような
ロードマップがあれば良いのに

というわけで

非IT部門向けのロードマップを作成！
「ノーコード/ローコード活用チャート」

ここまでのレベルに
育って欲しい



自ら進められる

気になりますね？

他の教育資料と何が違うの？

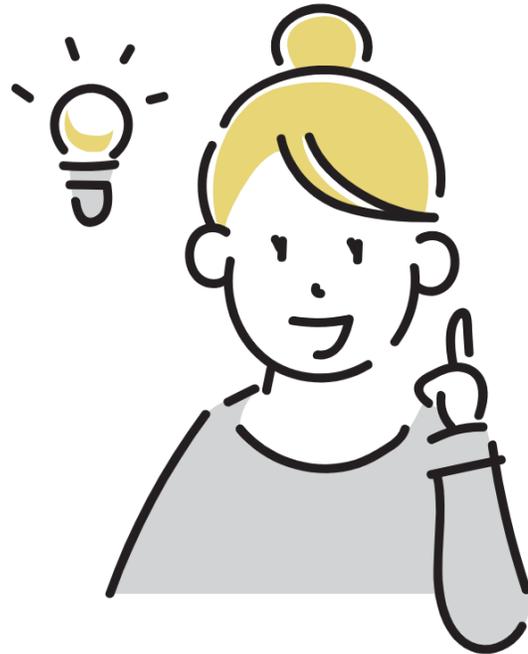
マニュアル読めばいいのでは？

どうせ読んでくれないでしょ



興味を持たせる工夫

第一印象は3秒で決まる



開発戦士

アンダラレンジャー

第1話 目指せユーザー開発のヒーローズ



ノーコード／ローコード時代のシステム開発 分科会

いかがだったでしょうか

なんだこれ！

変わった資料！

あのヒーロー資料見た？



楽しく進められる工夫

はじめに (1/2)

はじめに

プログラミングせずにシステム開発できるノーコード/ローコードツール※は、使いこなすことができればユーザー部門（非IT部門）が主体となって業務課題を解決できるようになる魔法のようなツールになります。

しかしながら、部門内で内製化を成熟させるためには、単にツールの使用方法を理解するだけでなく、業務分析や要件定義といったシステム開発を取り巻く多くの要素を知る必要があります。

※ ローコードツールは一部の機能実装にプログラミングが必要な場合があります。

ツールを触ってみたいけど
何から始めればいいのか？

開発で気を付けなきゃいけない
ポイントって…？



ノーコード/ローコード時代のシステム開発 分科会

3

ユーザー開発のヒーローを
一緒に目指しましょう！

ノーコード/ローコード時代のシステム開発 分科会

はじめに (2/2)



非IT部門でも業務アプリ
が開発できた！！



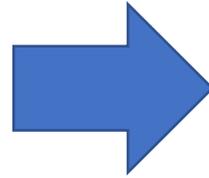
4

メインのフレームワーク

小さな目標を設定



視覚的に進捗を示す

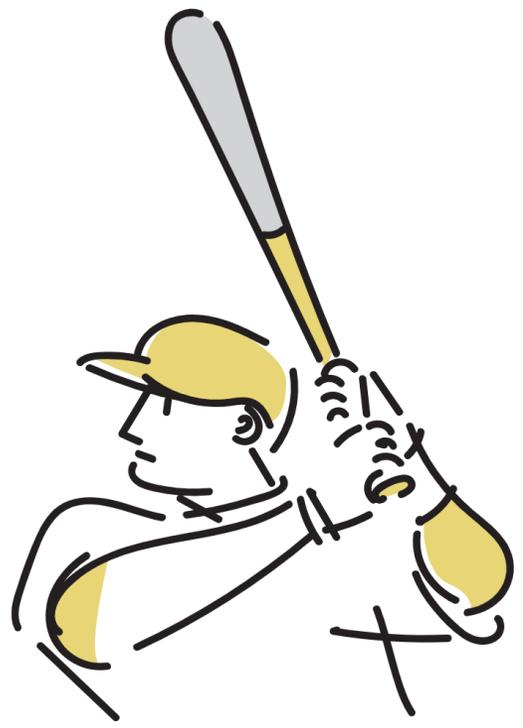


達成感



マンダラチャートとは？

達成したい目標



体のケア	サプリメントをのむ	FSQ 90kg	インステップ改善	体幹強化	軸をぶらさない	角度を付ける	上からボールを叩く	リストの強化
		FSQ 100kg	リリースポイントの安定	コントロール	不安をなくす	力まない	キレ	下半身主導
スイング	可動域	夜7杯 朝3杯	下肢の強化	体を開かない	メンタルコントロールをする	ボールを前でリリース	回転数アップ	可動域
はっきりとした目標目的をもつ	一喜一憂しない	頭は冷静に 心は熱く	体づくり	コントロール	キレ	軸で回る	下肢の強化	体重増加
ピンチに強い	メンタル	雰囲気流されない	メンタル	ドラ1 8球団	スピード 160km/h	体幹強化	スピード 160km/h	肩周りの強化
波をつくらない	勝利への執念	仲間を思いやる心	人間性	運	変化球	可動域	ライナーキャッチボール	ピッチングを増やす
感性	愛される人間	計画性	あいさつ	ゴミ拾い	部屋掃除	カウントボールを増やす	フォーク完成	スライダーのキレ
思いやり	人間性	感謝	道具を大切に使う	運	審判さんへの態度	遅く落差のあるカーブ	変化球	左打者への決め球
礼儀	信頼される人間	継続力	プラス思考	応援される人間になる	本を読む	ストレートと同じフォームで投げる	ストレートからボールに投げるコントロール	奥行をイメージ

ヒーロー作成チャート

ノーコード/ローコードツールの概要理解	開発手法の理解	公式ドキュメントの閲覧	要件定義の理解	業務プロセスの整理	ボトルネックや課題の特定	必要なデータ項目の整理	コスト・工数の概算見積	基本設計書の作成 (システム構成、画面、フロー、データ)
アプリ完成イメージの構築	基礎知識の習得	公式チュートリアルの実施	ステークホルダーの分析	業務プロセスの整理	業務改善の優先順位設定	動作テストの実施	小規模なアプリの開発	バージョン管理の理解
チーム内での操作共有セッション	活用/成功事例の調査	ツール基本操作の体験	業務フロー図の作成	開発可能な業務の特定	業務改善アイデアの洗い出し	ユーザーインターフェースの自己評価	実践を通じた開発ツールの操作体験	実務に関連した簡単なアプリの作成
作成したアプリの試験運用	開発したアプリの効果測定 (例: 時間短縮、ミス軽減)	ユーザーフィードバックの収集	基礎知識の習得	業務プロセスの整理	小規模なアプリの開発	メンターの設置	チャットツールや定例会で進捗や課題を共有	定期的な技術勉強会の開催
表彰制度の設置	フィードバックの収集と改善	改善点を反映したアプリのアップデート	フィードバックの収集と改善	ユーザー開発のヒーローズ	コミュニティの形成	成功事例を社内公开发表 (モチベーション向上)	コミュニティの形成	メンターとの個別フィードバックセッション
レポートや社内プレゼンによる全社共有の実施	他部門へのノウハウ展開	継続的な改善サイクルの確立	サポート体制の整備	自走力の強化	部門の統制力強化	社内ハッカソンを開催し、部門を越えた交流を促進	知識やノウハウを共有する場の設立	外部セミナー・カンファレンスへの参加
N/L開発の社内有識者をサポート担当者に設置	サポートリーダーの選出	ツールを活用したコミュニケーションの整備	自主的な学習時間の確保	学習環境の整備	外部オンライン学習の活用 (Udemy, YouTube等)	アプリの管理者を任命	アプリ作成フローを策定 (開発者・管理者)	アプリの管理簿作成
FAQ・ナレッジベースの作成	サポート体制の整備	共通言語の確立	応用機能に挑戦 (条件分岐、自動化フロー、動的UI/UX等)	自走力の強化	開発したアプリの改善点をリストアップ	アプリの使用法や仕様の文書化	部門の統制力強化	アプリの雛形作成
社内トレーニングの定期実施	開発ツールに関する問い合わせ先の明確化	社内相談窓口の設立	実務に活用できる高度なロジックを実装	ツール内での外部データやAPI連携の学習	新機能やトレンドのキャッチアップ	アプリの拡張機能の制限	アプリの利用範囲の制限	アプリの作成権限の制限

ヒーロー作成チャート



百聞は一見にしかず
小規模な開発から始める
設計開発テストの基本的な流れを知る

小規模な
アプリの開発

定義の理解	業務プロセス	ボトルネックや課題の特定	要件定義	要件・工数	基本設計書の作成 (システム構成、画面、フロー、データ)
ワークホル	業務改善の	業務改善の	動作テストの	小規模な	バージョン
の分析	先順位設定	先順位設定	実施	アプリの開発	管理の理解
プロ	業務改善	業務改善	ユーザーイン	実践を通じた	実務に関連し
の作成	アイデアの	アイデアの	ターフェース	開発ツールの	た簡単な
	洗い出し	洗い出し	の自己評価	操作体験	アプリの作成
知識の	業務プロセス	業務プロセス	メンターの	チャットツール	定期的な技術
習得	の整理	の整理	設置	や定例会で進捗	勉強会の開催
				や課題を共有	
ードバ	ユーザー開発	コミュニティ	成功事例を	コミュニティ	メンターとの個
の収集			社内で発表		別セッション

ヒーロー作成チャート



必要なデータ 項目の整理	コスト・工数 の概算見積	基本設計書の 作成 (システム構成、 画面、フロー、 データ)
動作テストの 実施	小規模な アプリの開発	バージョン 管理の理解
ユーザーイン ターフェース の自己評価	実践を通じた 開発ツールの 操作体験	実務に関連し た簡単な アプリの作成

「小規模なアプリ開発」
が出来るとなるためにはどんな事が必要か？

ユーザー開発

コミュニティ

コミュニティ

メンターとの個

ヒーロー作成チャート

使いやすいアプリなのか
効果があるアプリなのか
継続的な改善サイクルの確立



フィードバックの収集と改善

作成したアプリの試験運用

開発したアプリの効果測定
(例: 時間短縮、ミス軽減)

ユーザーフィードバックの収集

表彰制度の設置

フィードバックの収集と改善

改善点を反映したアプリのアップデート

レポートや社内プレゼンによる全社共有の実施

他部門へのノウハウ展開

継続的な改善サイクルの確立

サポート体制の整備

自走力の強化

部門統制力

基礎知識の習得

小規模アプリの

業務フローの作成

業務改善アイディア洗い出し

ステークホルダーの分析

業務プロセスの整理

業務改善優先順位

ヒーロー作成チャート

「フィードバックの収集と改善」
が出来るようになるためにはどんな事が必要か？



項目だけじゃわからない！

補足資料はあるの？

学ぶメリットは？
学ばないデメリットは？
具体的な解説を！



業務フロー図

ある業務において“**誰が、いつ、何を、どのように**”するのかを図式化したもの。業務の全体像を客観的に把握することができ、後続の開発フェーズにおいて有用な指標となる。

この項目を学習/実践することで…

- 👍 関係者間での共通認識が形成しやすくなる。
- 👍 業務の問題点や改善点の早期発見に繋がる。
- 👍 アプリケーション (システム) に落とし込む範囲が特定できる。
- 👍 開発が必要な機能やテストケースの洗い出しがしやすくなる。



業務フロー図

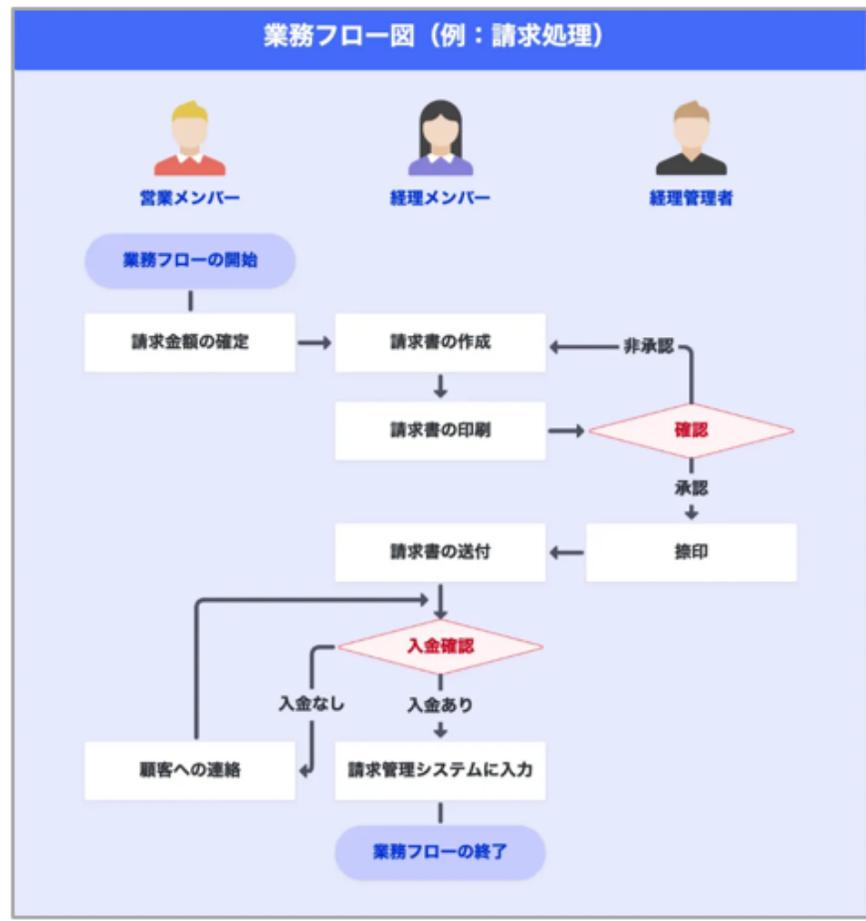
この項目を学習/実践しないと…

- 👎 どの業務範囲をアプリケーションとして実装したいのかわからない…
- 👎 実装したい機能の抜け漏れが発生してしまう…
- 👎 アプリケーションの開発計画を部門内に展開する際、説明が難しい…



業務フロー図

って具体的にどういうもの？



<フローで用いる基本的な記号>

開始・終了 (端子)		業務フローの開始・終了を表す
プロセス (処理)		工程・処理を表す
判断 (条件分岐)		Yes/No条件等でフローが変化する場合に使用
矢印		処理や条件分岐の間をつなぐ記号

参考：
業務フロー図の書き方を0から丁寧に解説 記号や便利ツールも紹介【2024年11月24日】
https://product.strap.app/magazine/post/knowhow_workflow

部門の統制力強化

をするためには何を決める必要がある？

アプリケーションの権限の制限

権限設定機能のアクセス制限し、適切なスキーマを持つメンバーの開発を許可

アプリケーションの利用範囲の制限

アプリケーションが利用可能な部門、ユーザー、データアクセス権限を明確に定義し、不正利用を防止する。

アプリケーションの拡張機能の制限

サードパーティ製プラグインや高度な機能の使用を制限し、セキュリティリスクと機能の複雑性の増大を防止する。

データのプライバシーの保護や、コンプライアンス遵守を目的とする。

システムの安定性、メンテナンスの容易性を確保する。

動作テスト

部門の統制力強化

テストと呼ばれる、アプリケーションの作り手側目線のテストを行う。既定業務フロー図や処理パターンを基に、網羅的にテストケースを作成する

条件を満たすかどうか(○か×か)で処理が変わるシステムの場合、テストを行い正しい処理がなされるかを確認する。

No.	条件A	条件B
1	○	○
2	○	○
3	○	○
4	○	○
5	×	×
6	×	×
7	×	×
8	×	×

共通言語の確立

に必要な5つのステップとは？

- 共通言語確立の必要性をプロジェクトメンバーに説明し、合意を得る。
- 関連部門の用語集や文章の収集を行う。
- 収集したものから、開発対象の業務フロー内に登場する用語をリストアップする。
- リストアップした用語の意味を再定義し、共通言語としてまとめる。
- 共通言語が定義された用語集やガイドラインをプロジェクトメンバーへ展開する。

説明しよう！

マンダラレンジャーは何故生まれたのか！？

近い未来の断崖絶壁、その名も「2025年の崖」「2030年問題」！IT人材が足りない！そんな危機に立ち向かうため、今、最強の開発戦士「マンダラレンジャー」が誕生した！

ノーコード/ローコードツールとマンダラチャートという切り札を手に、非IT部門を「ユーザー開発のヒーローズ」へと変身させる！

マンダラチャートとは！？

その正体は戦略を視覚化する究極の秘密兵器！中央に最終目標を据え、周囲の8つのエリアに最終目標達成に必要な要素を、さらにその要素を満たすための具体的な行動計画を、周囲の8つのエリアにそれぞれ配置する！そしてヒーローの必殺技を組み立てるが如く、複雑な戦略を一目で理解できる驚異のダイアグラム！

きみもユーザー開発のヒーローズの一員になろう！

はじめ

ツールを触ってみたいけど何から始めればいいのか？

開発で気を付けたいポイントって…？

実践することで…

未経験でも実現可能なことが理解できる。

開発のモチベーションアップに繋がる。

ノーコード/ローコードツールの特徴を掴むことで、開発の利点を活かした設計が可能となる。

ノーコード/ローコードツールの概要理解

ノーコードツールとは何か？

「何ができるか」を学ぶ。

ノーコード/ローコード時代のシステム開発

全62ページの大ボリューム

マンダラチャートの使い方

ノーコード/ローコードツールの概要理解	開発手法の	公式ドキュメ	要件定義	業務プロセスの整理	ボトルネックや課題の特定	必要なデータ項目の整理	コスト・工数の概算見積	基本設計書の作成 (システム構成、画面、フロー、データ)
アプリ完成イメージの構築				業務プロセスの整理	業務改善の優先順位設定	動作テストの実施	小規模なアプリの開発	バージョン管理の理解
チーム内での操作共有セッション				開発可能な業務の特定	業務改善アイデアの洗い出し	ユーザーインターフェースの自己評価	実践を通じた開発ツールの操作体験	実務に関連した簡単なアプリの作成
	開発したアプリ	ユーザーフィードバックの収集				メンターの設置	チャットツールや定例会で進捗や課題を共有	定期的な技術勉強会の開催
		改善点を反映したアプリのアップデート				成功事例を社内で発表 (モチベーション向上)	コミュニティの形成	メンターとの個別フィードバックセッション
	よる全社共有の	ノウハウ展開				社内ハッカソンを開催し、部門を越えた交流を促進	知識やノウハウを共有する場の設立	外部セミナー・カンファレンスへの参加
N/L開発の社内	サポート	ツールを活用し		学習環境の整備	外部オンライン学習の活用 (Udemy, YouTube等)	アプリの管理者を任命	アプリ作成フローを策定 (開発者・管理者)	アプリの管理簿作成
FAQ・ナレッジベースの作成				自走力の強化	開発したアプリの改善点をリストアップ	アプリの使用方法や仕様の文書化	部門の統制力強化	アプリの雛形作成
社内トレーニングの定期実施				ツール内での外部データやAPI連携の学習	新機能やトレンドのキャッチアップ	アプリの拡張機能の制限	アプリの利用範囲の制限	アプリの作成権限の制限

現状確認

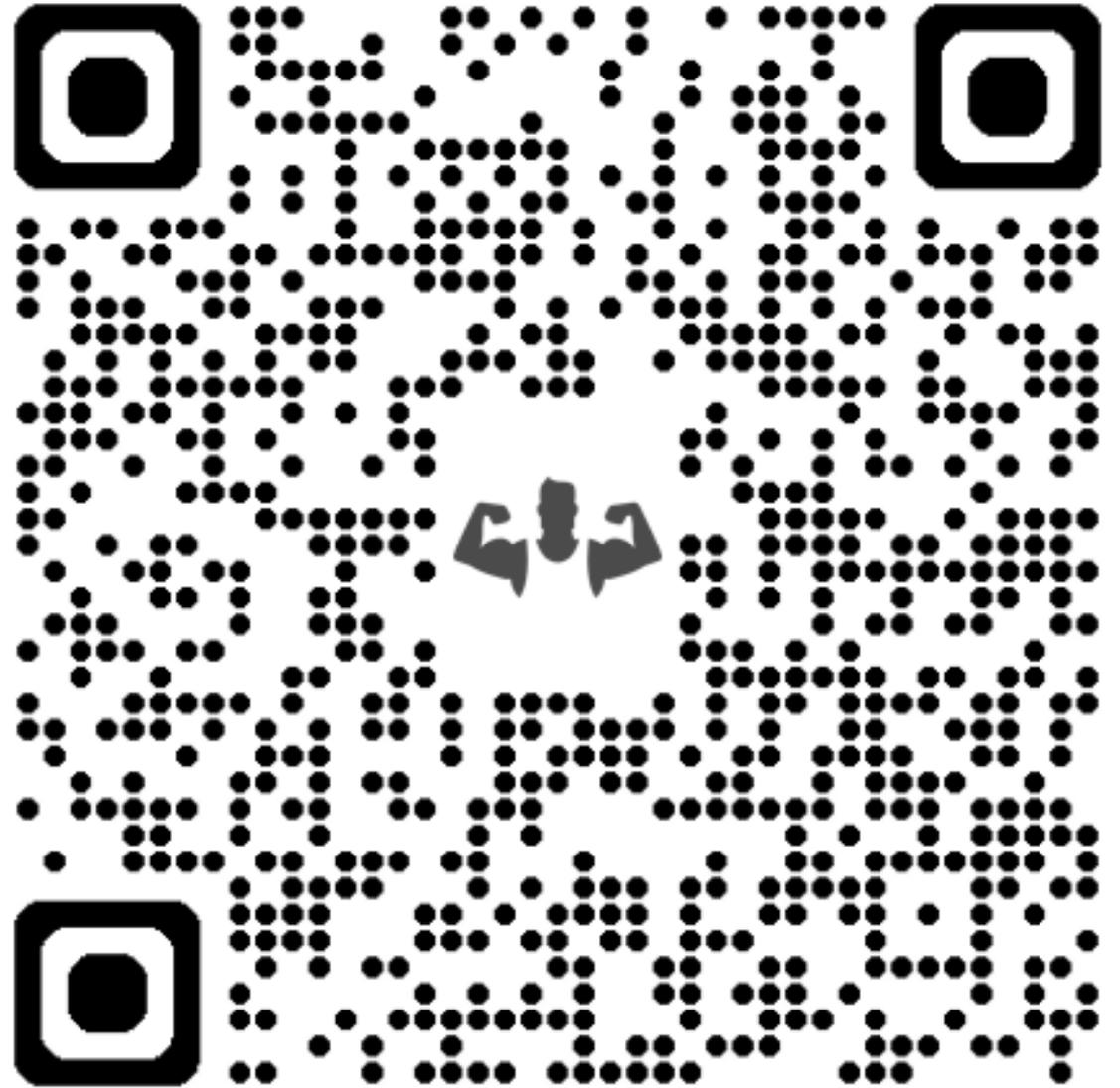
完了チェック

項目決定

取り組み

ぜひご覧ください

約4MB、ご注意ください



目次

ノーコード/ローコードの必要性

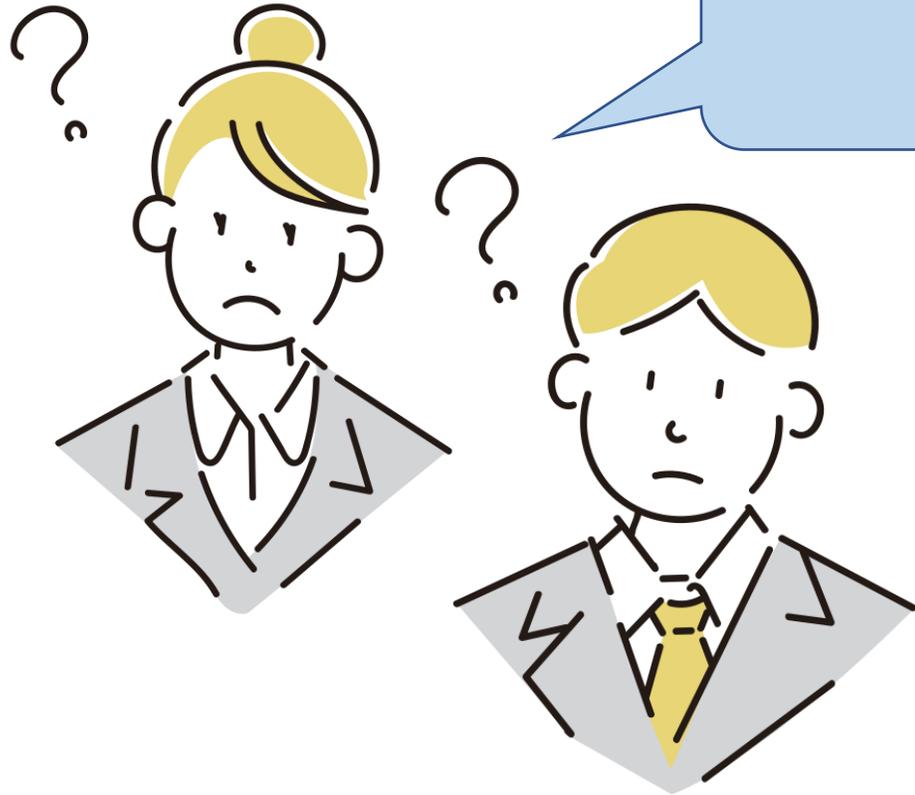
- ✓ 不足するIT人材
- ✓ ノーコード/ローコードツールという打ち手

当分科会の研究と提案

- ✓ ノーコード/ローコード活用チャートの提案
- ノーコード/ローコード活用チャートの有効性

まとめ

有用性について



役立つ資料なの？

IT知識が少なくても
理解できるの？

アンケートを実施

「ノーコード/ローコード活用チャート」 の有用性を確認

- ・活用チャートを読んで頂きアンケート形式で評価
- ・回答者の属性「IT部門」または「非IT部門」
- ・「IT部門」と「非IT部門」で別のアンケート
- ・回答総数31件

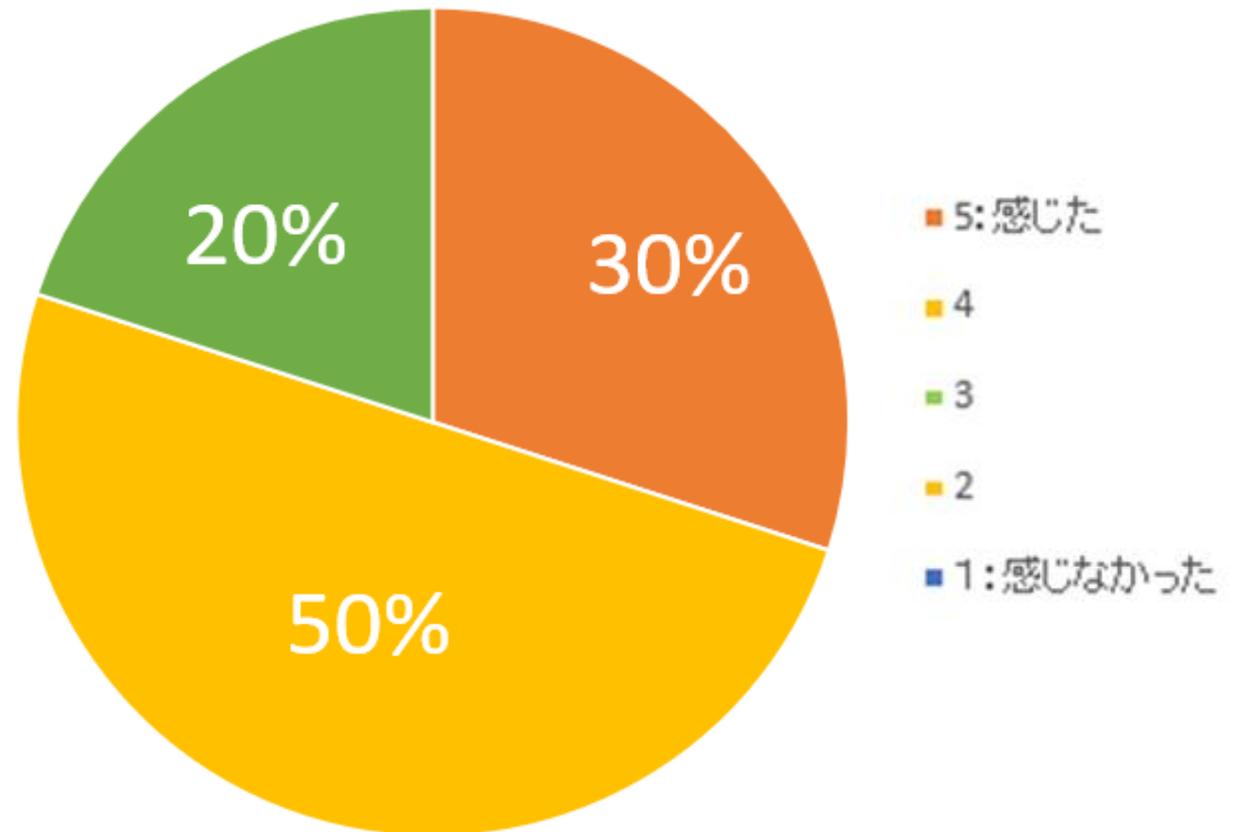


IT部門からの回答

約8割の方が
有用と回答

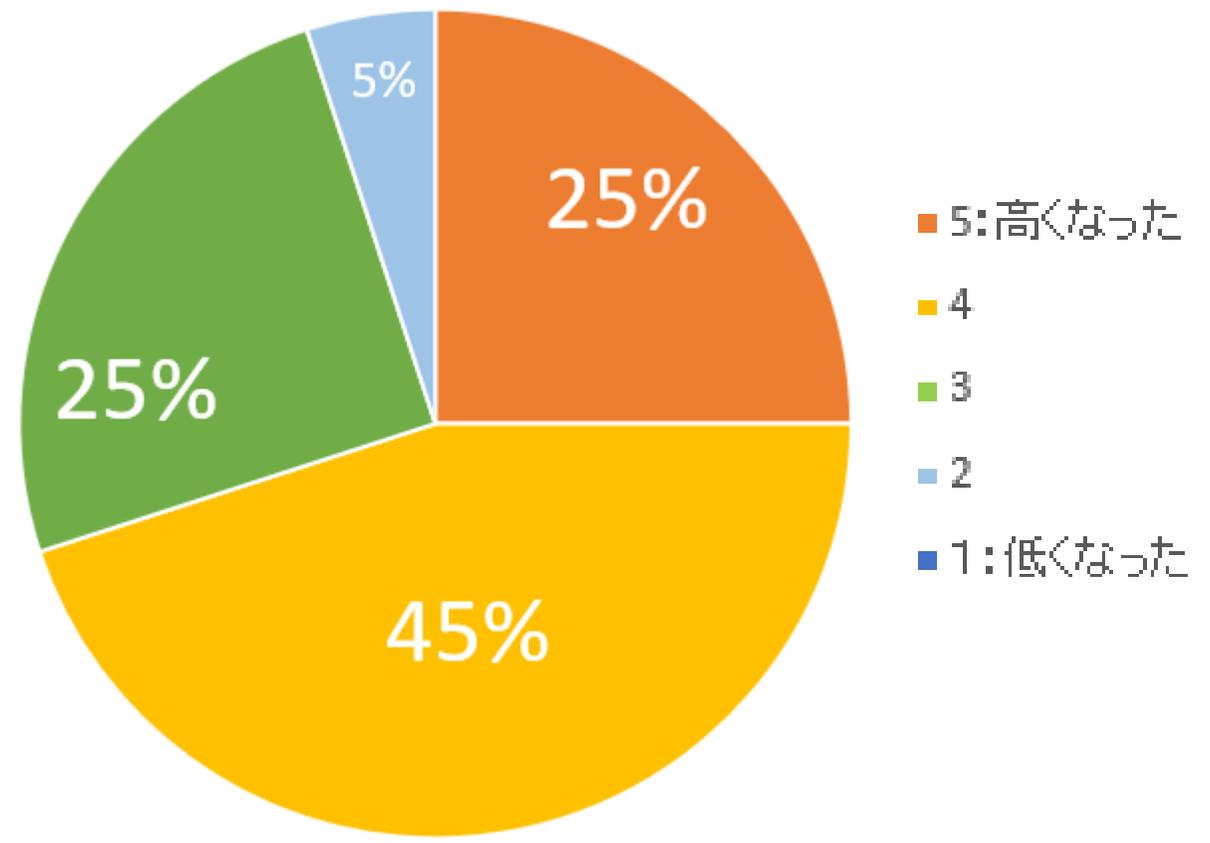


添付の資料は非IT部門にとって
有用な資料であると感じましたか



非IT部門からの回答

ノーコード/ローコードツールを活用した
開発への興味が変わりましたか。



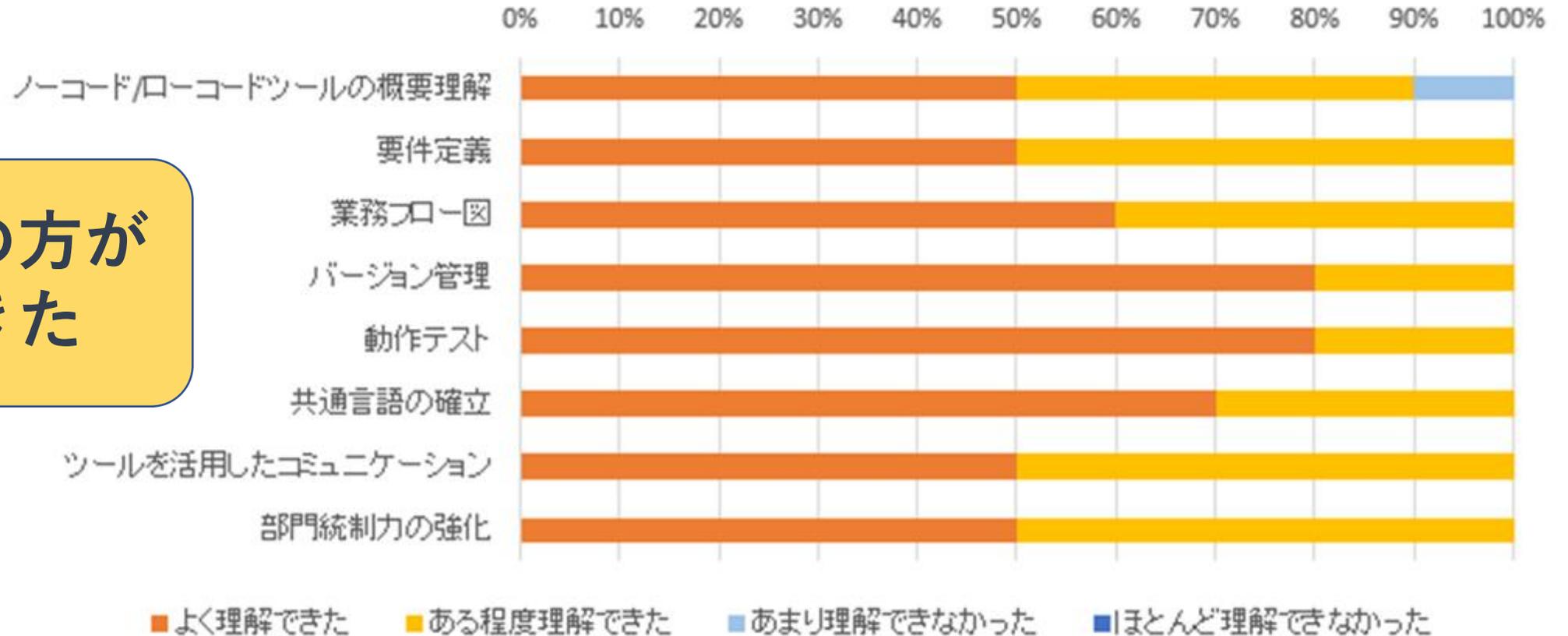
約7割の方が
興味UPと回答



IT部門からの回答

資料内の各解説ページ(資料4～29 p)の内容について、
どの程度理解できましたか

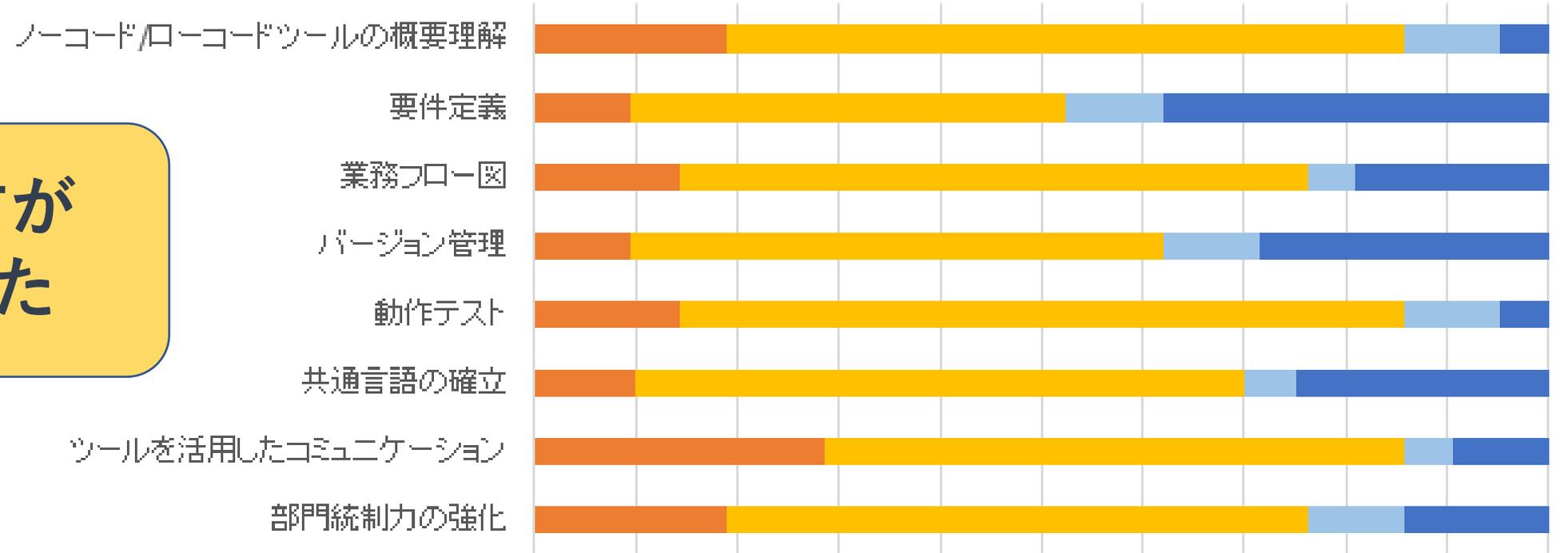
ほぼ**全て**の方が
理解できた



非IT部門からの回答

資料内の各解説ページ(資料4~29 p)の内容について、
どの程度理解できましたか

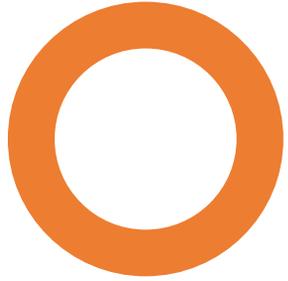
0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%



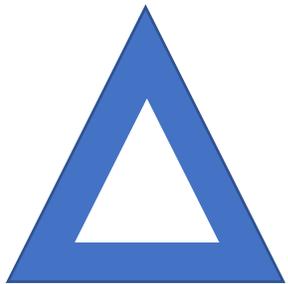
約7割の方が
理解できた

■よく理解できた ■ある程度理解できた ■あまり理解できなかった ■ほとんど理解できなかった

コメント抜粋



- ・ 視覚的で読みやすい
- ・ 導入直後の初期資料に良さそう
- ・ 非IT部門に向けていそう



- ・ 失敗時の具体例を交えてほしい
- ・ 多くの項目の解説ページがほしい
- ・ 資料の使い方や作り手の意図がほしい

さらなる改善を実施

各項目の重要性を追記

実施しないとどうなる？を追記

チャートの使い方を明記

付録資料を追加

Ver 1.0

Ver 1.1

Ver 1.2

to be continued

重要なポイント

各社でヒーローは違う

自由にカスタムして下さい



to be continued...

目次

ノーコード/ローコードの必要性

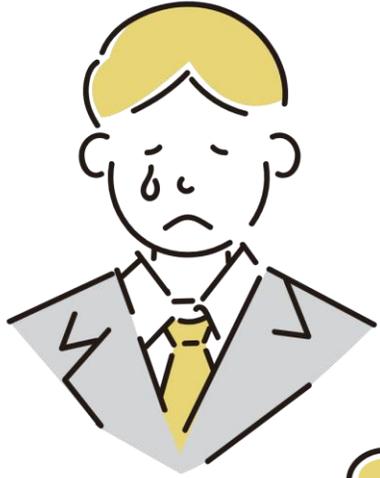
- ✓ 不足するIT人材
- ✓ ノーコード/ローコードツールという打ち手

当分科会の研究と提案

- ✓ ノーコード/ローコード活用チャートの提案
- ✓ ノーコード/ローコード活用チャートの有効性

○ まとめ

まとめ

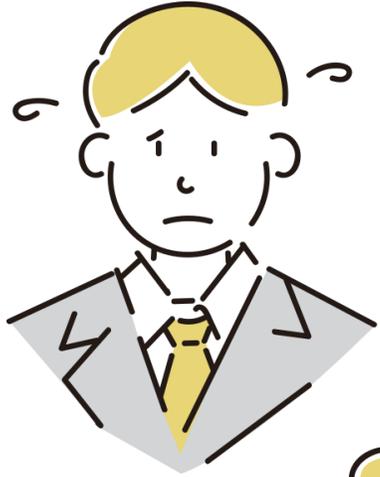


人口・IT人材は減っていく



業務は減らない

まとめ

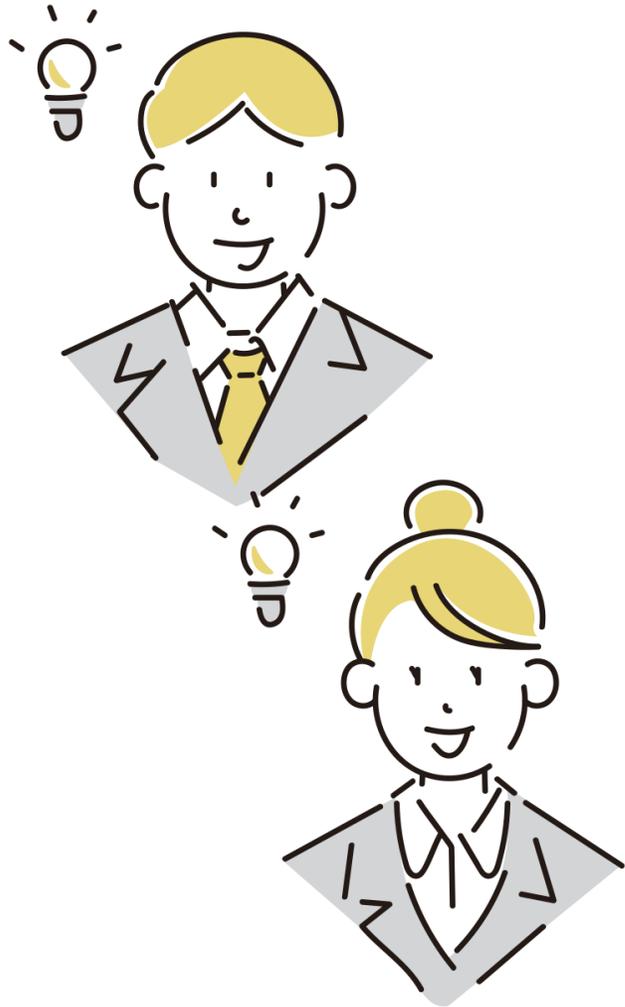


生成AI、DX化、自動化…



新技術の生涯学習

まとめ

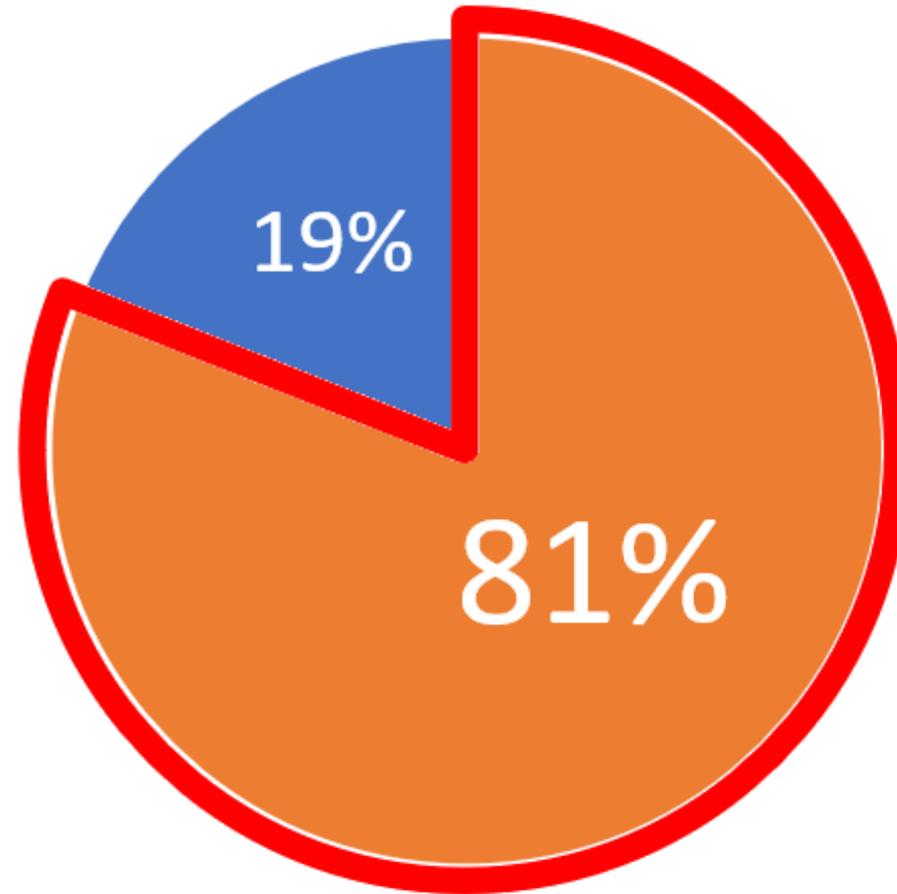


ノーコード/ローコードなど
ITツールをもっと活用しよう

非IT部門に
ヒーローになって貰おう

非IT部門のやる気！

あなたはノーコード/ローコードツールについて
学ぶ機会があれば参加してみたいですか？



- 参加したい
- 参加したくない



ヒーロー
68万人

俺たちの
戦いは
これからだ！



ノーコード/ローコード時代のシステム開発分科会