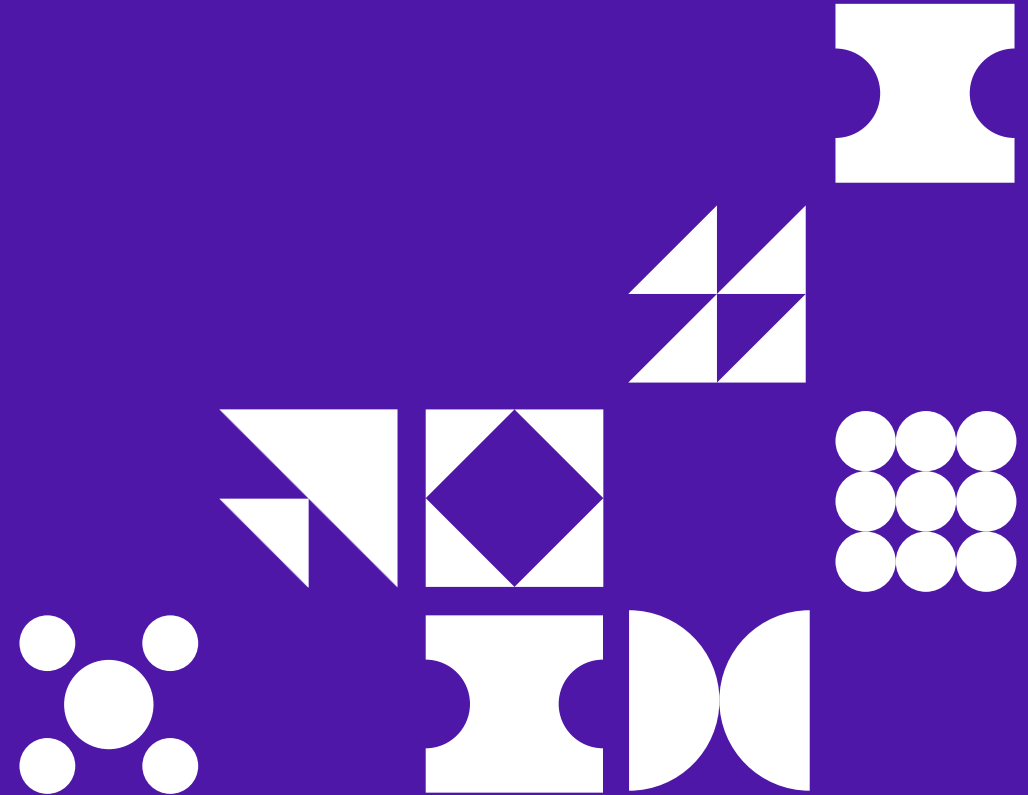


# 2020年度 「アジャイルプロジェクトの実態」 に関するアンケート

- ◆ データ編
- ◆ 分析と提言編



アジャイル研究会・PMI日本支部事務局

# はじめに

本資料は、2020年3月2日 ～ 2020年3月16日にPMI日本支部アジャイル研究会が実施したアンケート調査「2020年度アジャイルプロジェクトの実態」の結果です。

PMI日本フォーラム2020のC- 4 セッション「2020年度アジャイルプロジェクトの実態に関するアンケート-データ編、分析・提言編-」として、2020年7月4日～12日にオンデマンド配信した内容を公開用に加筆修正したものです。

日本国内におけるアジャイル推進の参考資料となるよう、一般に公開いたします。

調査にご協力いただいた皆さま、ありがとうございました。

2020年9月

一般社団法人PMI日本支部

アジャイル研究会



# 過去の報告

過去の報告は下記にて公開しております。合わせてご覧ください。

【アジャイルPM研究会】2015年度  
アジャイル プロジェクト マネジメント 意識調査報告

[https://www.pmi-japan.org/topics/lcop/agilepm\\_20151125.php](https://www.pmi-japan.org/topics/lcop/agilepm_20151125.php)

【アジャイルPM研究会】2016年度  
アジャイル プロジェクト マネジメント 意識調査報告

[https://www.pmi-japan.org/topics/lcop/agilepm\\_20151101.php](https://www.pmi-japan.org/topics/lcop/agilepm_20151101.php)

【アジャイルPM研究会】2017年度  
アジャイル プロジェクト マネジメント 意識調査報告

[https://www.pmi-japan.org/topics/lcop/agilepm\\_20171204.php](https://www.pmi-japan.org/topics/lcop/agilepm_20171204.php)

【アジャイルPM研究会】2018年度  
アジャイル プロジェクト マネジメント 意識調査報告

[https://www.pmi-japan.org/topics/lcop/agilepm\\_20181003.php](https://www.pmi-japan.org/topics/lcop/agilepm_20181003.php)

【アジャイルPM研究会】2019年度  
アジャイルプロジェクトマネジメント意識調査報告

[https://www.pmi-japan.org/topics/lcop/agilepm\\_20191106.php](https://www.pmi-japan.org/topics/lcop/agilepm_20191106.php)

# 目次

## <データ編>

- 調査概要
- 回答者の属性
- 関連資格等の状況
- 組織のプロジェクトへの取り組み
- 回答者のアジャイルへの認識
- 回答者の経験したアジャイル
- 親しい友人や同僚に薦める可能性はどのくらいありますか？(ネット・プロモーター・スコア)

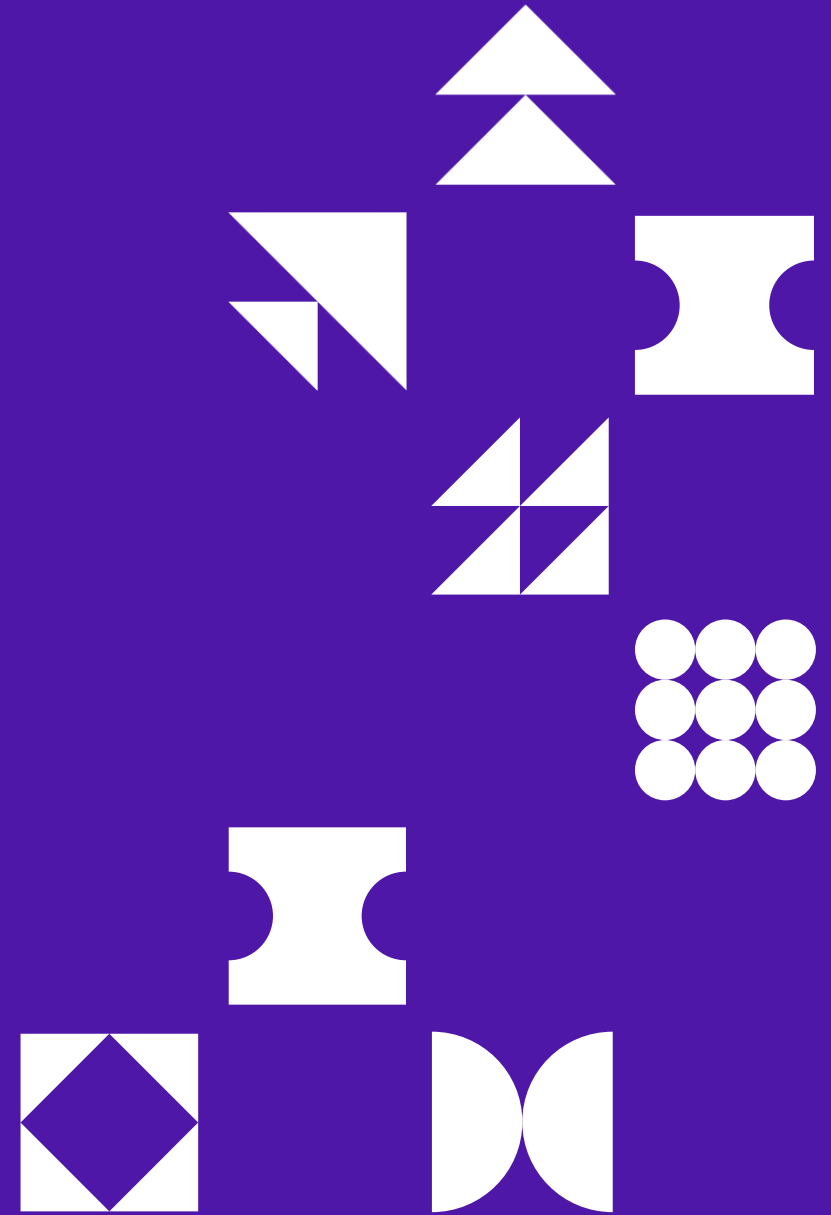
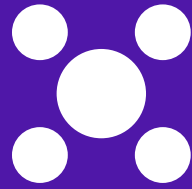
## <分析と提言編>

- アジャイルはなぜ回答者に推奨されなかったのか？
- アンケートから見えた課題
- 多様性のあるアプローチの実現を目指す  
：ディシプリンド・アジャイル・デリバリー（DAD）
- ライフサイクル
- プロセスの目標と実践/戦略
- まとめ



2020年

# アジャイル プロジェクト マネジメント 意識調査報告 <データ編>



## <データ編>

- ▶ 調査概要
- ▶ 回答者の属性
- ▶ 関連資格等の状況
- ▶ 組織のプロジェクトへの取組み
- ▶ 回答者のアジャイルへの認識
- ▶ 回答者の経験したアジャイル
- ▶ 親しい友人や同僚に薦める可能性はどのくらいありますか？  
(ネット・プロモーター・スコア)

分析と提言編 へ続く

# 調査概要

- 目的
  - アジャイル プロジェクト マネジメントの導入／展開を推進するための具体策を研究会で検討し、広く情報発信していく。
- 調査方法
  - Webアンケートによる調査
- 調査期間
  - 2020年3月2日 - 2020年3月16日
- 調査対象
  - 有効回答数 266名

2020年

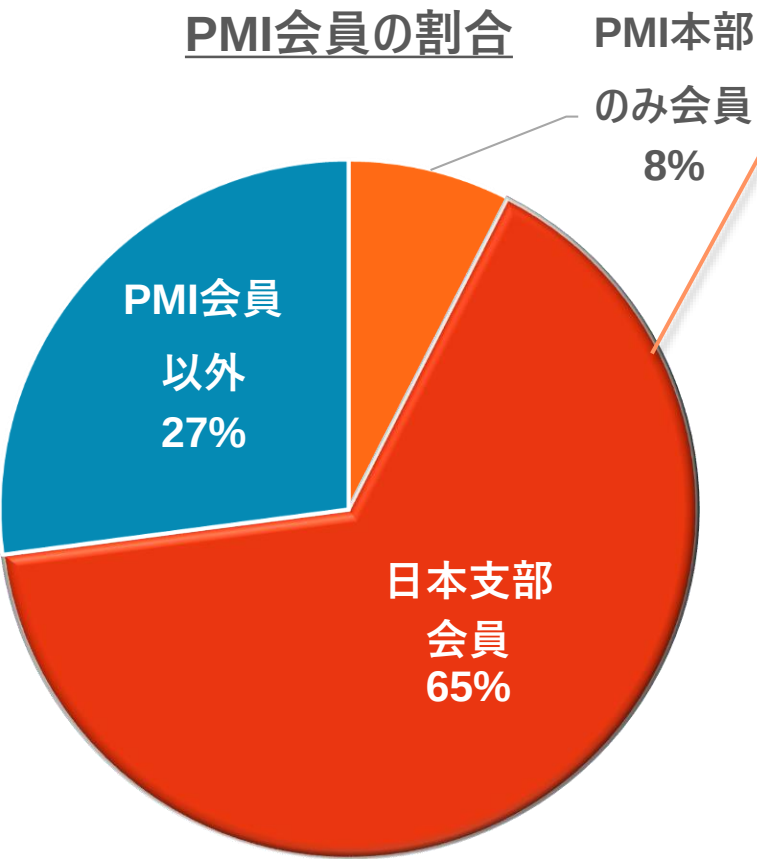
アジャイル プロジェクト マネジメント意識調査報告＜データ編＞

# 回答者の属性

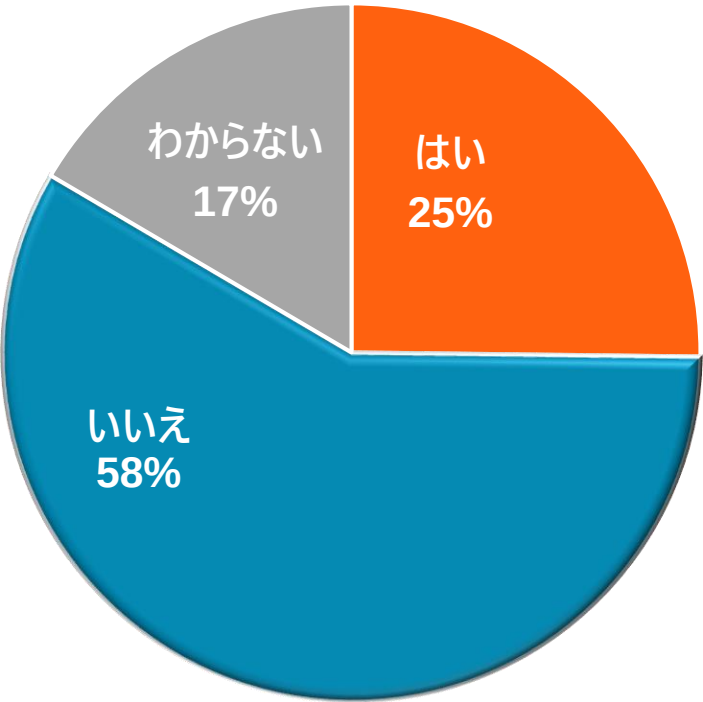
# 回答者の属性 | P M I との関係

回答者の7割以上がPMI会員

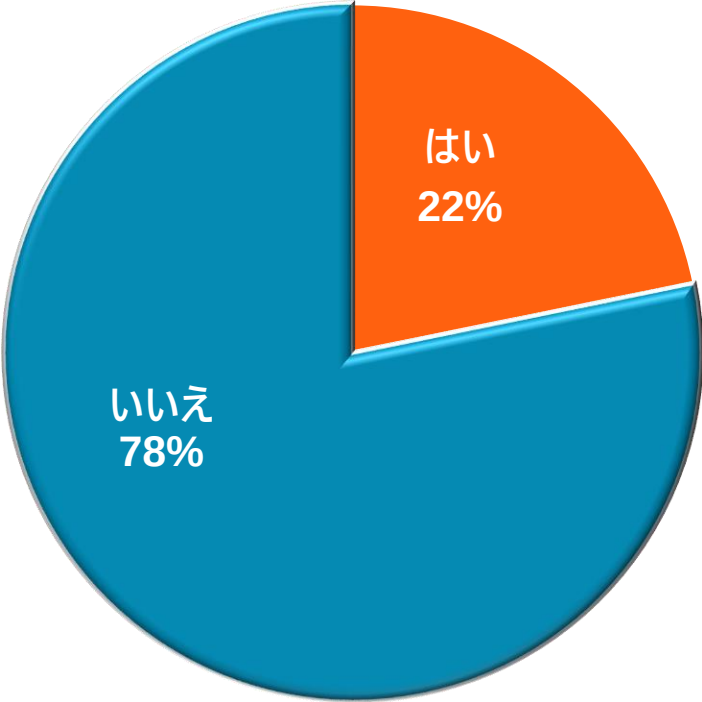
PMI会員の割合



所属組織が法人スポンサー



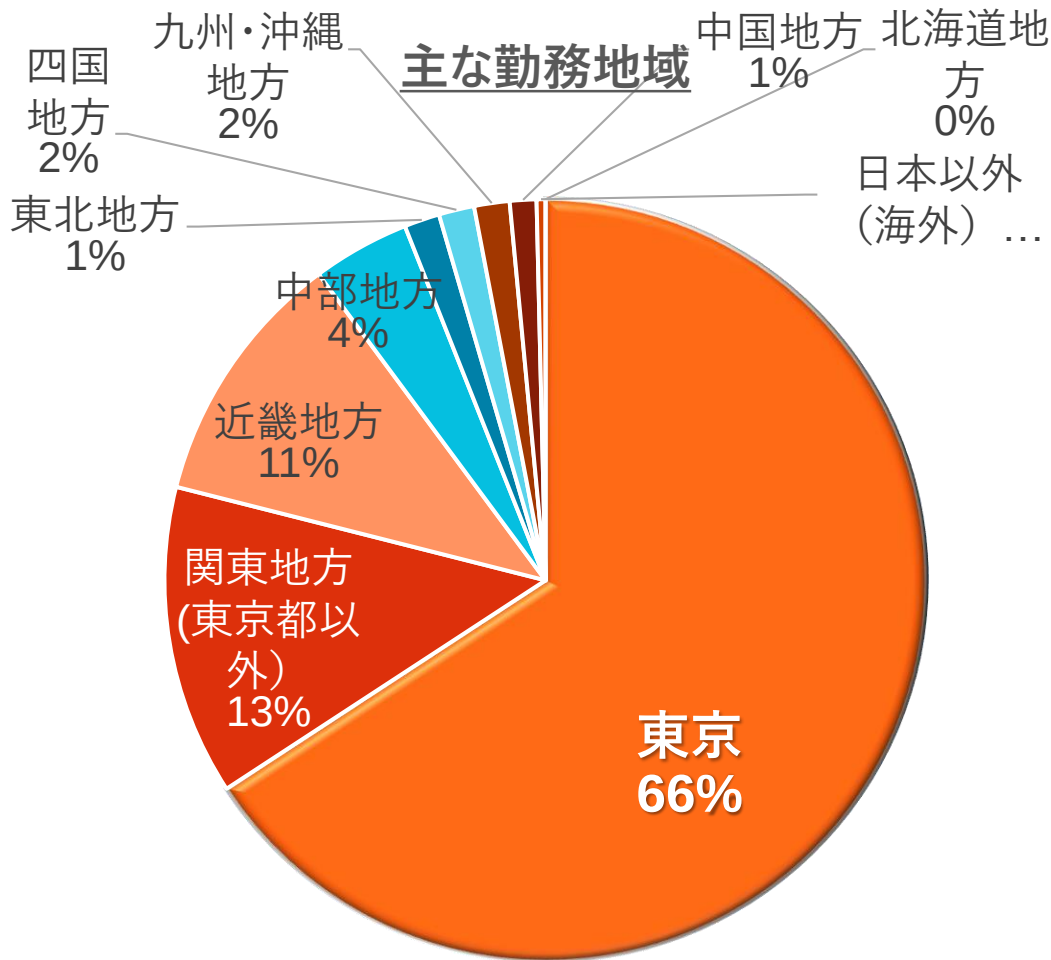
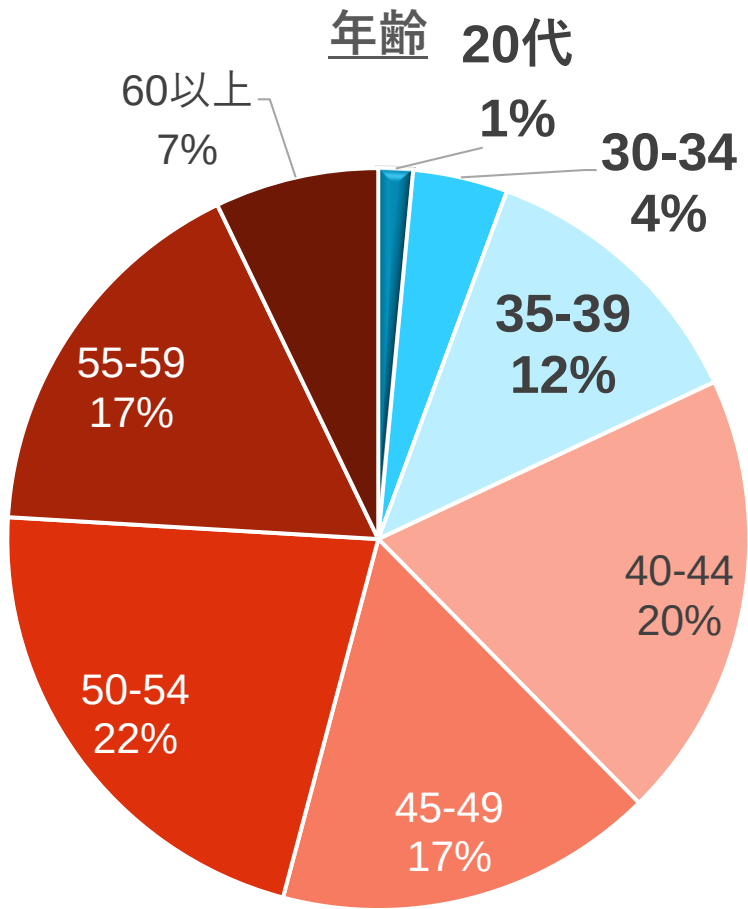
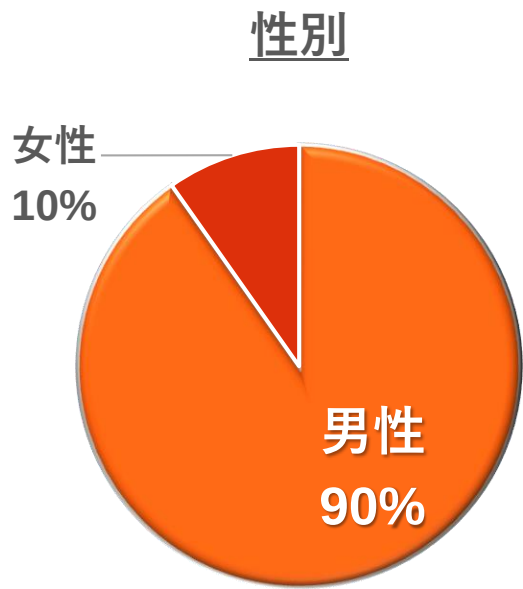
過去にこのアンケートに回答



n=266

# 回答者の属性 | 性別・年齢と勤務地

20~30代あわせて17%  
40代37%、50代39%が中心

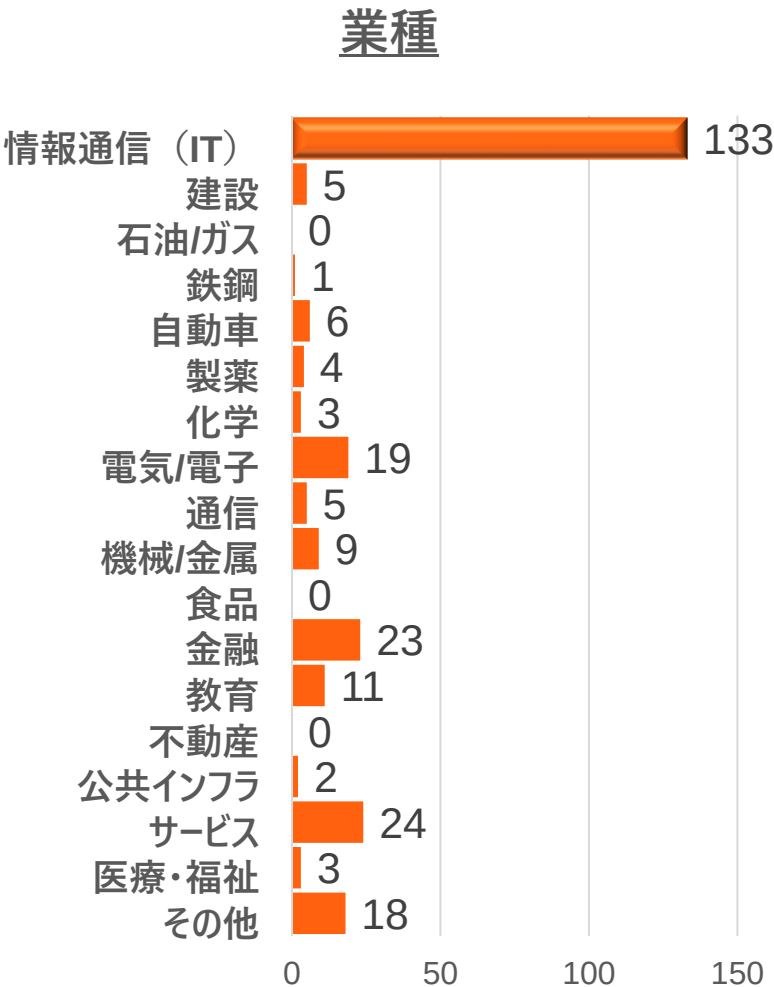
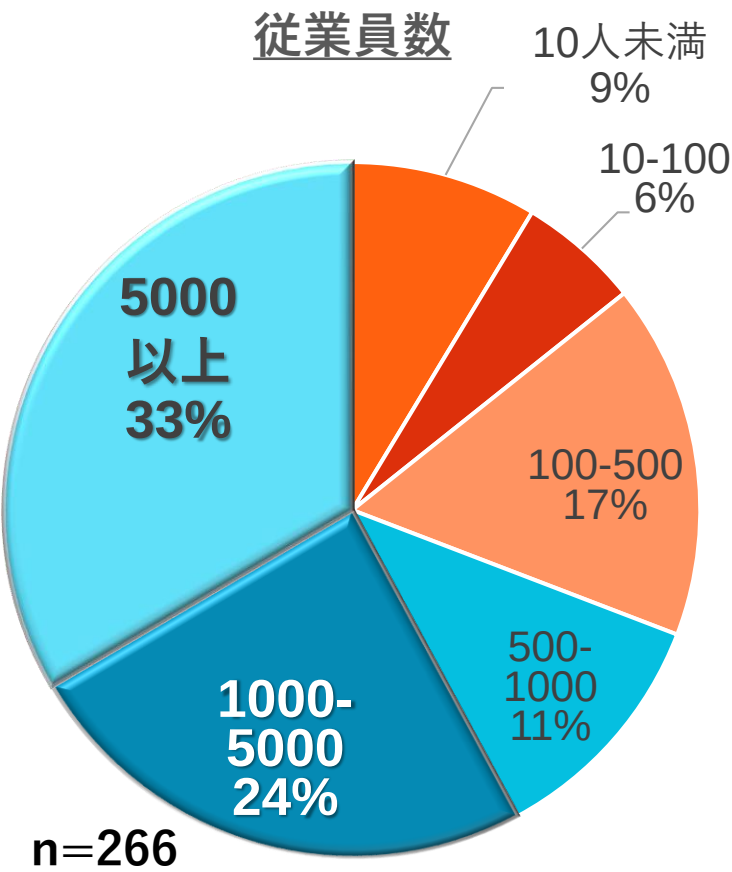


n=266

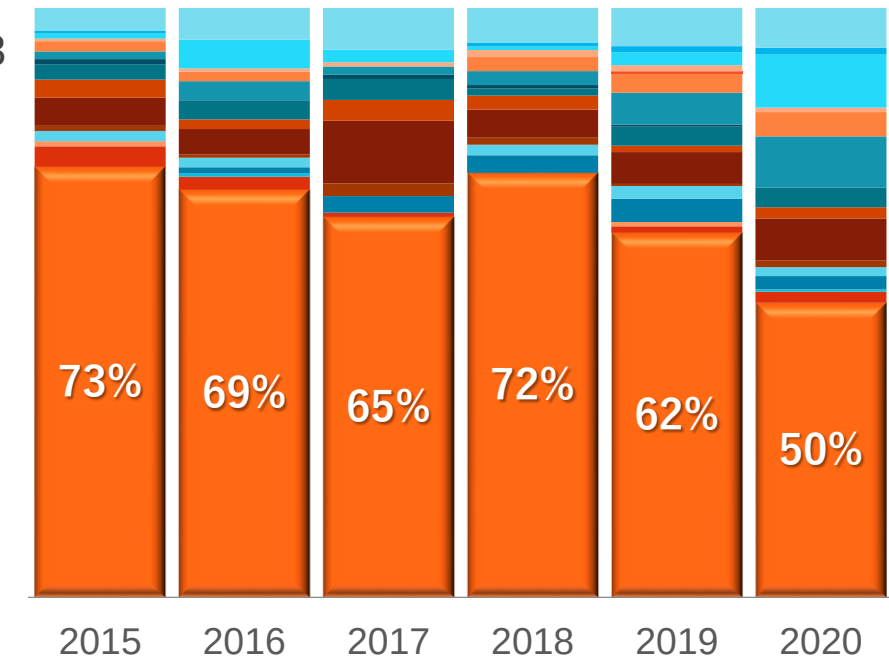


# 回答者の属性 | 所属企業の従業員数と業種

半数以上が1,000人以上の大企業に所属



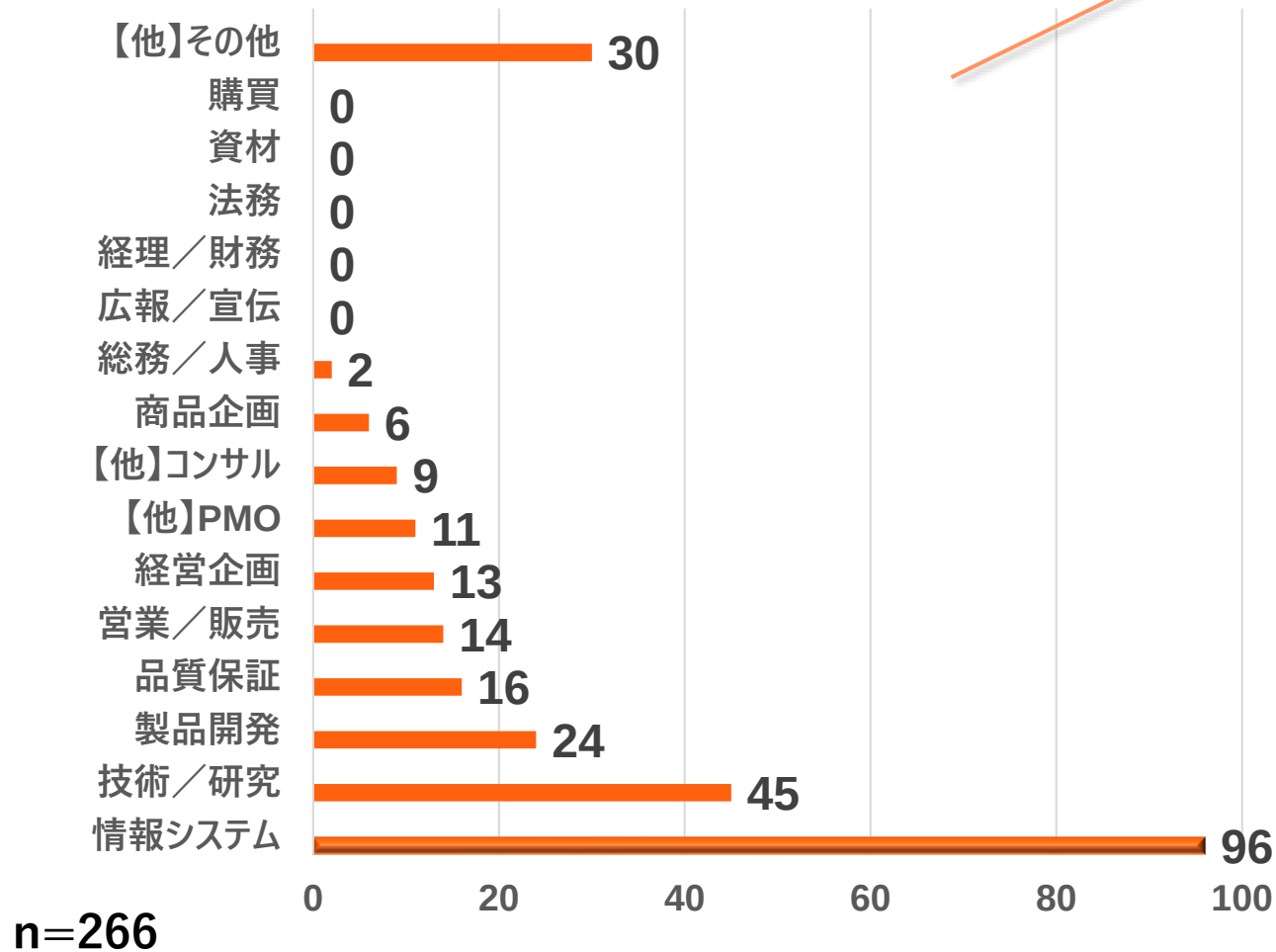
(参考) 所属企業の業種の推移



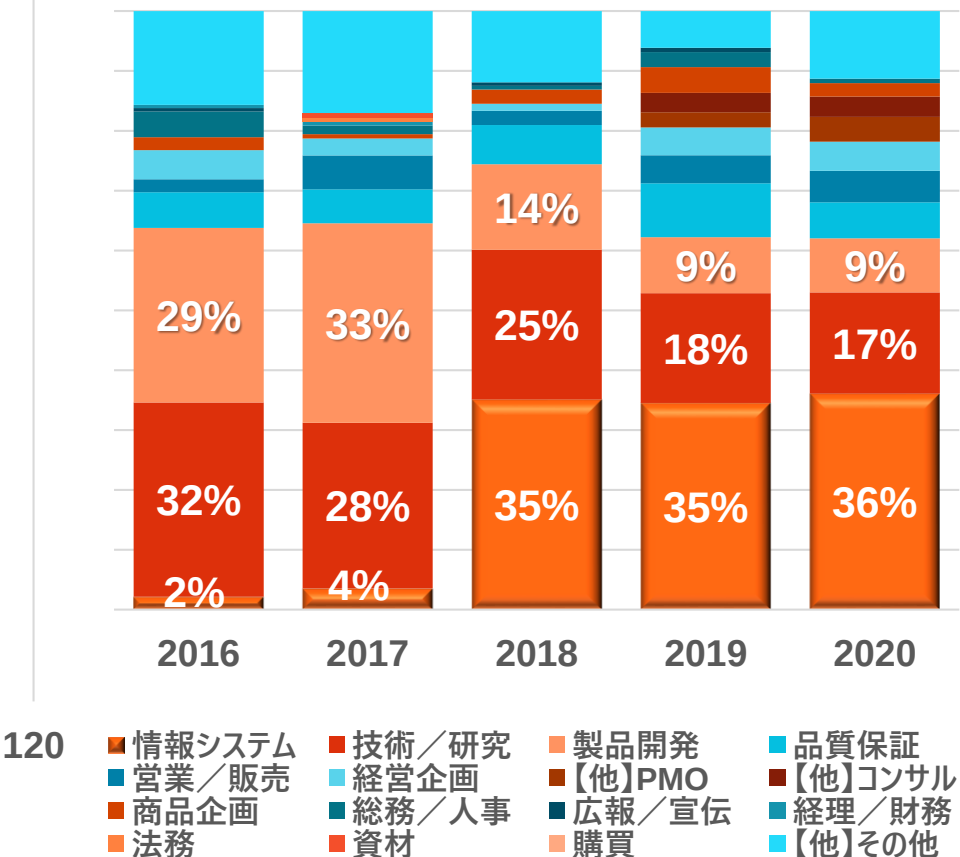
- 情報通信 (IT)
- 建設
- 石油/ガス
- 鉄鋼
- 自動車
- 製薬
- 化学
- 電気/電子
- 通信
- 機械/金属
- 食品
- 金融
- 教育
- 不動産
- 公共インフラ

# 回答者の属性 | 所属部門

約 6 割が情報システム部門以外



(参考) 所属部門の推移

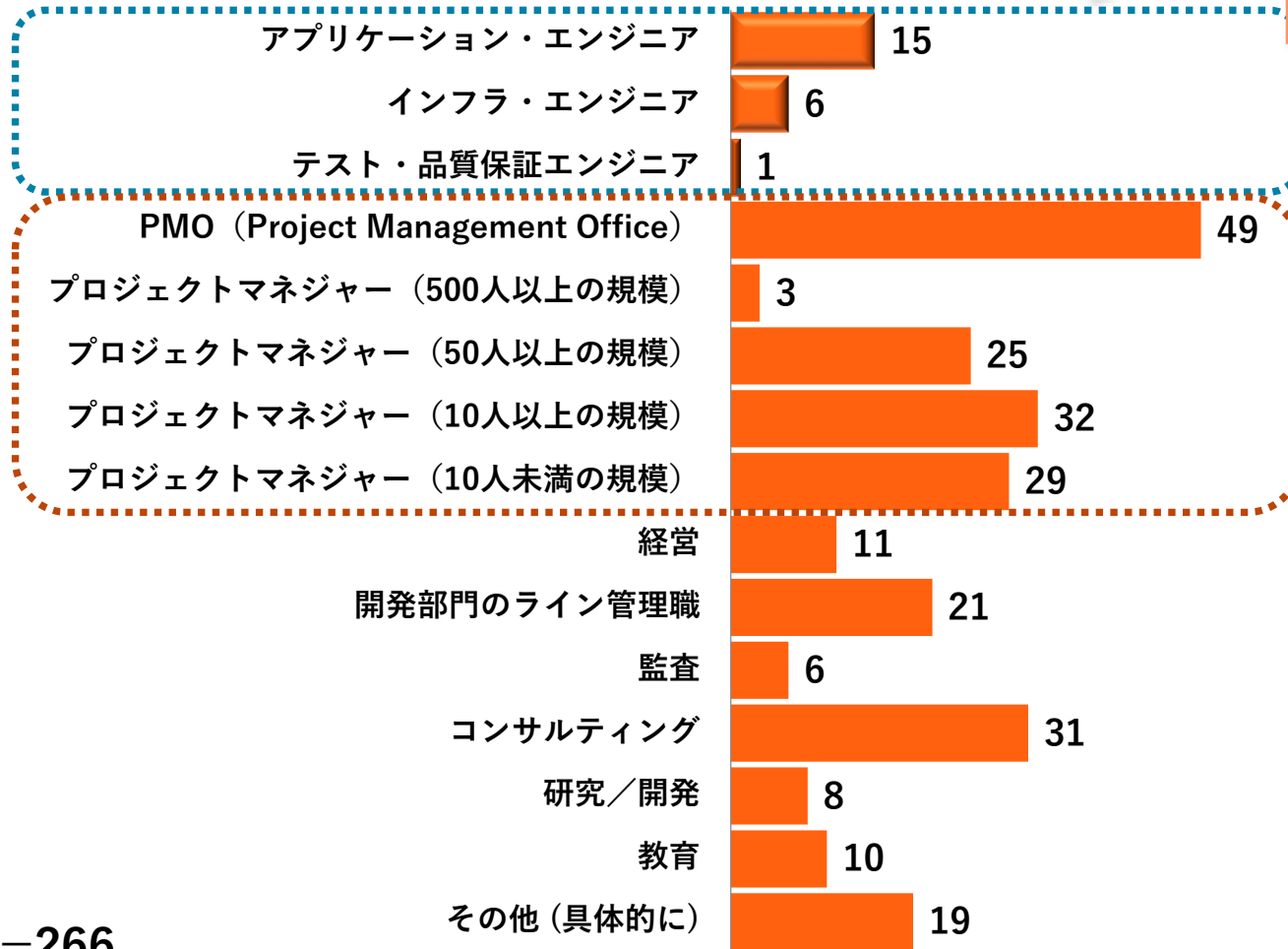


※ 情報システムの選択肢は2018年度から 12



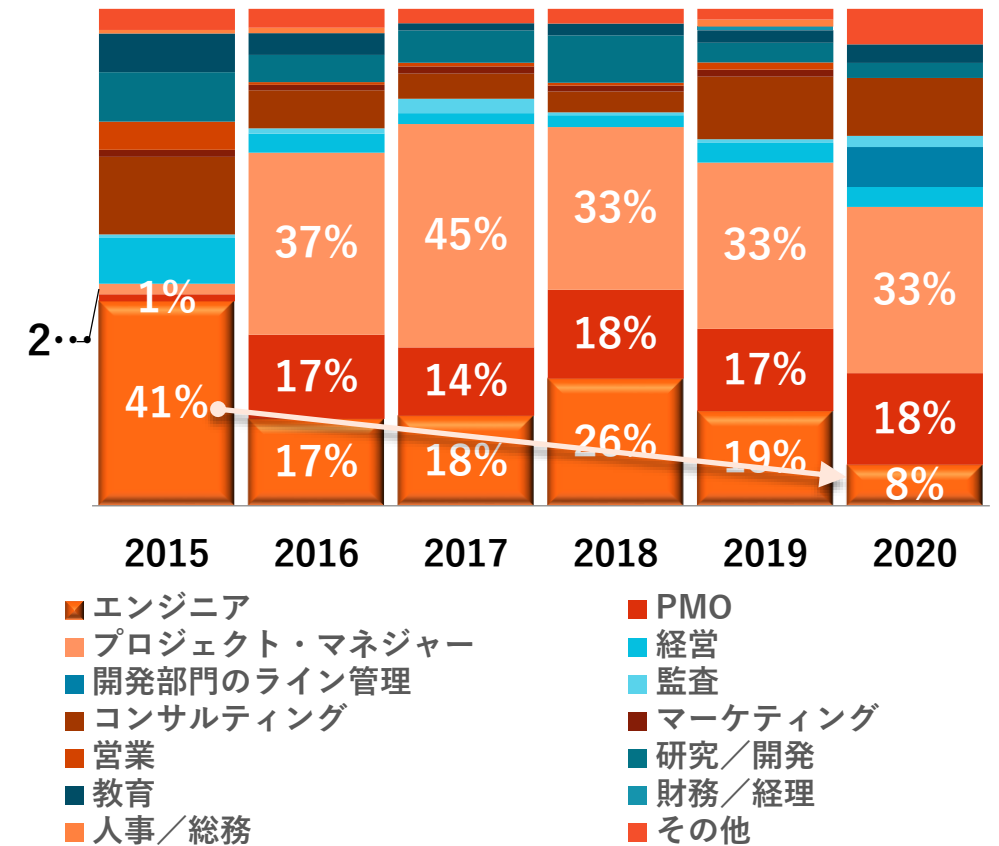
# 回答者の属性 | 職種

プロジェクトマネージャーとPMOが半分、  
エンジニアが減少傾向

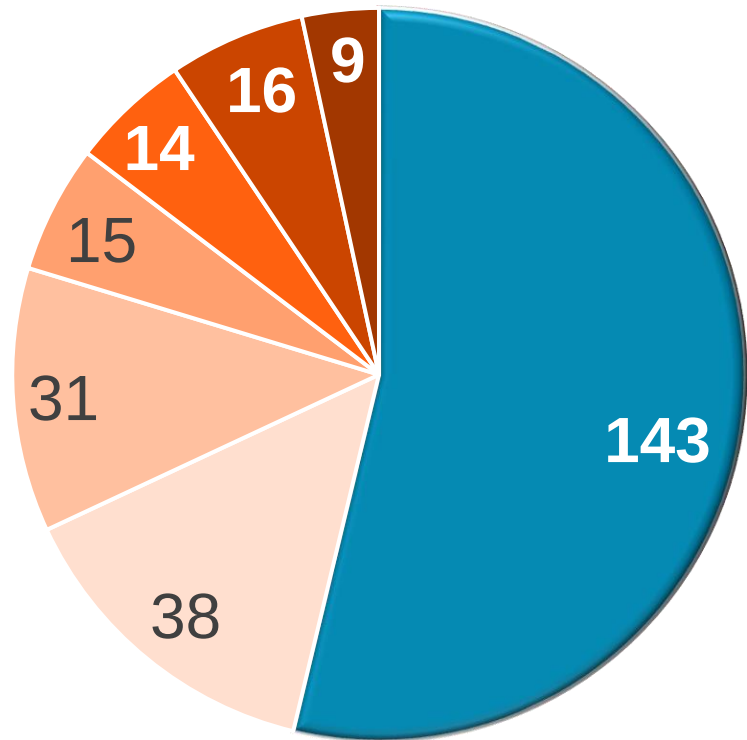


n=266

(参考) 職種の推移



# 回答者の属性 | アジャイル経験年数

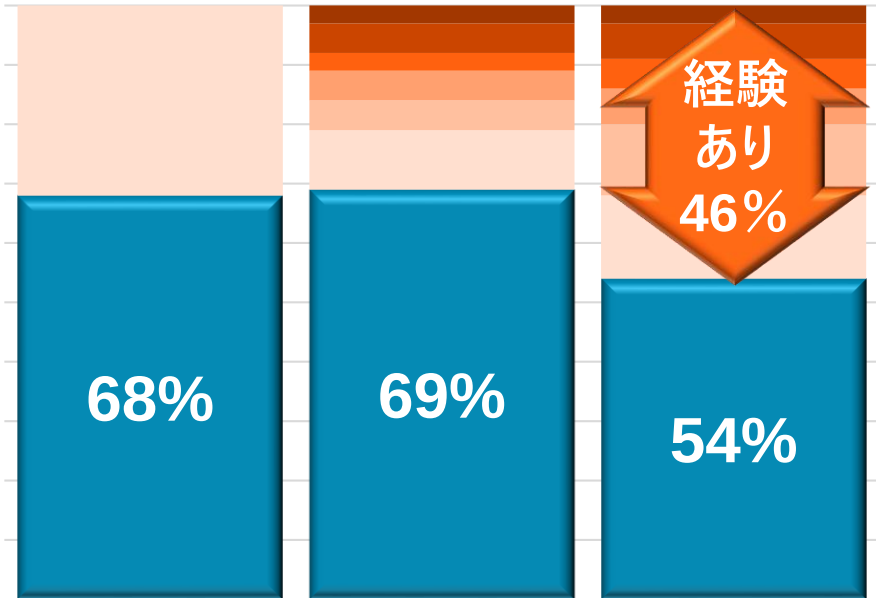


- 未経験
- 1年未満
- 1年以上 2年未満
- 2年以上 3年未満
- 3年以上 5年未満
- 5年以上 10年未満
- 10年以上

n=266

経験者はまだ半分弱だが、  
これまでで最も高い比率

(参考) アジャイル経験年数の推移



- 未経験
- 1年未満/経験あり
- 1年以上 2年未満
- 2年以上 3年未満
- 3年以上 5年未満
- 5年以上 10年未満
- 10年以上

2020年

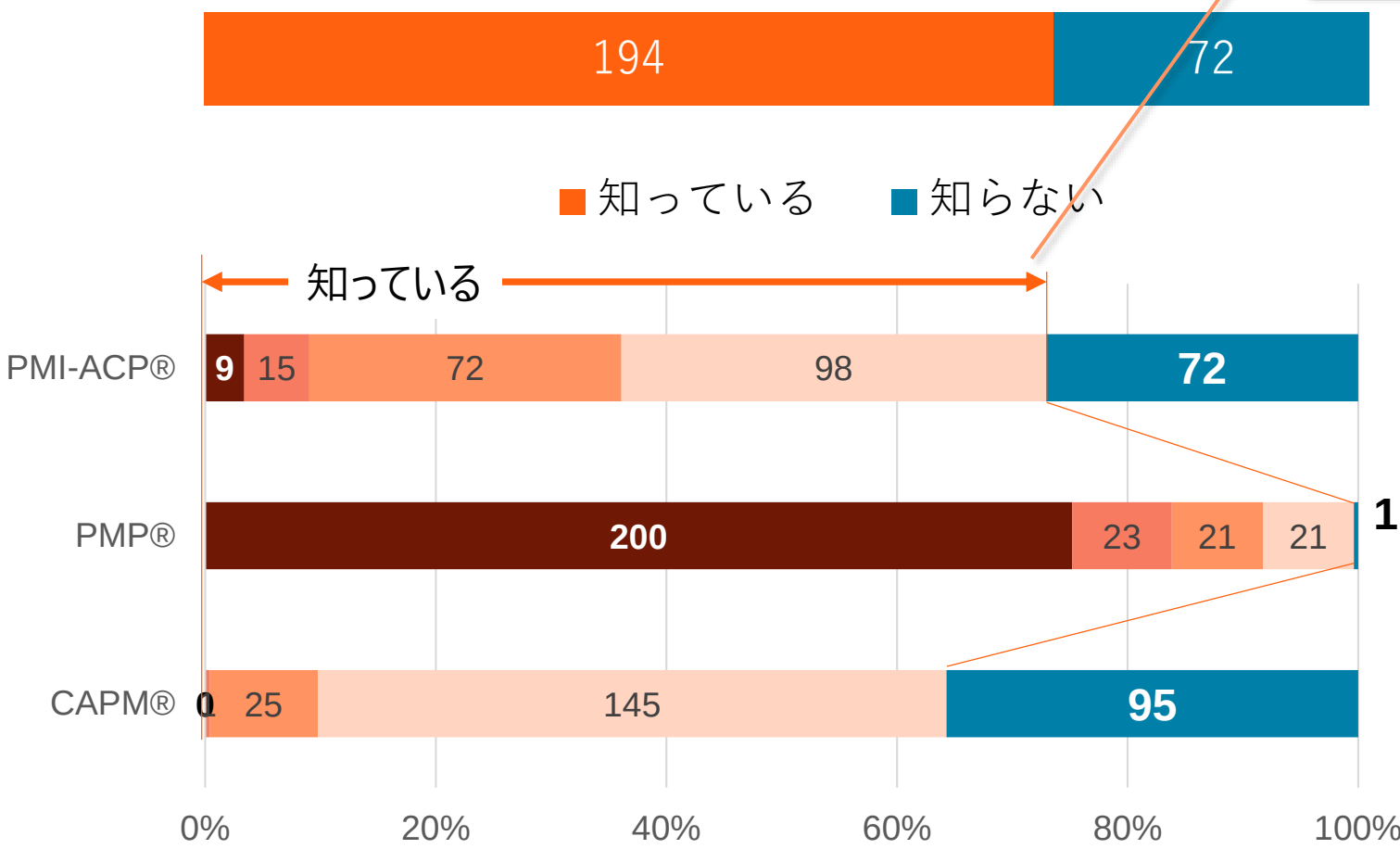
アジャイル プロジェクト マネジメント意識調査報告＜データ編＞

# 関連資格等の状況

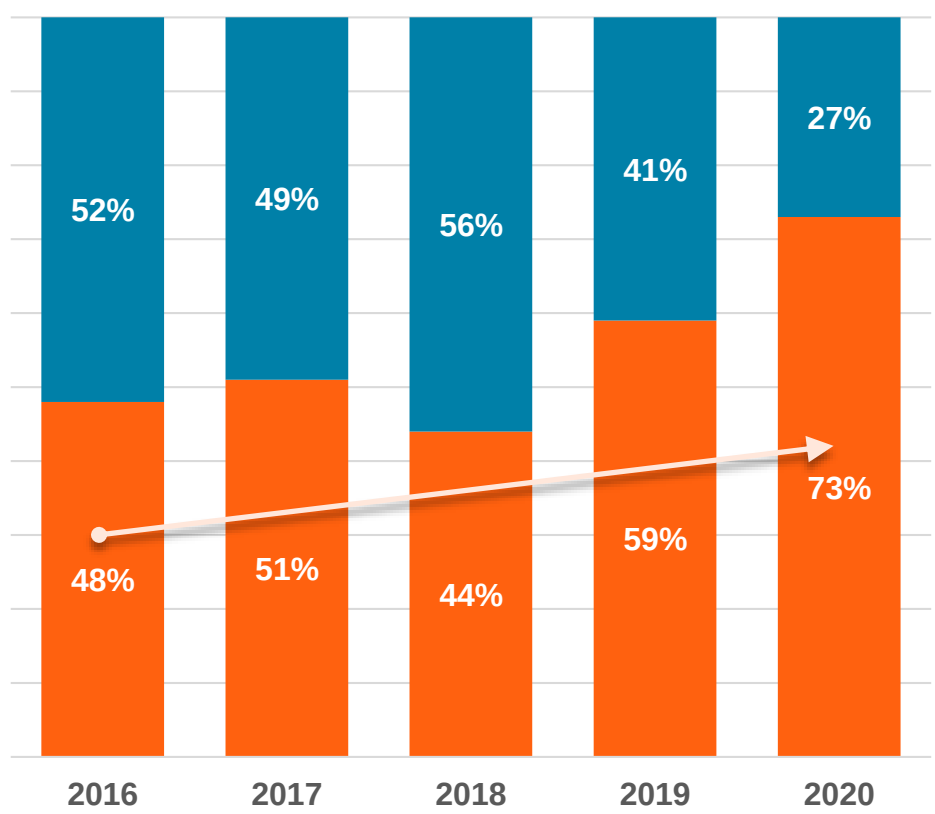
# 関連資格の状況 | 日本語で受験できるPMIの資格

PMP-ACP®認定資格の認知度

認知度は向上、  
取得者と取得予定者は多くない



(参考) PMP-ACP®認知度の推移

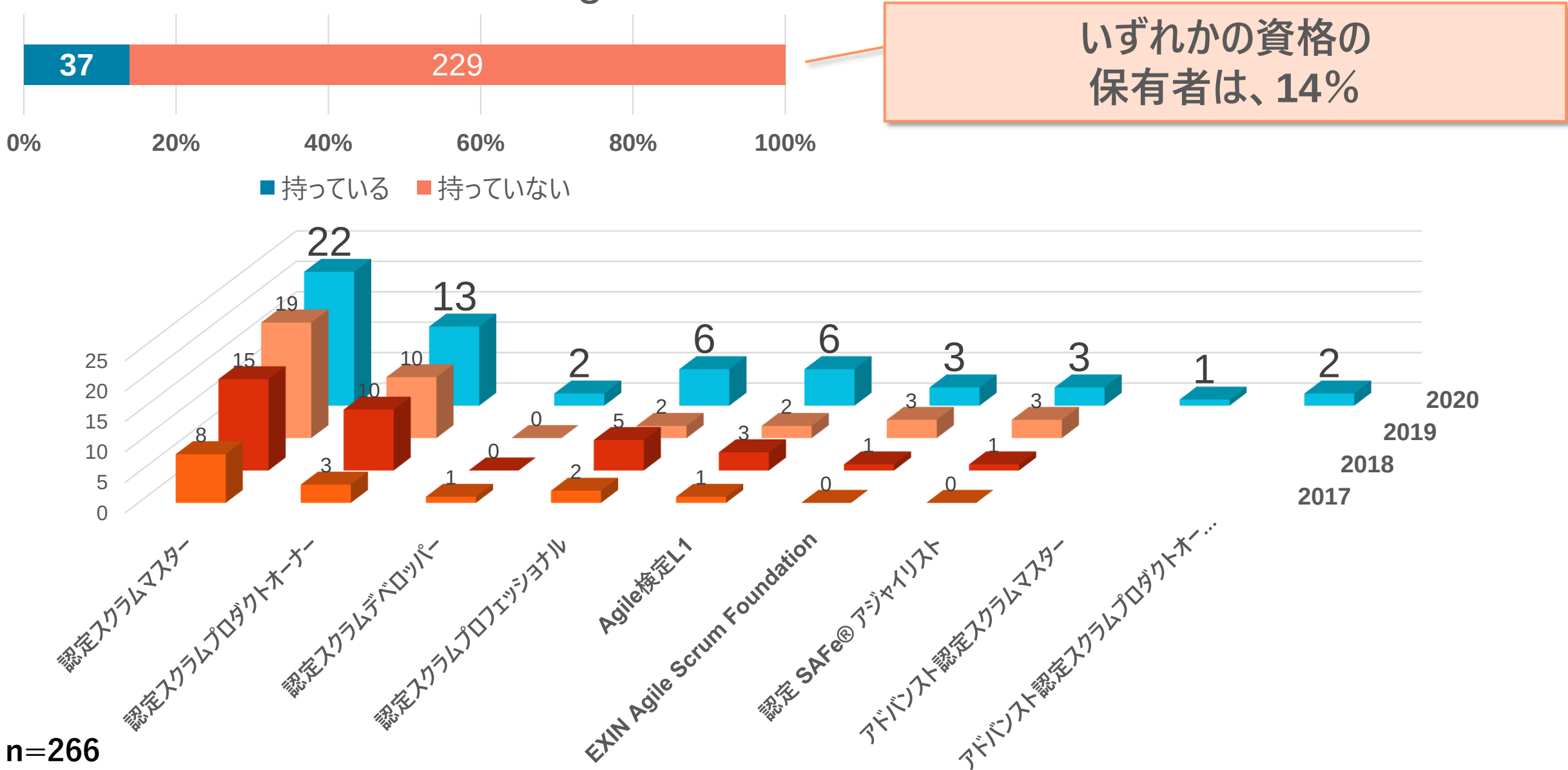


n=266 ■ 取得している ■ 取得予定 ■ 興味がある ■ 知っている ■ 知らない

PMI ACP; Agile Certified Practitioner, PMP; Project Management Professional, CAPM; Certified Associate in Project Management

■ 知っている ■ 知らない

# 関連資格の状況 | その他のAgile関連資格の取得状況

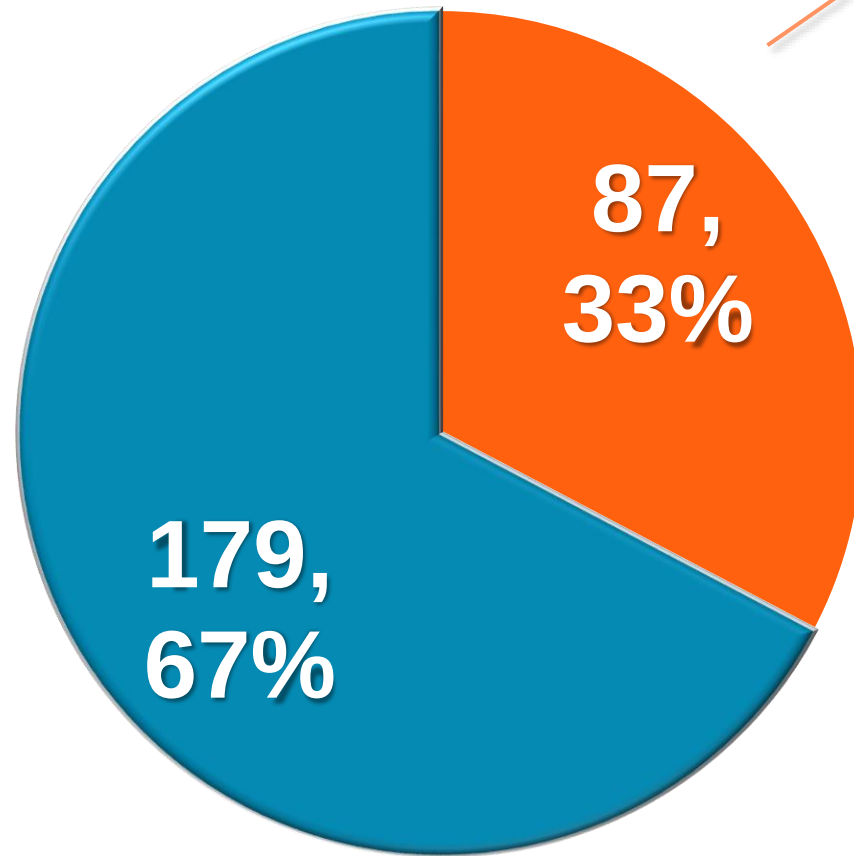


n=266



# ディシプリンド・アジャイルについて | PMIのディシプリンド・アジャイルの買収

3 割が「知っている」



■ 知っている ■ 知らない

n=266



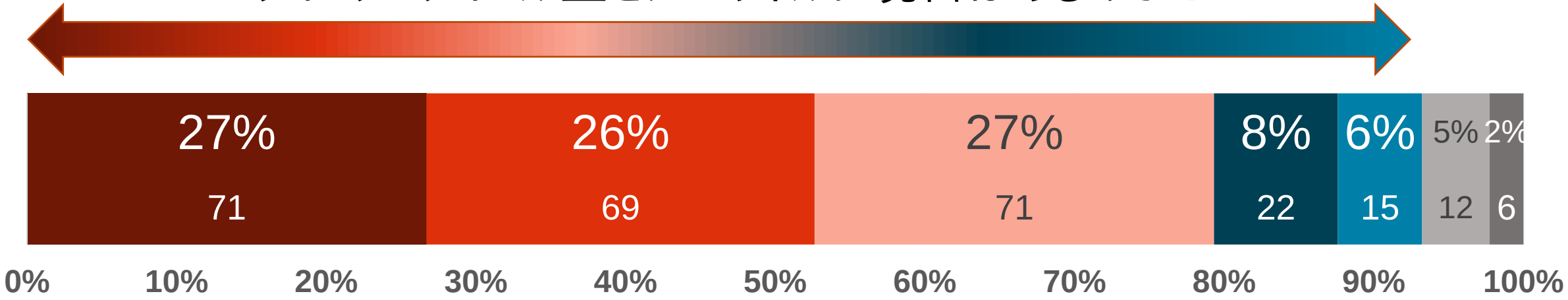
2020年

アジャイル プロジェクト マネジメント意識調査報告＜データ編＞

# 組織のプロジェクトへの取組み

# 組織のプロジェクトへの取組み | 主要なプロジェクトマネジメント・アプローチ

ウォーターフォール型とアジャイルに境目はあるのか？



- ウォーターフォール型で、厳密なゲート管理を行っている
- ウォーターフォールに近いが、ある程度の手戻りを受け入れる余裕を見込んでいる
- ウォーターフォールに近いが、状況に応じて柔軟に変更することを前提にしている

- スクラムなど、特定のアジャイル・アプローチを適用している
- 特定のアジャイル手法を前提とせず、自分の組織の手法を開発している

- わからない
- その他 (具体的に)

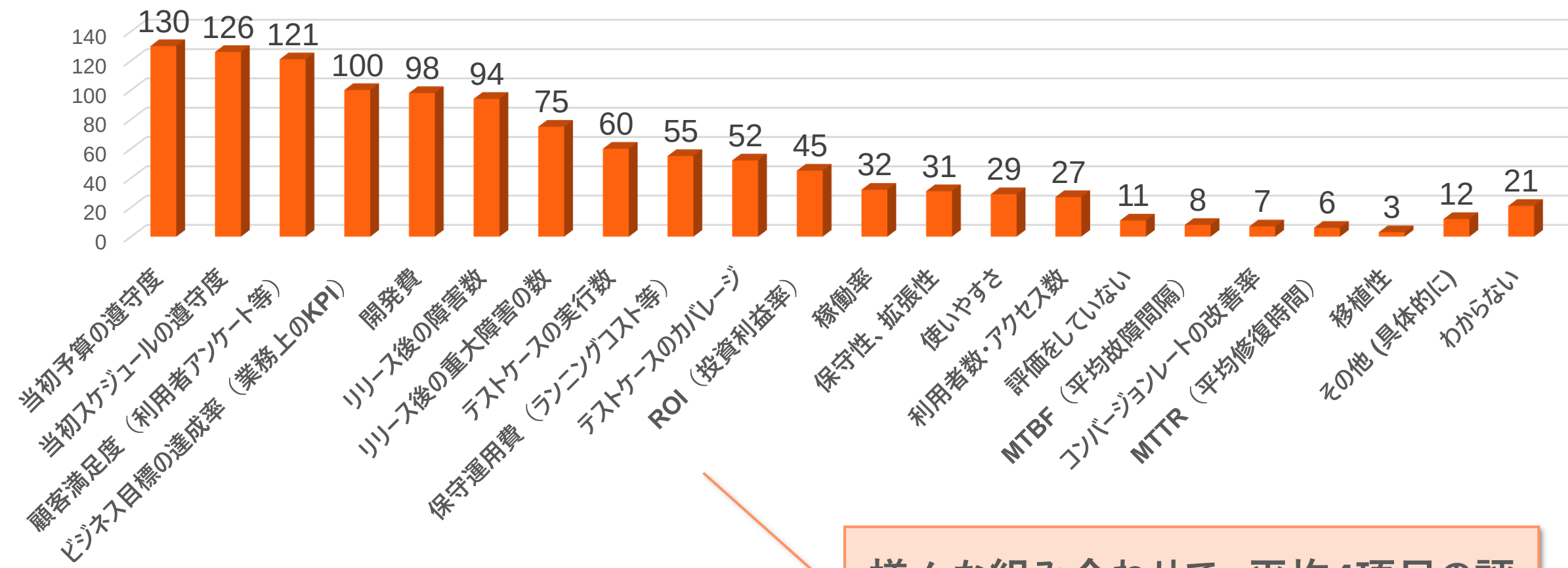
「ウォーターフォール」と称していても、  
実はバリエーションがある

n=266





# 組織のプロジェクトへの取組み | プロジェクトの評価指標

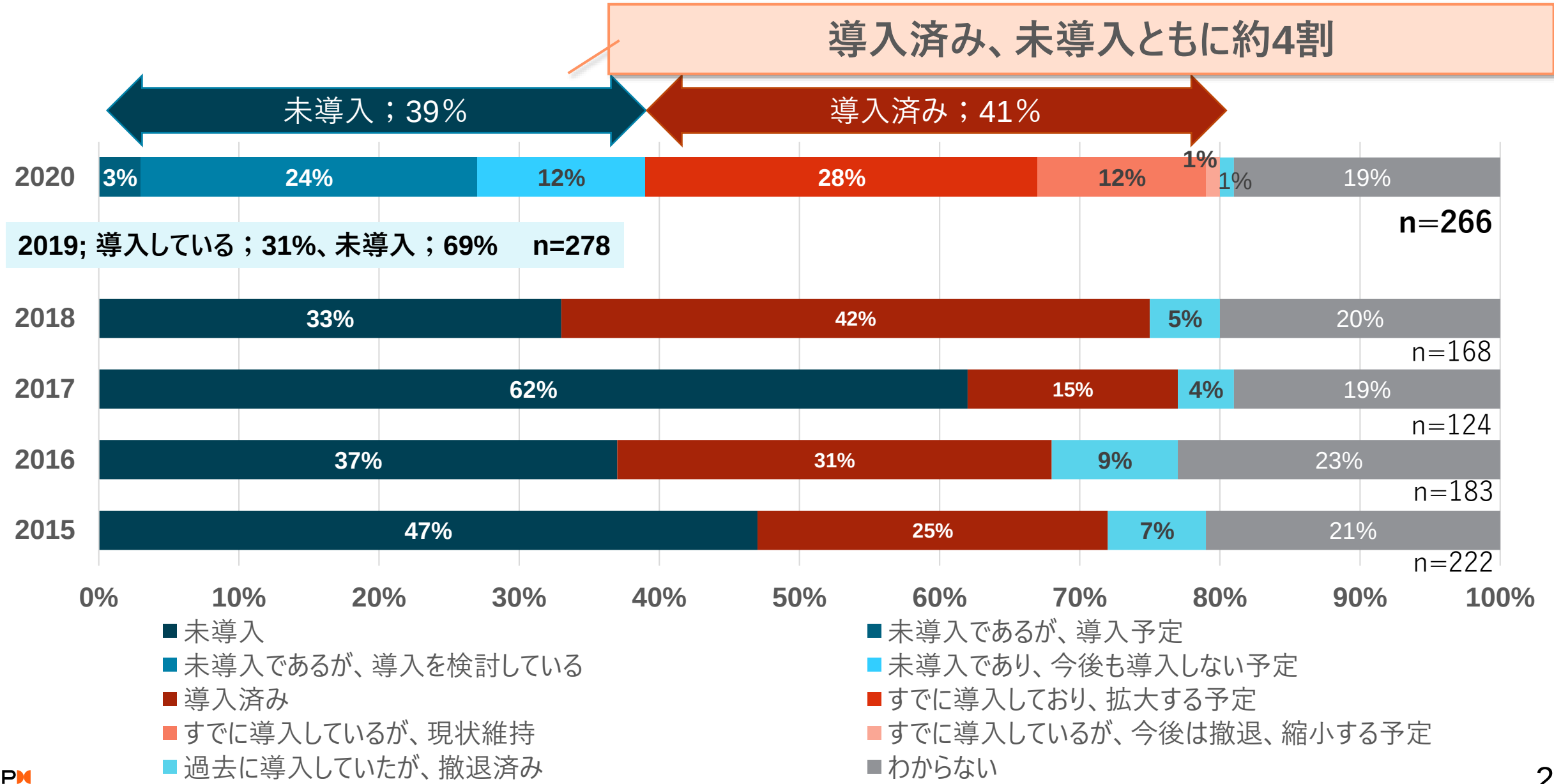


n=266



様々な組み合わせで、平均4項目の評価指標を使っている

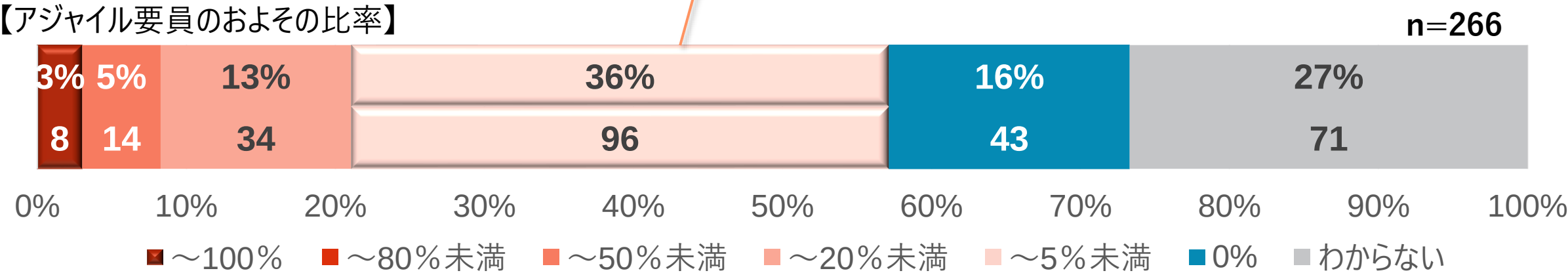
# 組織のプロジェクトへの取組み | アジャイル導入方針



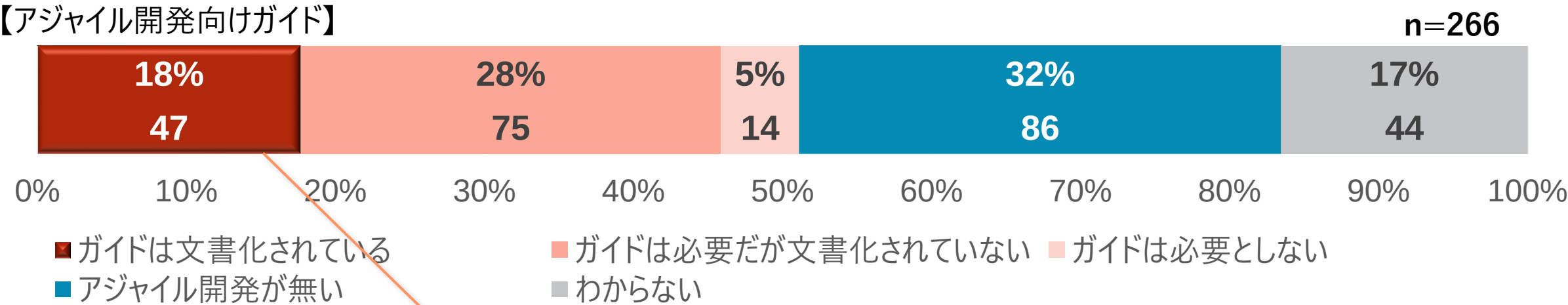
# 組織のプロジェクトへの取組み | アジャイル関連要員/アジャイル開発ガイド

多くの組織でアジャイル関連要員は5%未満

【アジャイル要員のおよその比率】



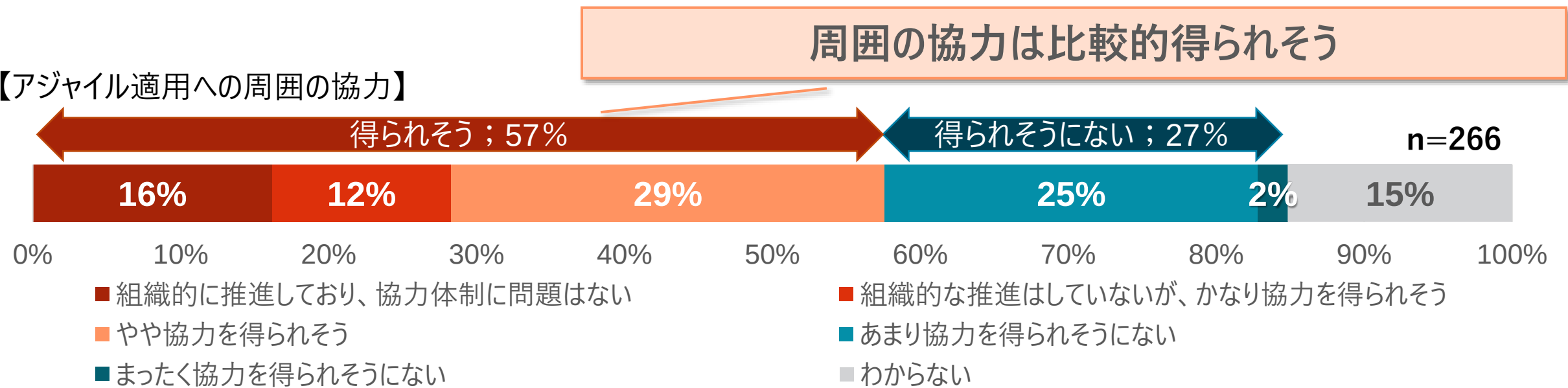
【アジャイル開発向けガイド】



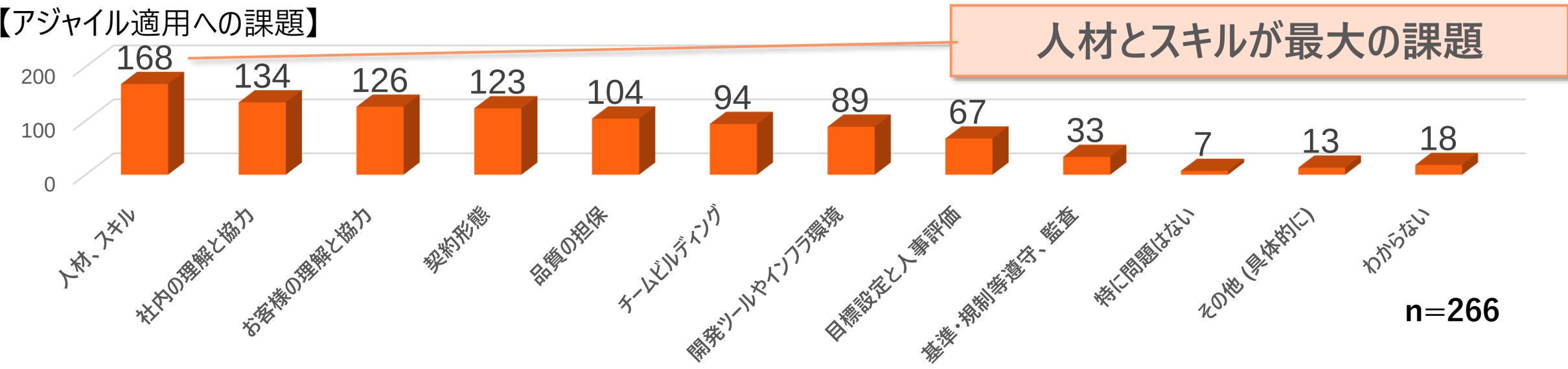
「文書化」は比較的進んでいる

# 組織のプロジェクトへの取組み | アジャイル適用への周囲の協力/適用への課題

【アジャイル適用への周囲の協力】



【アジャイル適用への課題】



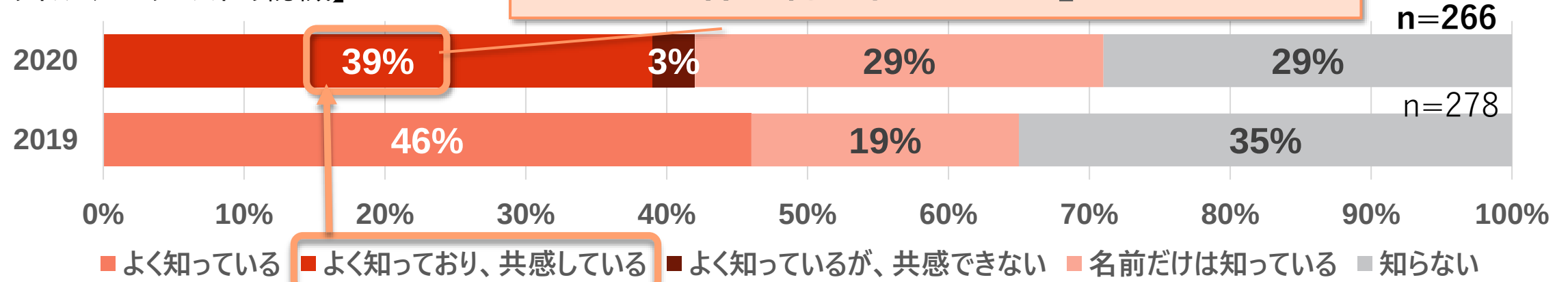
2020年

アジャイル プロジェクト マネジメント意識調査報告＜データ編＞

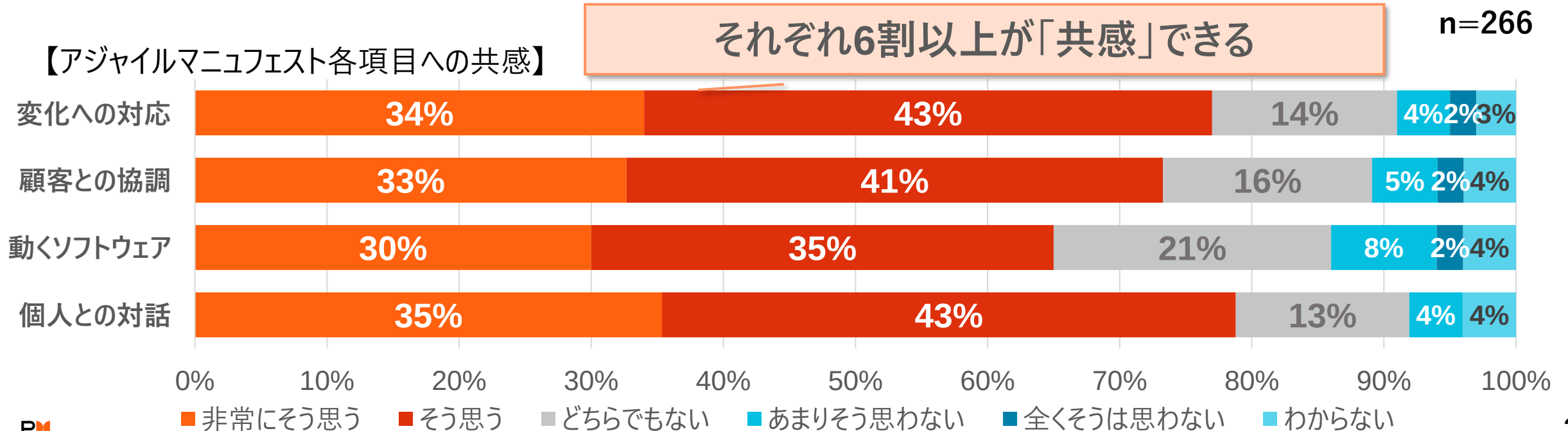
# 回答者のアジャイルへの認識

# 回答者のアジャイルへの認識 | アジャイルマニフェスト

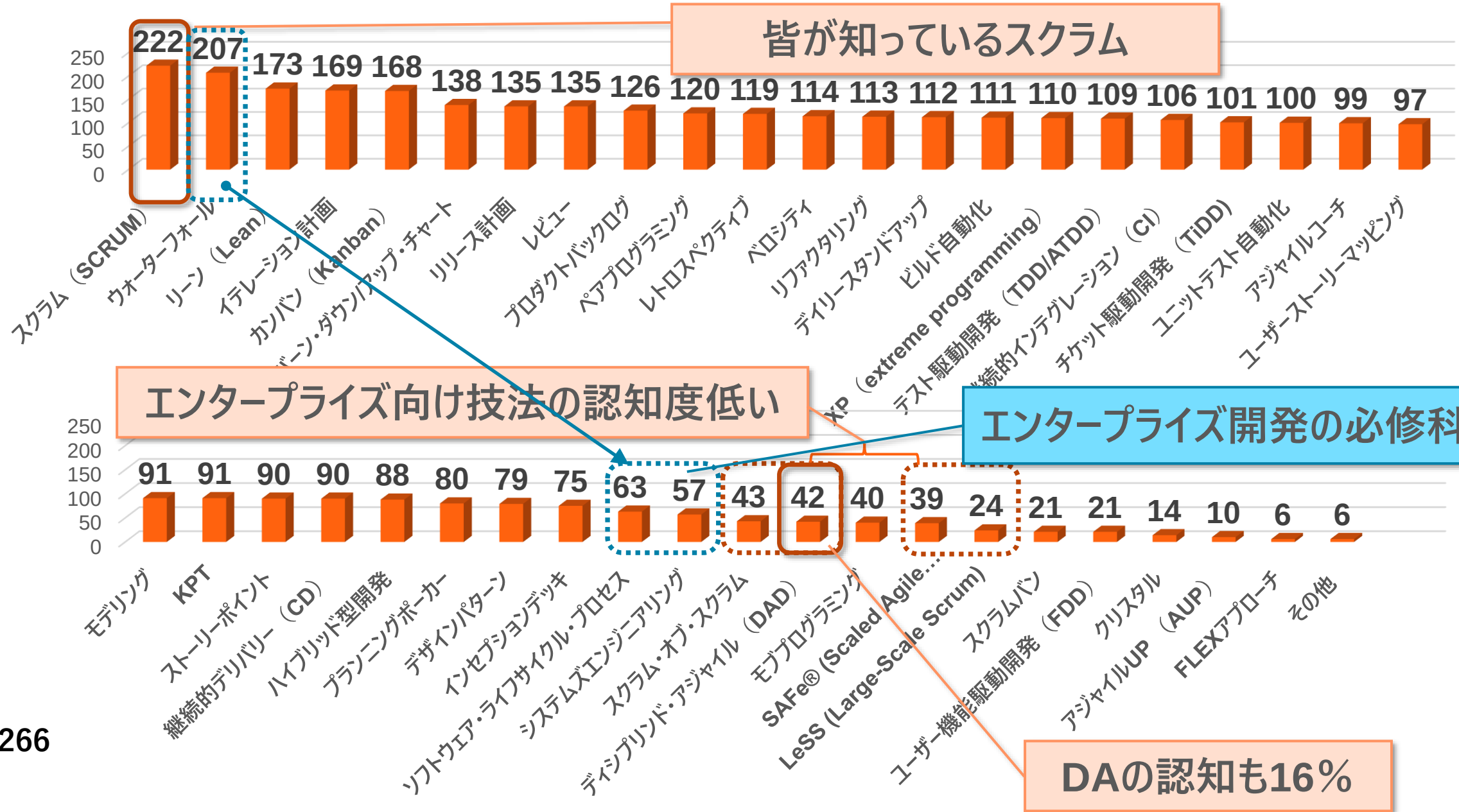
## 【アジャイルマニフェストの認識】



## 【アジャイルマニフェスト各項目への共感】



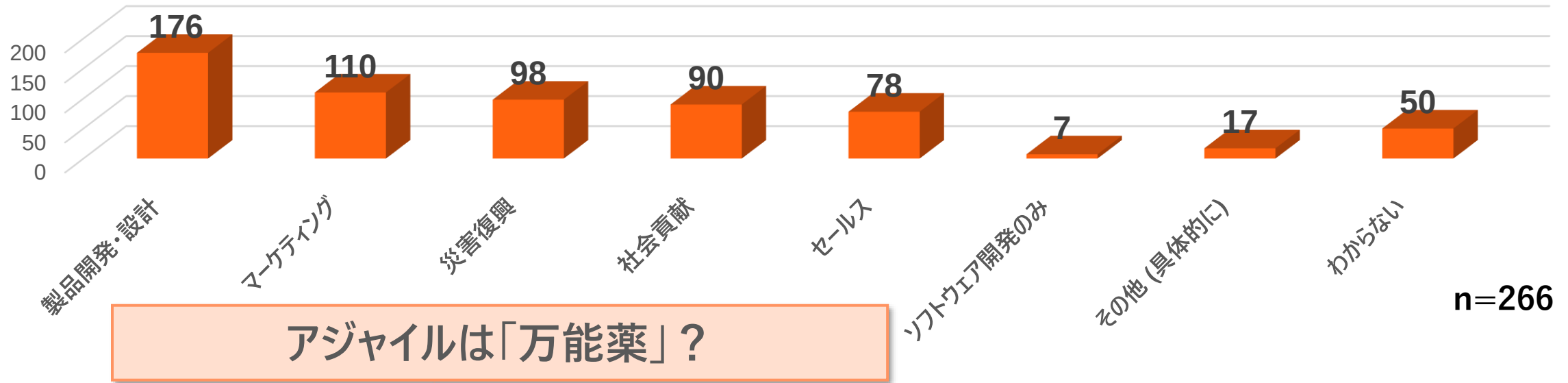
# 回答者のアジャイルへの認識 | 関連用語の認知度



n=266



# 回答者のアジャイルへの認識 | ソフトウェア開発以外の適用可能領域



## □ その他（具体的に）

<コーポレート業務> 企画、PoC, 戦略策定, 他に、企画なども。そもそも製造業はずっと昔からAGILEでしたよね？  
設問がおかしいと思いますが、組織運営, 経営全般, 危機管理, 人事・採用, 内部監査、特に継続的モニタリングとの連携において、教育開発

<研究開発全般> R&D, 不確実で複雑な領域のすべて, 食品開発, 基本的にモノ作りすべて  
全て, 新規事業開発, 不確実性が高いプロジェクト全般, すべてのプロジェクト

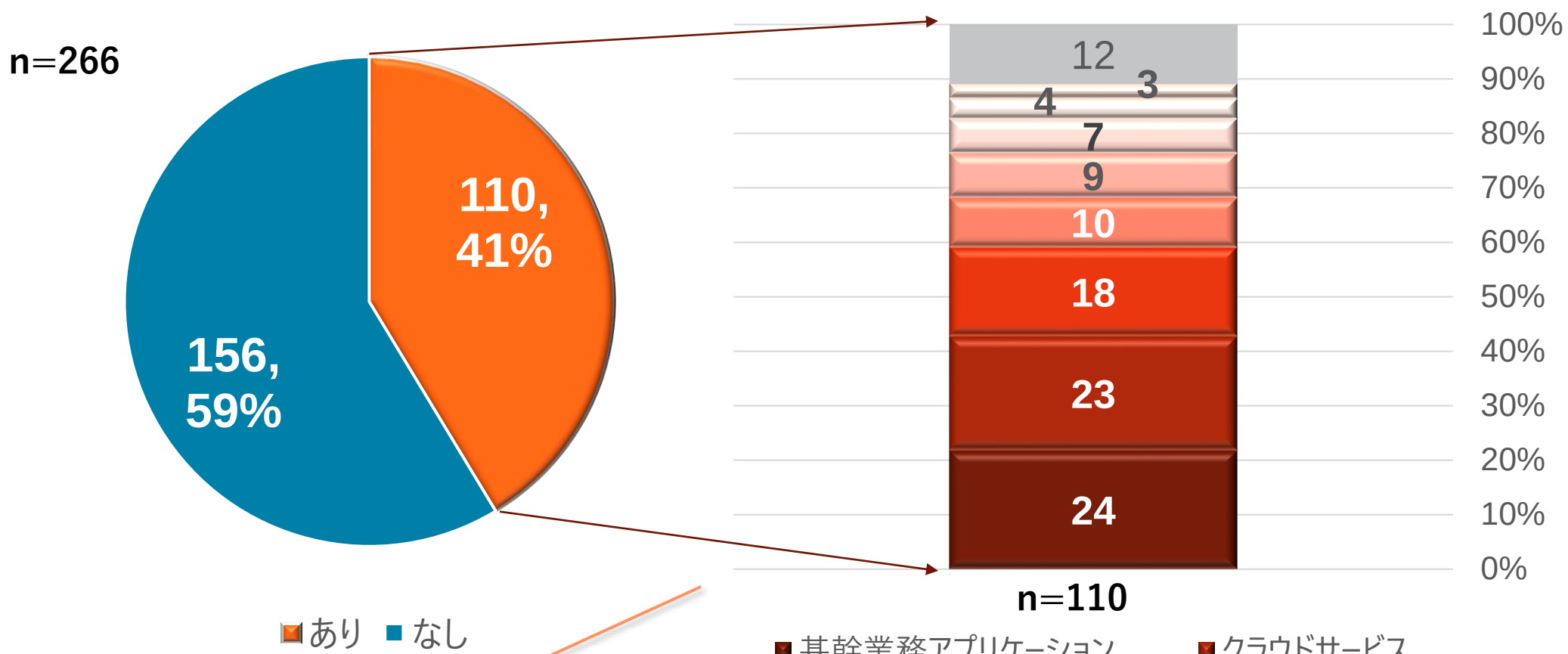


2020年

アジャイル プロジェクト マネジメント意識調査報告＜データ編＞

# 回答者の経験したアジャイル

# 回答者の経験したアジャイル | アジャイル適用業務の経験とその種類



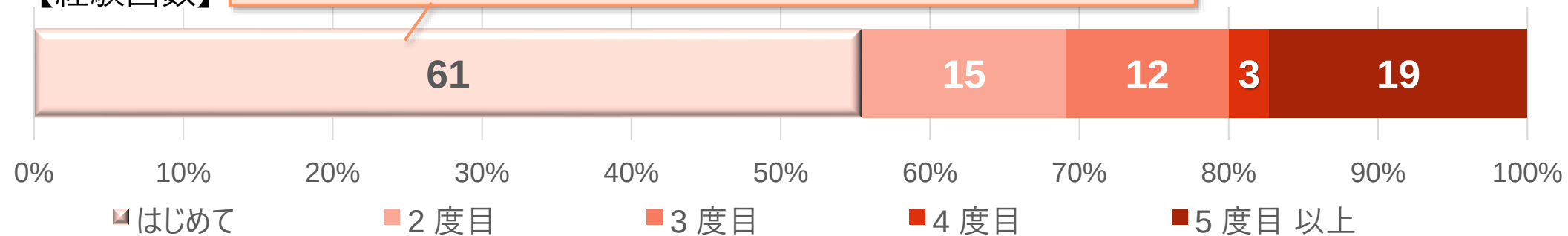
回答者の4割が経験者、従来領域の他に、クラウドサービスやアプリ開発に活用

- 基幹業務アプリケーション
- 自社用サービスやツール
- 外販用パッケージソフト
- IoTソリューション関連
- その他
- クラウドサービス
- スマートデバイス向けアプリ
- 組み込みソフト
- ロボット関連

# 回答者の経験したアジャイル | 経験回数と役割

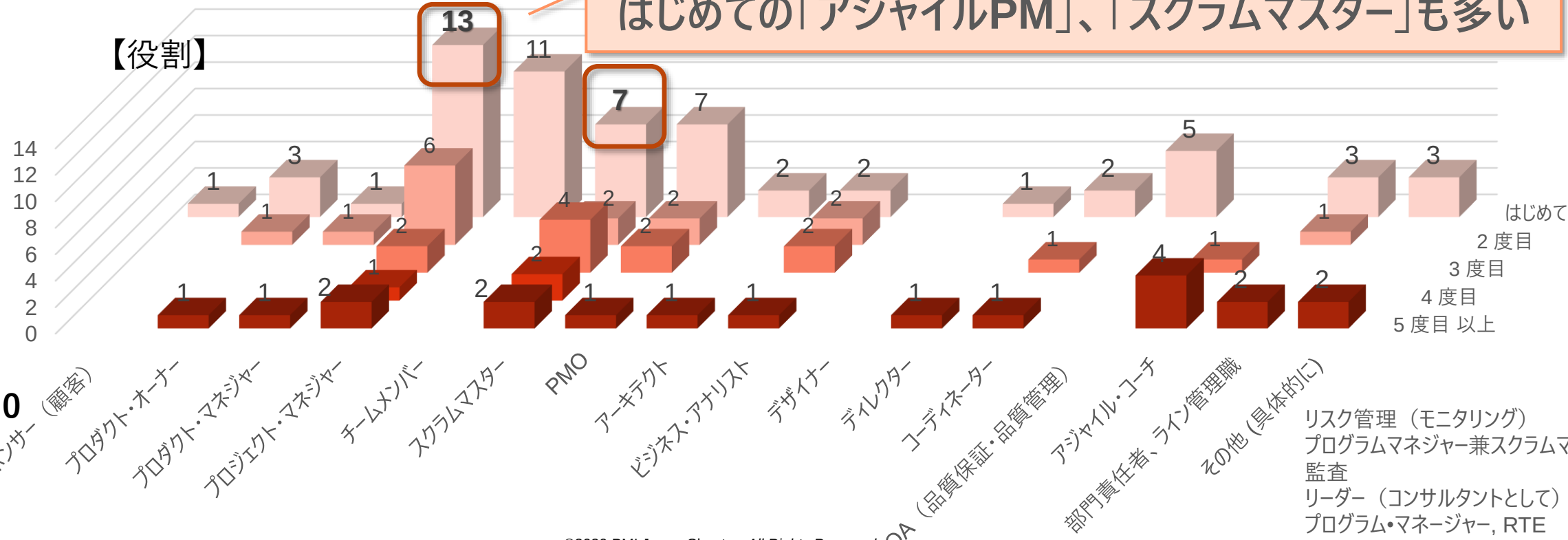
【経験回数】

経験者の半数以上が「はじめてのアジャイル」



【役割】

はじめての「アジャイルPM」、「スクラムマスター」も多い



n=110

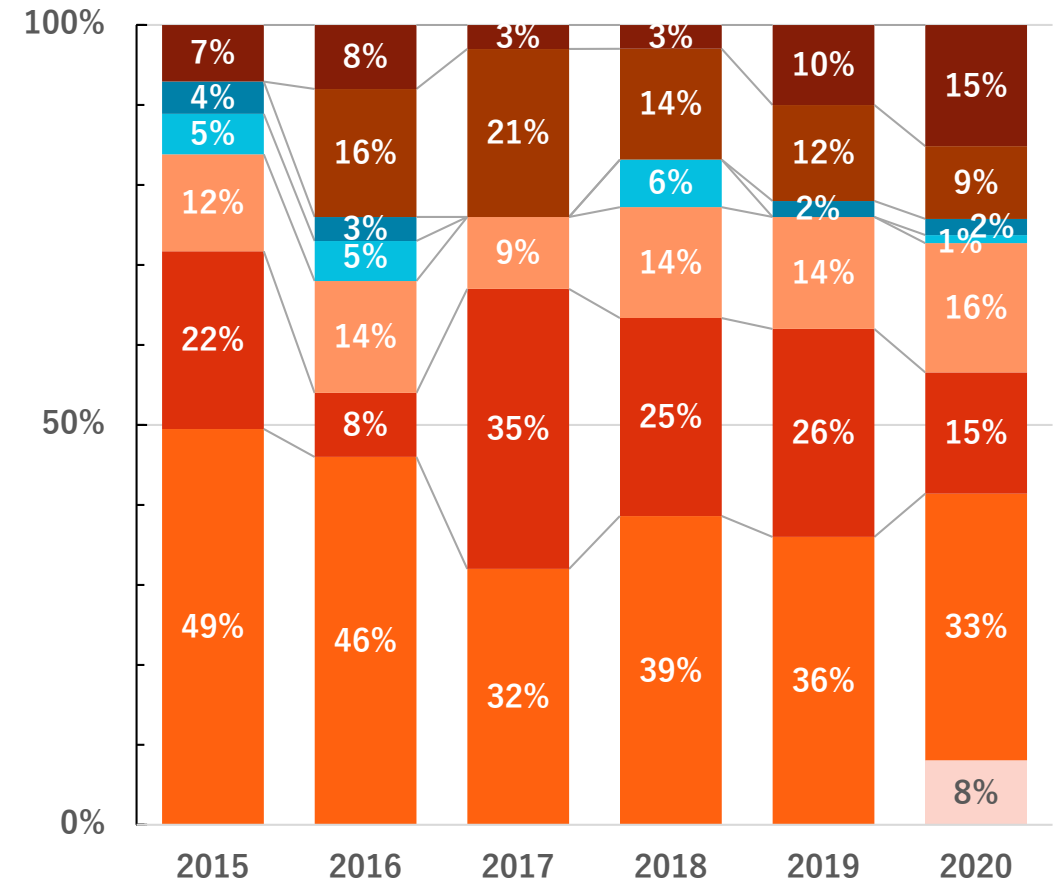


リスク管理 (モニタリング)  
プログラムマネジャー兼スクラムマスター  
監査  
リーダー (コンサルタントとして)  
プログラム・マネージャー, RTE

# 回答者の経験したアジャイル | アジャイル採用理由

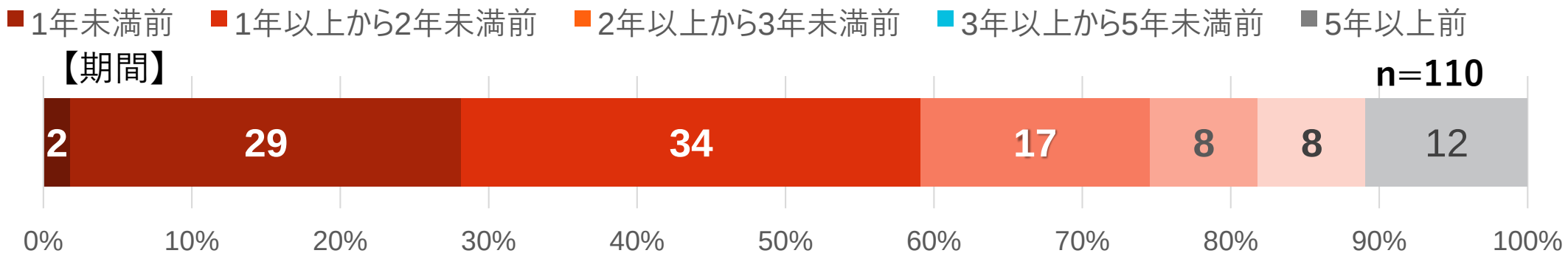
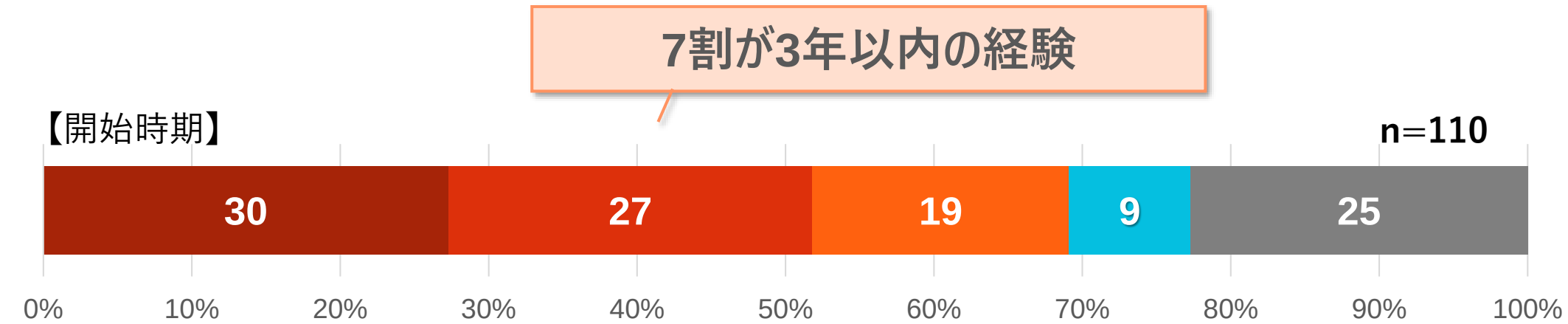


- 組織として適用することが規定されている
- チームが自主的に判断し導入
- 経営層や上司からのトップダウンでの指示
- スポンサーや発注先からの指示
- セミナーなどで興味をもったから
- 競合他社との兼ね合い（他社との差別化）
- 自分がアジャイルを推進する立場で周囲を説得した
- その他（具体的に）



小規模な実験検証、製品アーキテクチャー下層がアジャイルで開発したため、導入後にPMOとして入ったので知らない。、既に導入されていた、結果的にそうだった。保秘の都合もある。、調達条件に記載されていた、当たり前の事なので。、そのプロダクト開発には探索型アプローチが最適であったため、スマートデバイス向けアプリ開発は顧客からのフィードバックが重要な要素であったため。、salesforceが推進して、顧客に説明していたから、試行錯誤しながら開始したので自然とアジャイルになった、アプリ開発自体はハイブリッド、部分的なプロセスにアジャイルプラクティスを適用、組織としてウォーターフォールの課題やリスクを受容する文化がそもそも存在していない、結果的に最適な形に変化していった、リリーススケジュールがタイトであったため。、顧客にとっての成果を最大にするため、それが当たり前だから

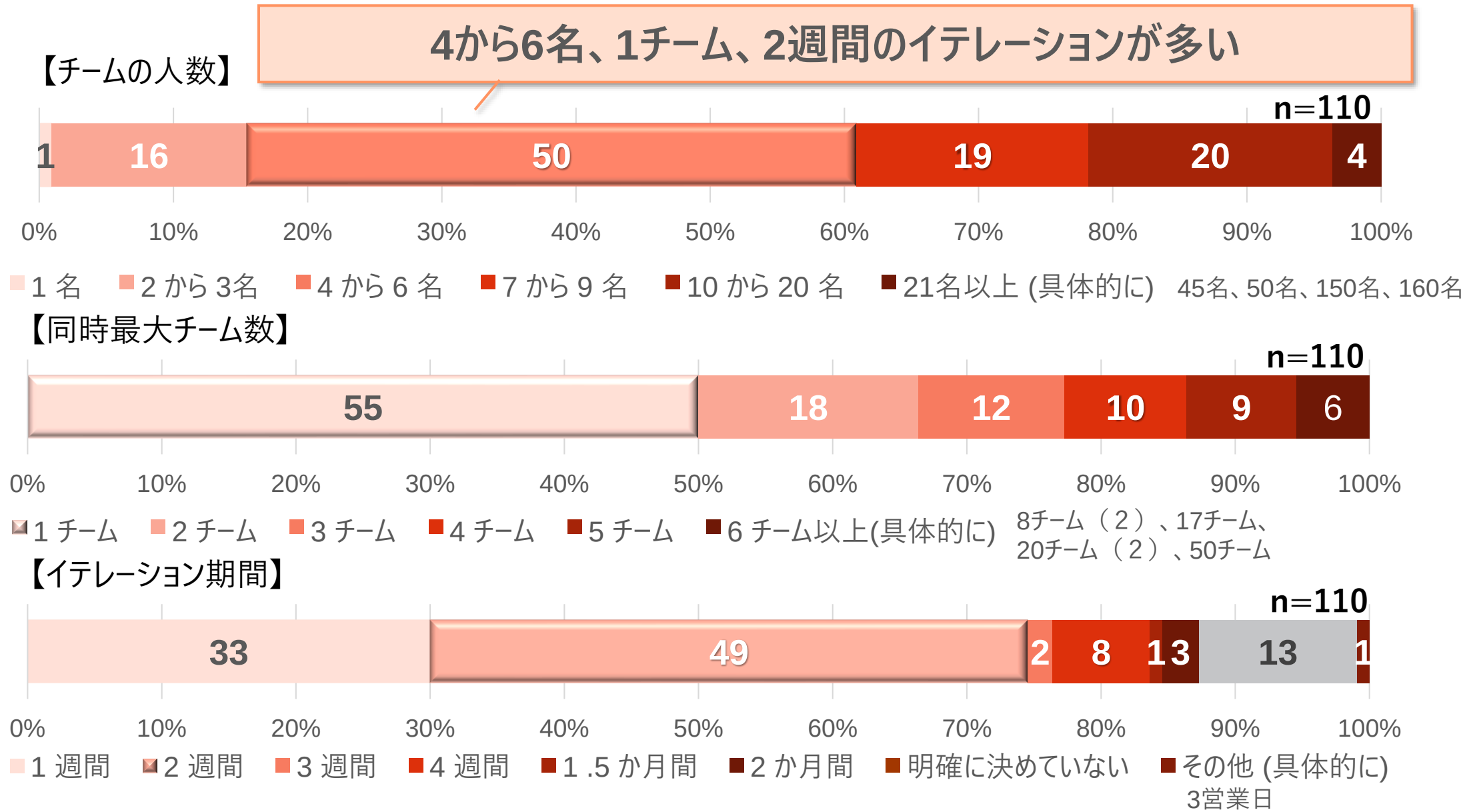
# 回答者の経験したアジャイル | 開始時期と期間



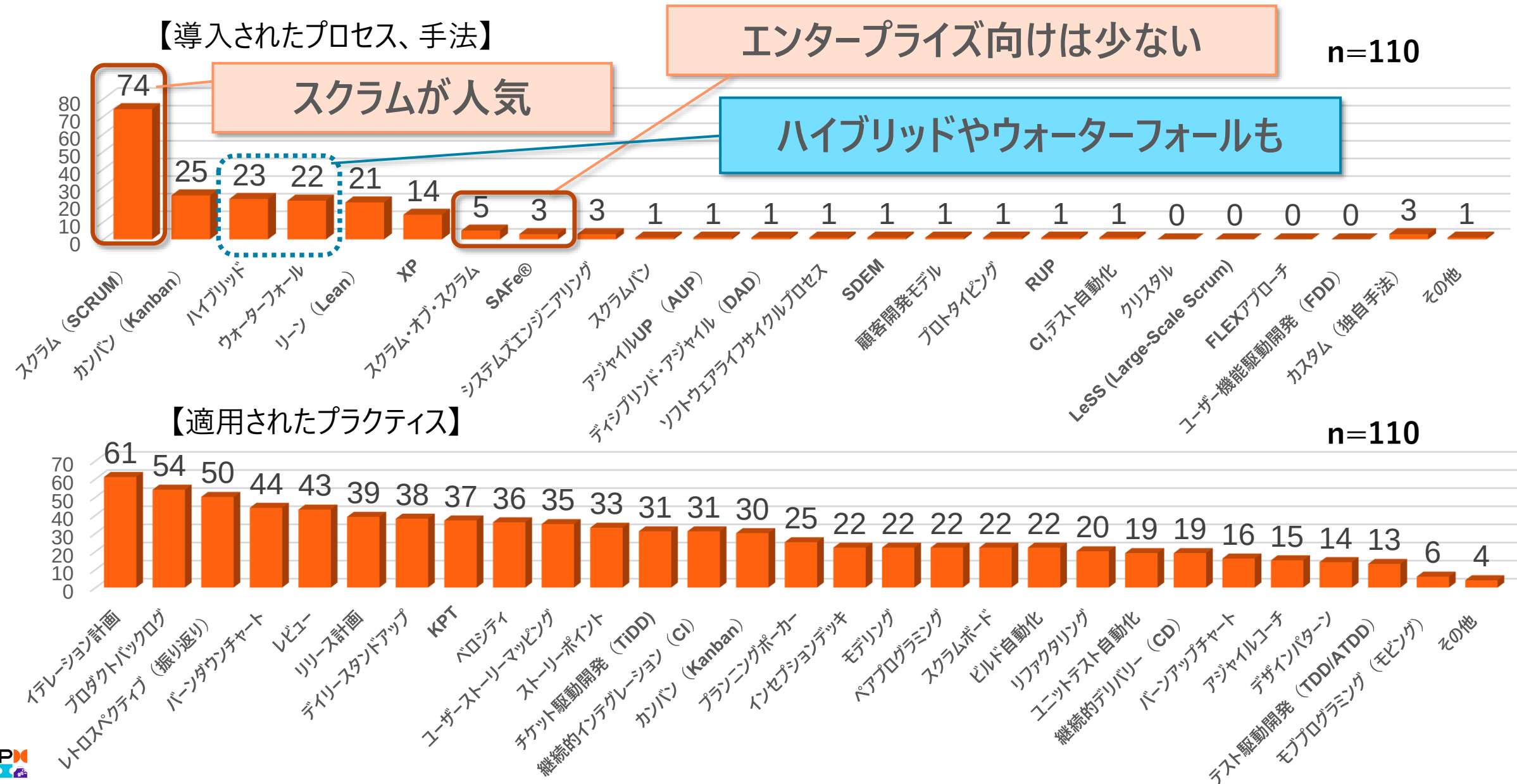
- 2か月未満
- 2か月以上から6か月未満
- 6か月以上から12か月未満
- 12か月以上から18か月未満
- 18か月以上から24か月未満
- 24か月以上
- 今も継続中

**6割が1年以内の期間**

# 回答者の経験したアジャイル | チーム人数、チーム数とイテレーション期間



# 回答者の経験したアジャイル | 導入されたプロセス・手法、プラクティス

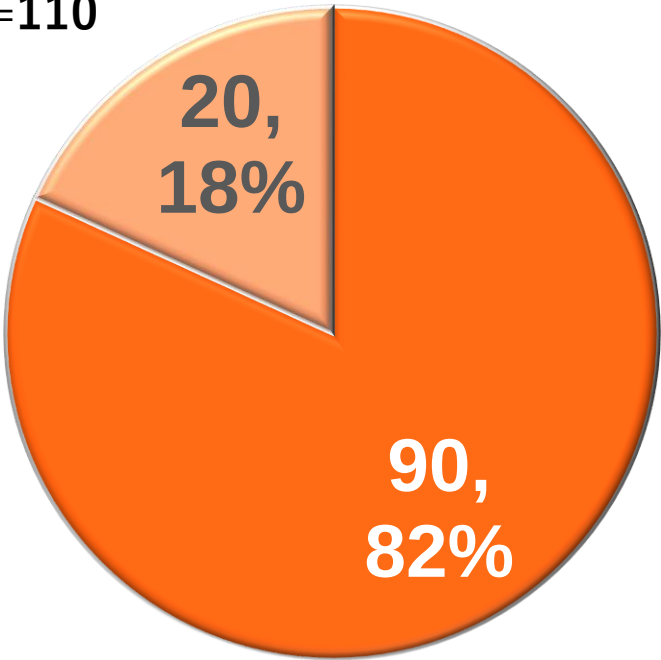


# 回答者の経験したアジャイル | ソフトウェア開発との関係

ソフトウェア開発が8割

【ソフトウェア開発を伴うか】

n=110

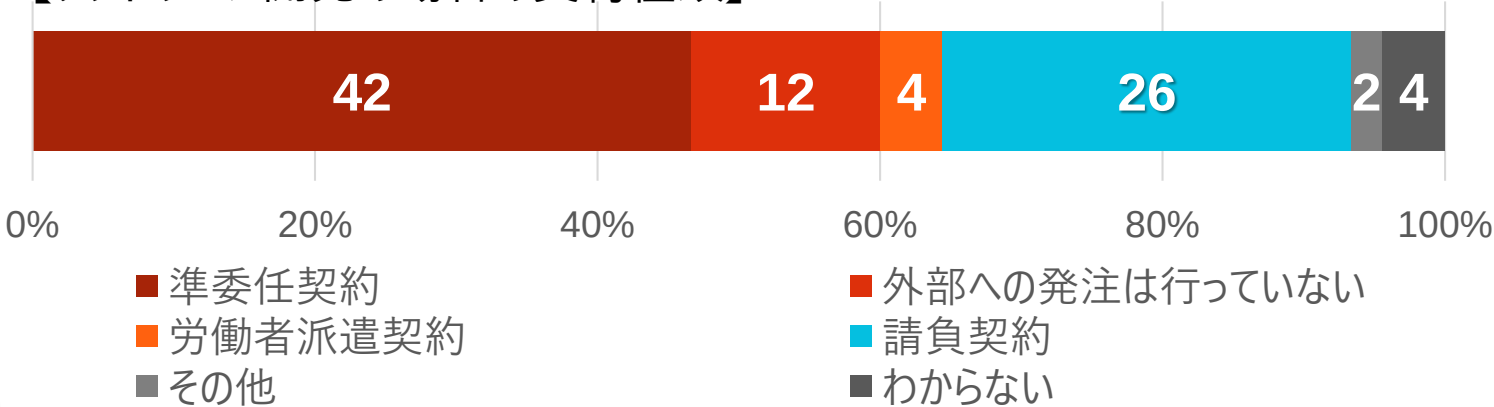


■ソフトウェア開発 ■ソフトウェア開発ではない

種類は準委任、サイクルはプロジェクト期間が多い

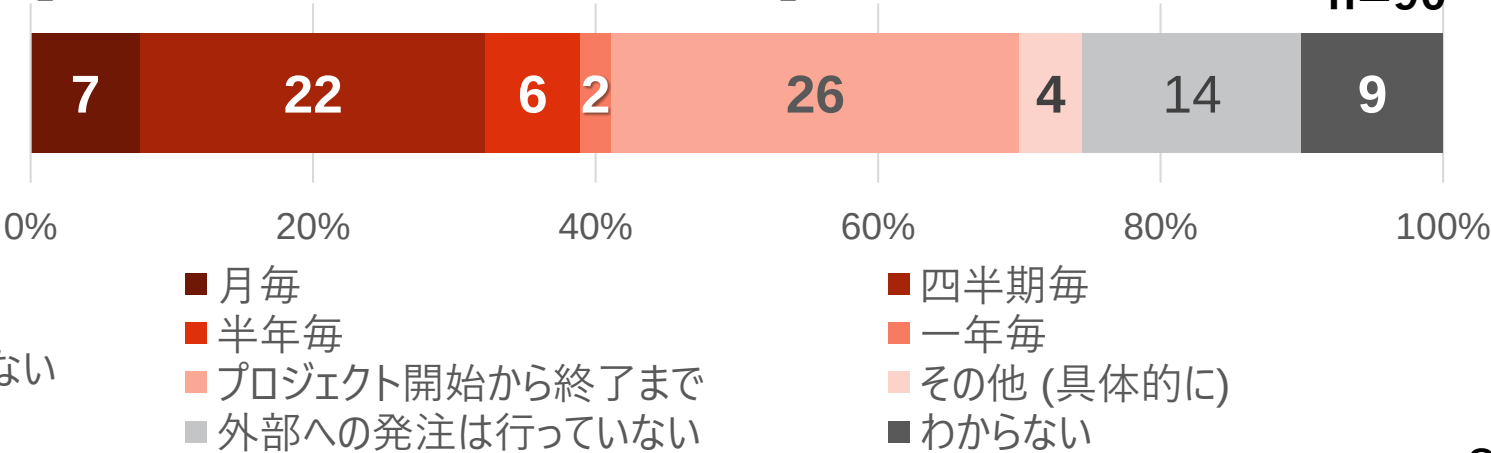
【ソフトウェア開発の場合の契約種類】

n=90



【ソフトウェア開発の場合の契約サイクル】

n=90



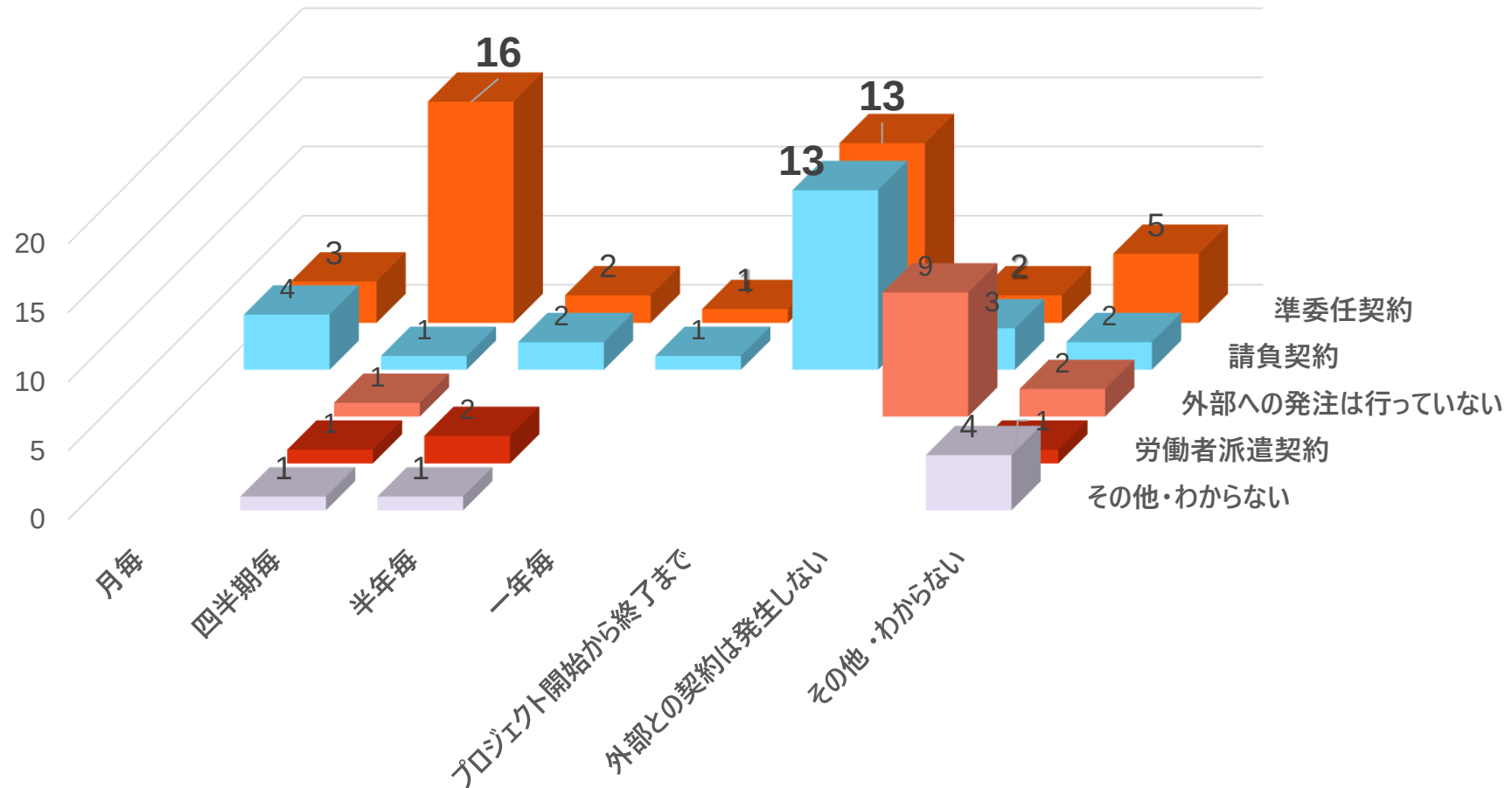


# 回答者の経験したアジャイル | ソフトウェア開発との関係

組み合わせてみると、四半期ごとの準委任契約が最多

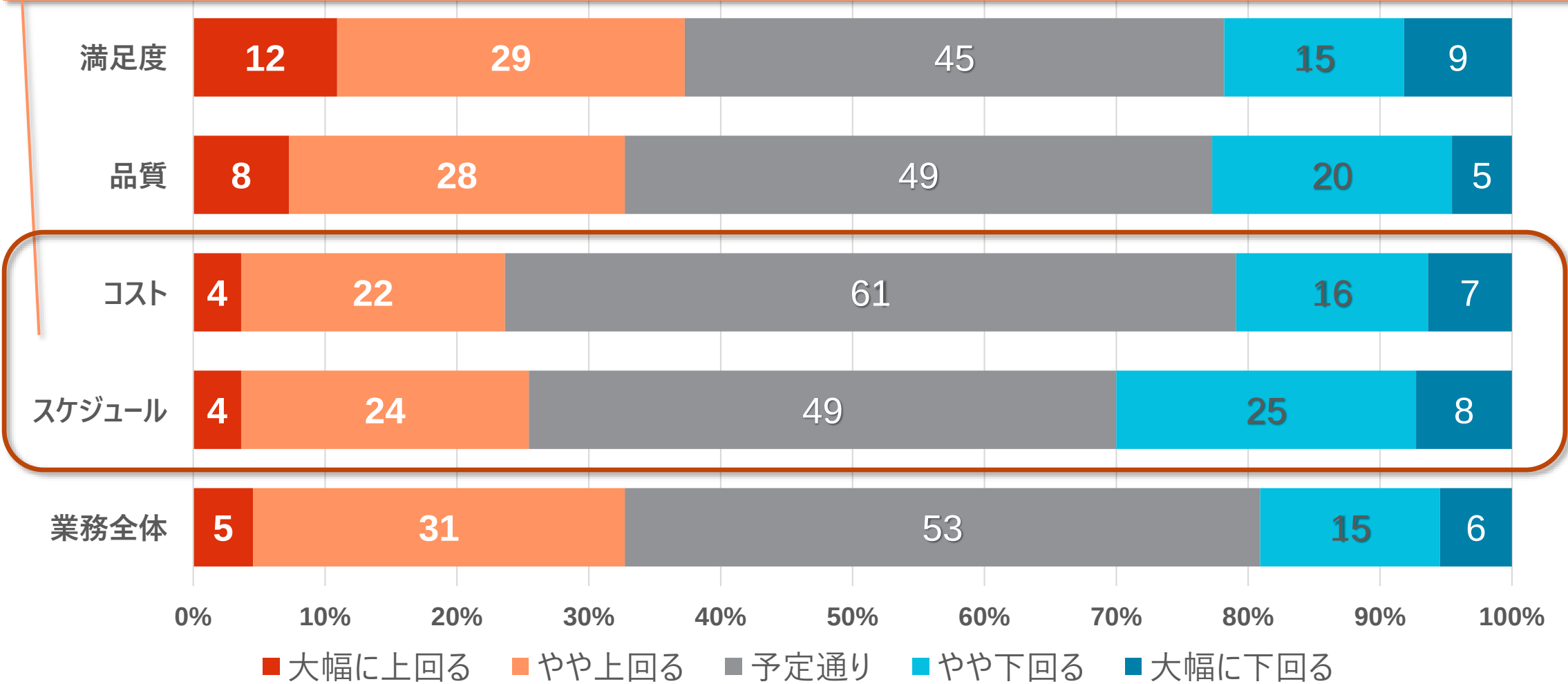
【ソフトウェア開発の場合の契約種類と契約サイクル】

n=90



# 回答者の経験したアジャイル | 成果達成度（当初予定比）

「コスト」「スケジュール」に関して、成果が「下回る」が若干多い

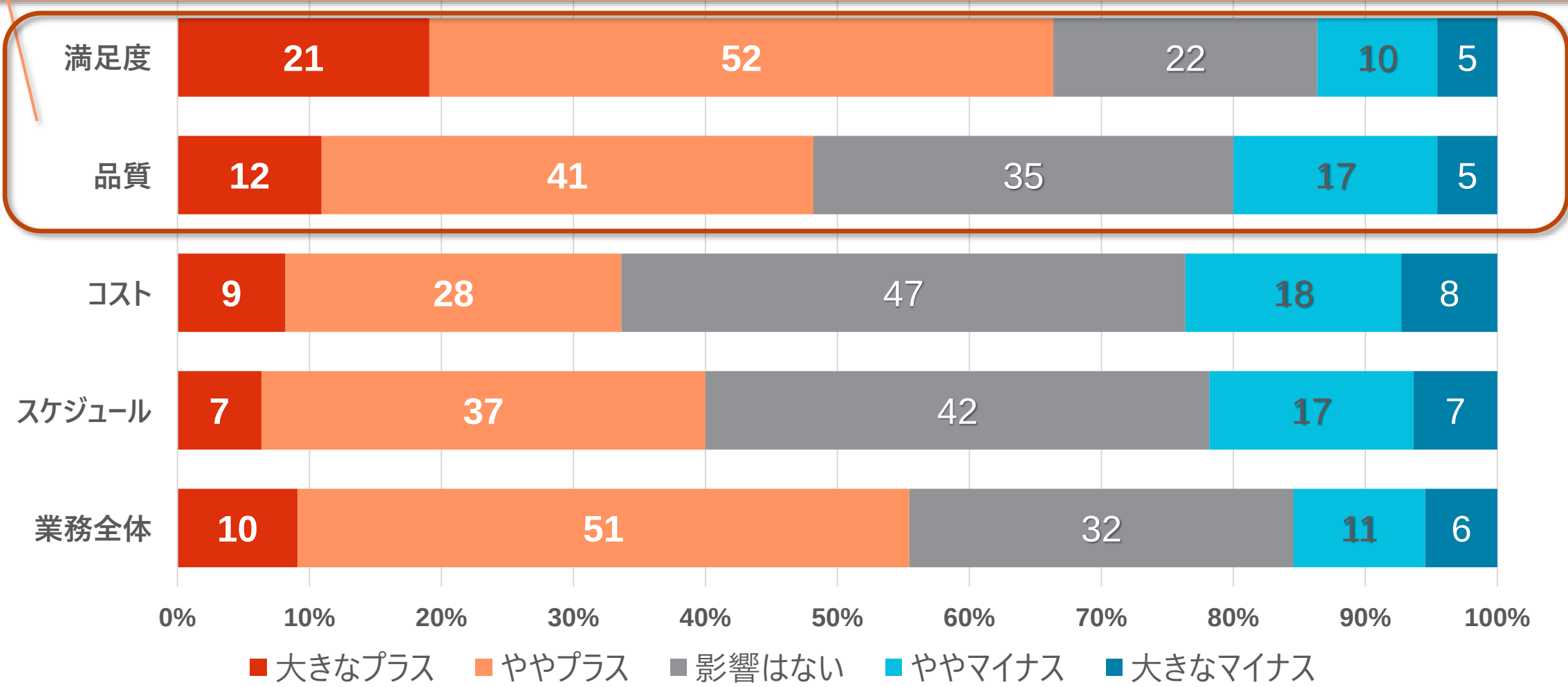


n=110



# 回答者の経験したアジャイル | 成果へのアジャイルの貢献

「満足度」と「品質」について、アジャイルの「プラス」の効果が評価されている



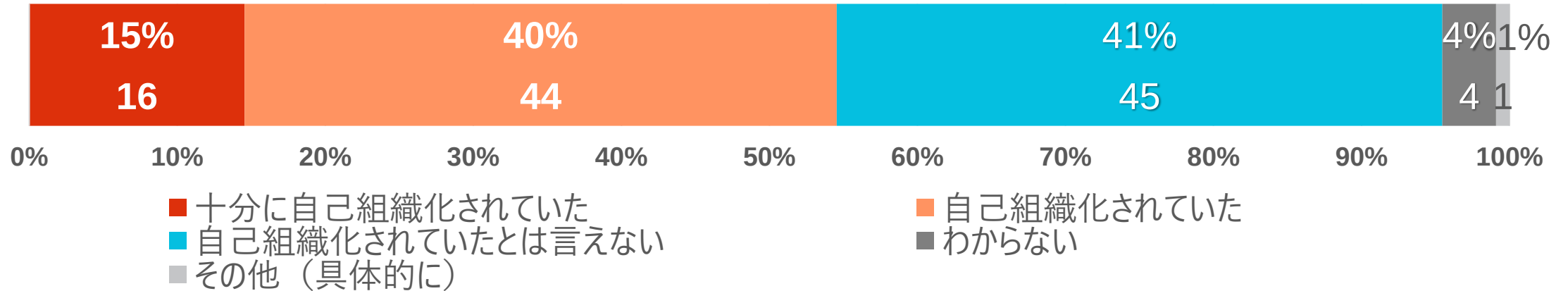
n=110



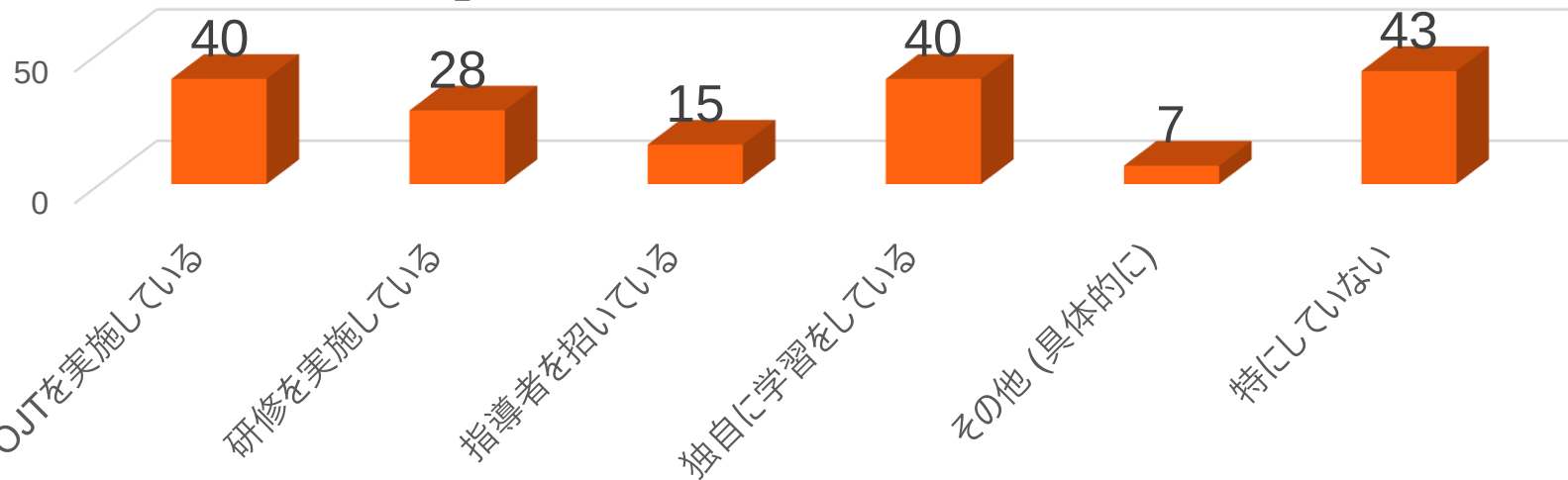
# 回答者の経験したアジャイル | チームの評価と人材育成

## 半数以上は「自己組織化されていた」との回答

【あなたのチームは、与えられた問題と制約の解決を自律的行えたか】



【あなたのチームの人材育成法】



その他(具体的に)

- ・ エキスパート選抜
- ・ チーム内でのクロスファンクシヨンの促進
- ・ 外部研修受講を推奨
- ・ 人として当たり前の育成をする  
と自然にアジャイルになる
- ・ スクラムマスター兼アジャイル  
コーチとして振る舞う
- ・ 学習計画をチームで立案する
- ・ Kickoff時のトレーニング

n=110



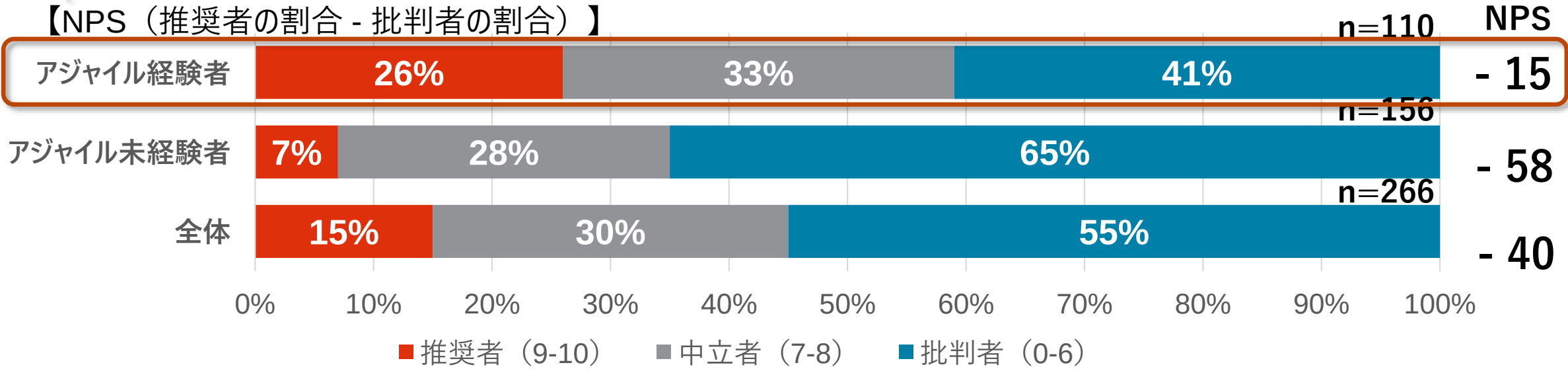
2020年

アジャイル プロジェクト マネジメント意識調査報告＜データ編＞

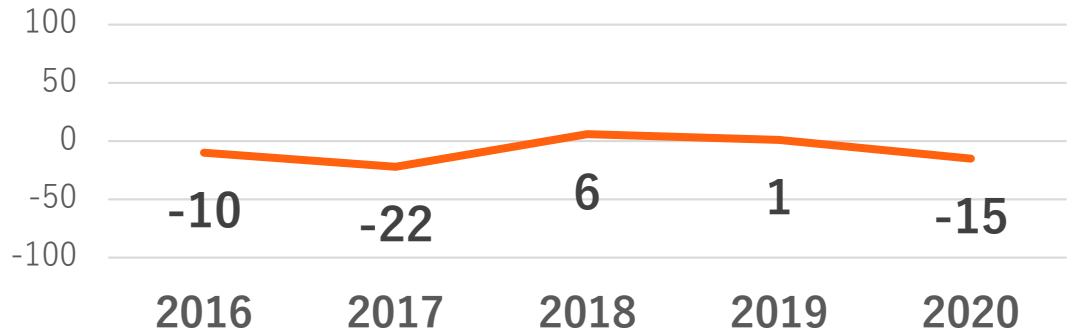
親しい友人や同僚に薦める  
可能性はどのくらいありますか？  
(ネット・プロモーター・スコア)

# 親しい友人や同僚に薦める可能性はどのくらいありますか？

アジャイル経験者でも批判者が大幅に上回り、NPSは-15

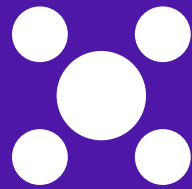


【アジャイル経験者のNPSの推移】



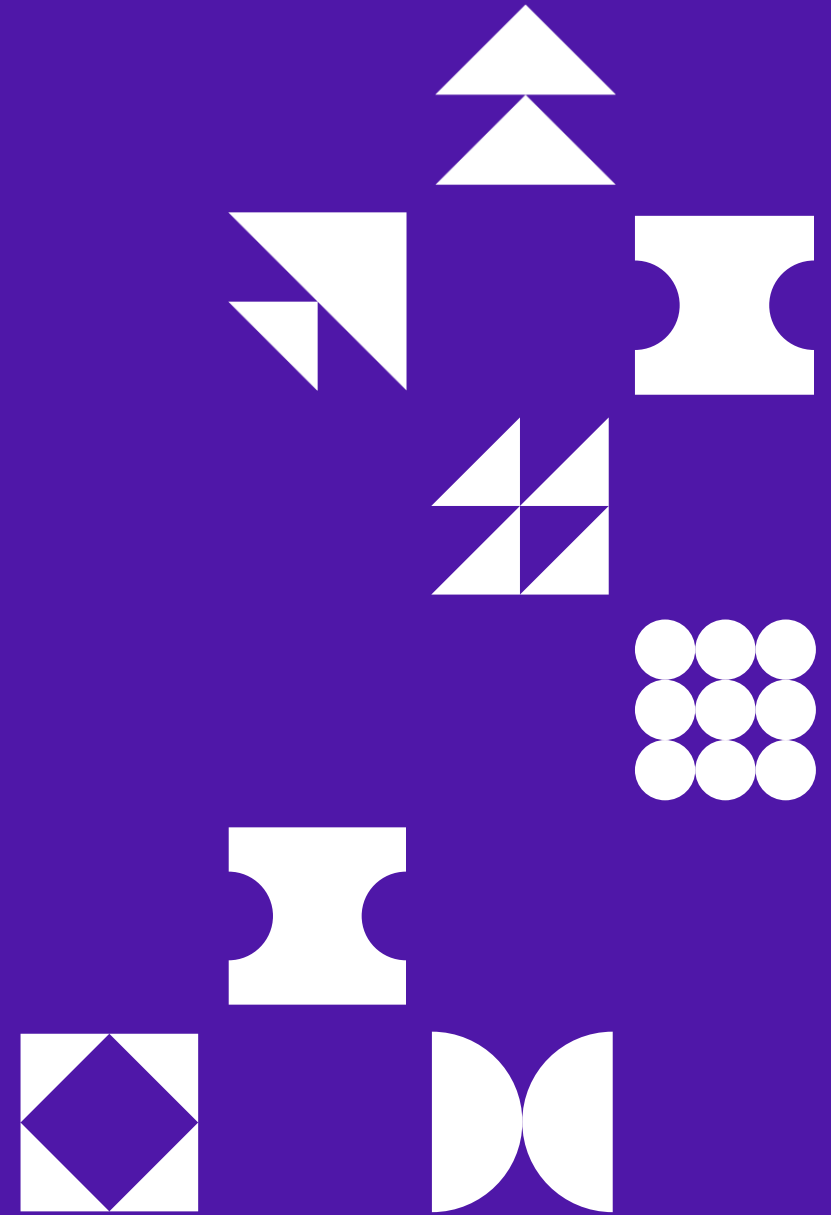
| 分類              | 評価                                                     |
|-----------------|--------------------------------------------------------|
| 推奨者 (Promoter)  | ロイヤルティが高い熱心な顧客。自らが継続購入客であるだけでなく、他者へサービスを勧める『推奨』の役割も担う。 |
| 中立者 (Passive)   | 満足はしているが、それ程熱狂的ではなく、競合他社になびきやすい。                       |
| 批判者 (Detractor) | 劣悪な関係を強いられた不満客。放置しておくと悪評を広める恐れがある。                     |

2020年



# アジャイル プロジェクト マネジメント 意識調査報告 < 分析と提言編 >

~コンテキストに応じた  
多様性のあるアプローチを~



## <分析と提言編>

- ▶ アジャイルはなぜ回答者に推奨されなかったのか？
- ▶ アンケートから見えた課題
- ▶ 多様性のあるアプローチの実現を目指す  
：ディシプリンド・アジャイル・デリバリー（DAD）
- ▶ ライフサイクル
- ▶ プロセスの目標と実践/戦略
- ▶ まとめ

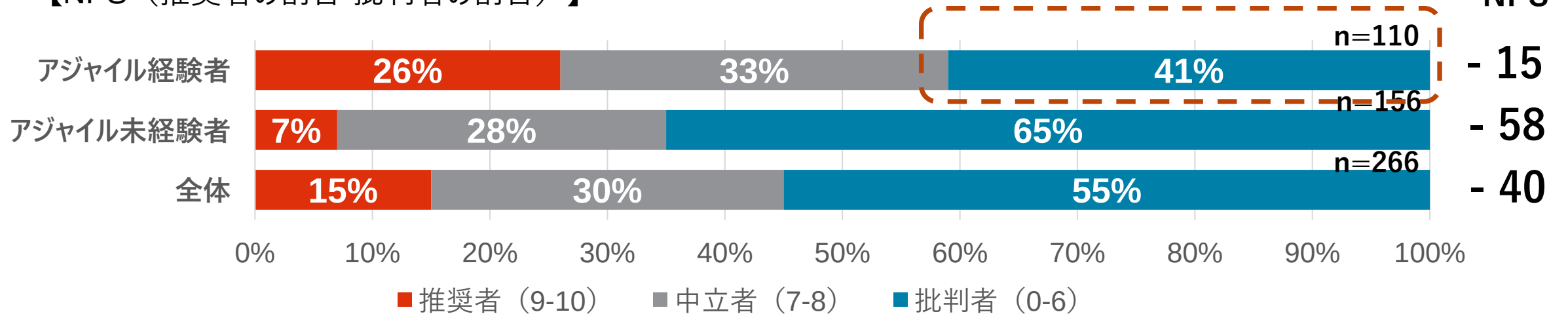


# アジャイルはなぜ回答者に推奨されなかったのか？



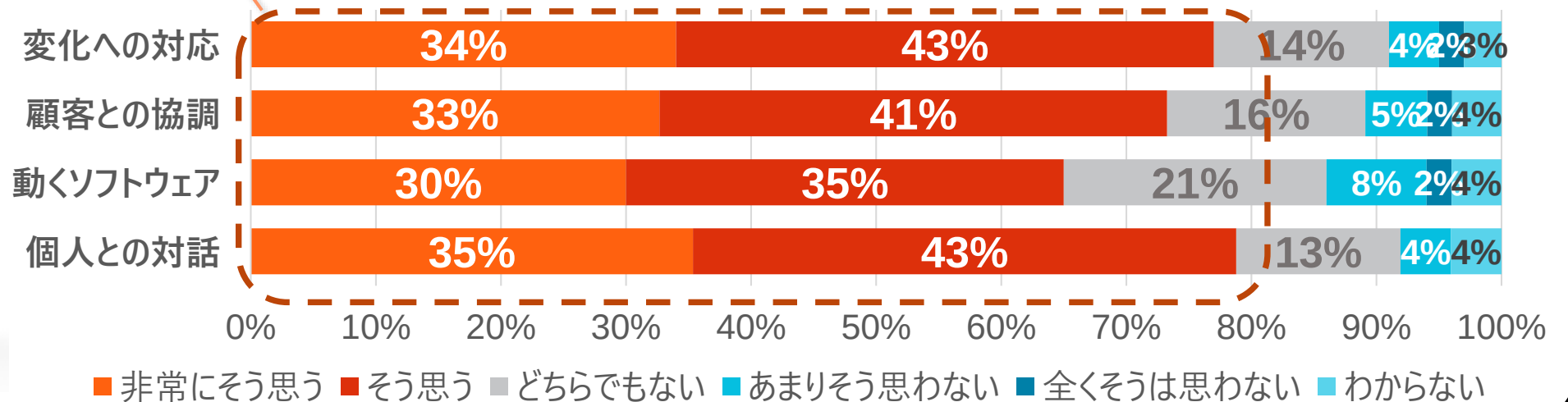
# アジャイル経験者がアジャイルを推奨していない

【NPS（推奨者の割合-批判者の割合）】

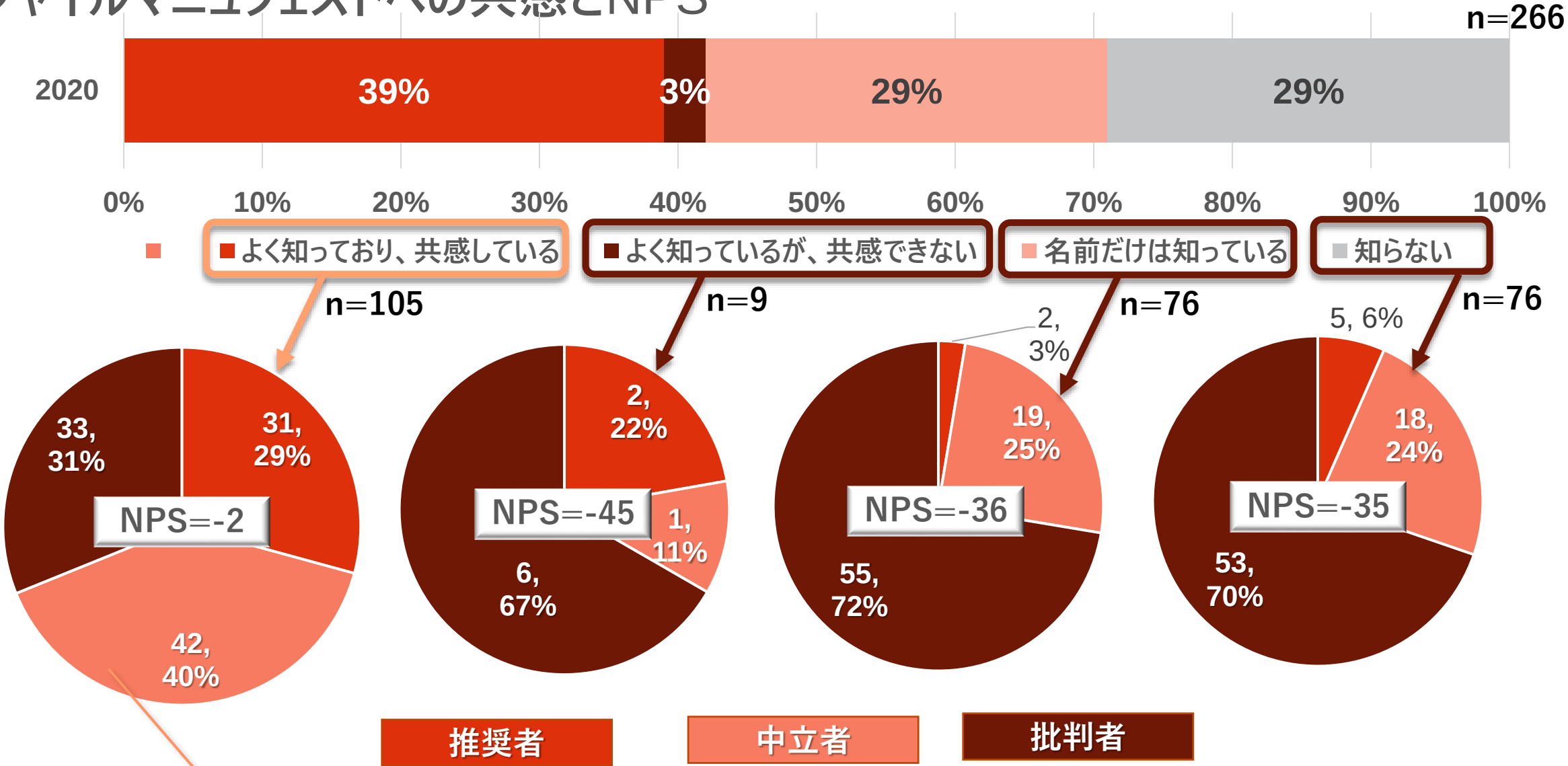


【アジャイルマニフェストへの共感】

アジャイルマニフェストへの共感比率は高い



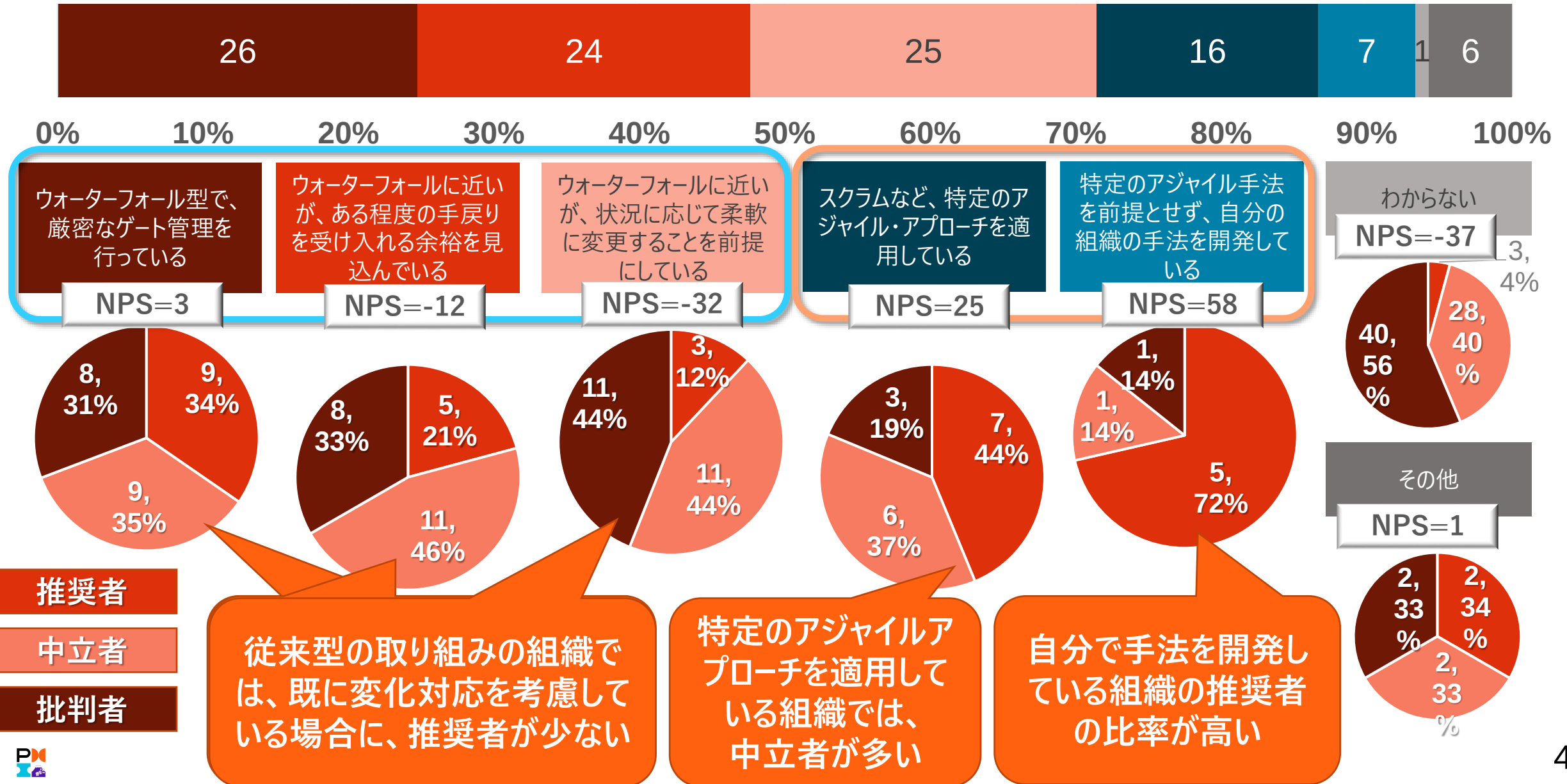
# アジャイルマニフェストへの共感とNPS



アジャイルマニフェストに共感している人の1/3が批判者

# 「よく知っていて共感している人」の組織のプロジェクトへの取組み毎のNPS

n=105



# 「厳密なゲート管理を行っている」の組織のNPSは比較的高い

n=105

26

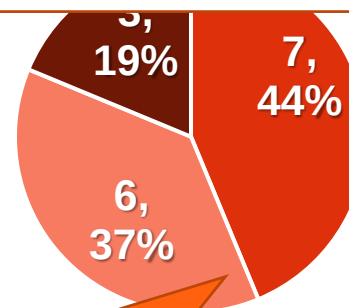
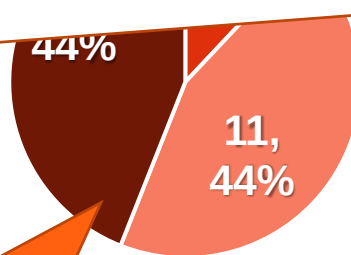
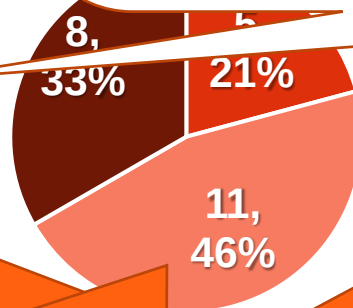
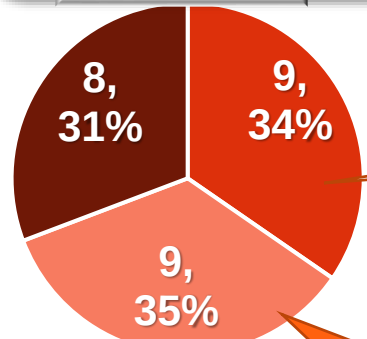
0% 10%

ウォーターフォール型で、  
厳密なゲート管理を  
行っている

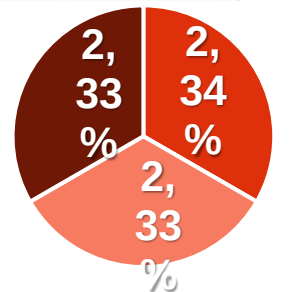
NPS=3

(推奨者)

- アジャイルの考えに基づいた仕事の進め方は、人間の仕事の進め方として自然であり、理にかなっているから。
- ウォーターフォールではイノベーションを起こすのは難しい。ユーザーと密接に連携しながら、ユーザーの期待にこたえられるような製品を作りたい。
- これからのVUCAの時代において、アジャイルの考え方は必須
- 変化に対応できる開発手法が必須となっている。



その他  
NPS=1



推奨者

中立者

批判者

従来型の取り組みの組織では、既に変化対応を考慮している場合に、推奨者が少ない

特定のアジャイルアプローチを適用している組織では、中立者が多い

自分で手法を開発している組織の推奨者の比率が高い

# 「厳密なゲート管理を行っている」の組織のNPSは比較的高い

n=105

26

0% 10%

ウォーターフォール型で、  
厳密なゲート管理を  
行っている

NPS=3

8,  
31%

9,  
35%

推奨者

中立者

批判者

(批判者)

- アジャイルでうまく成果を出すには、前提条件があり、メンバーに経験者（もしくは、有識者）が含まれている必要がある。
- アジャイルによるお客様の価値や認識により単純に薦めるべきではない。
- アジャイルの適用が適しているかは、ケースバイケースであると考えている。
- プロジェクトの内容やその時のアサインメンバー等に大きく影響を受けるため、毎回フラットに判断することがよいと思ったため
- 会社の予算取り方法、リリース前のレビューで、プロジェクトに参加していない上席による承認（規模に応じた段階的な承認行為）が必要な場合、ハイブリッドというより、ウォーターフォール的な要素が強くなるため、開発だけになり、チーム内で不満がたまる。
- 実務経験がないがマニフェスト等考え方に賛同し。メンバーの知恵を結集し共感し前進するアプローチに希望がある

従来型の取り組みの組織では、既に変化対応を考慮している場合に、推奨者が少ない

特定のアジャイルアプローチを適用している組織では、中立者が多い

自分で手法を開発している組織の推奨者の比率が高い

33  
%

2,  
33  
%

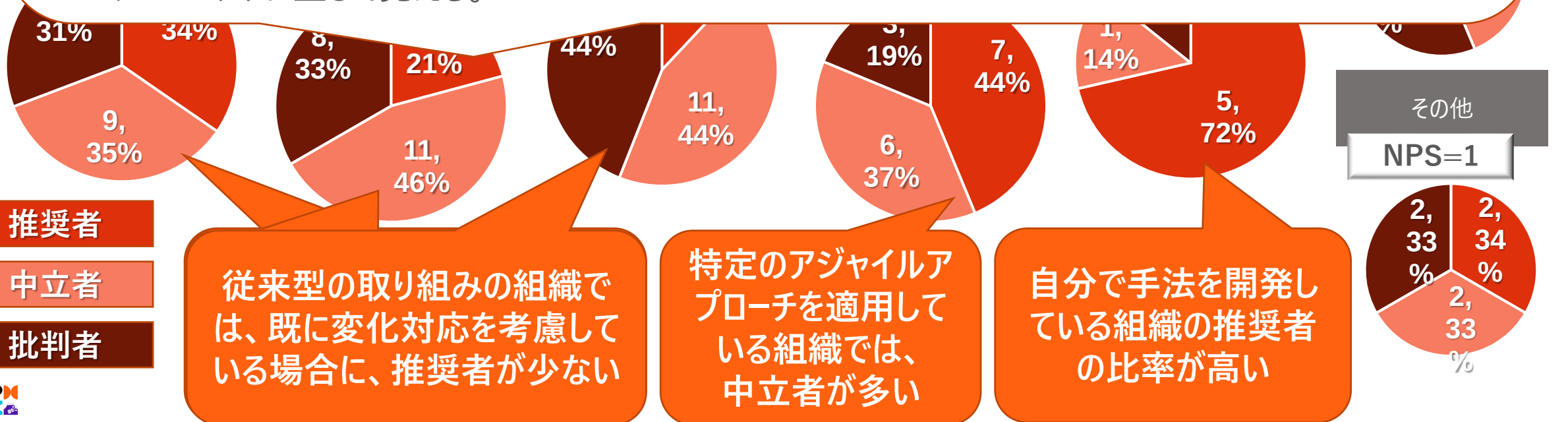
2,  
34  
%

# 「手戻りを受け入れる余裕」の組織のNPSは中立者が増える

n=105

(推奨者)

- アジャイルは協力して新しいものを作るときに必要な、人として当たり前の考え方である。人として仲間を尊重して、工夫し合ってよいものを作る30年前の日本のモノ作りを上手にコンセプト化してくれているので、積極的に活用すべき。一方「誰かの言うことを聞けばよい」とか、「何かに書いてあることをやればよい」とか、「高価な資格を取得すべき」とするような、アジャイルを標榜するものの殆どもニセモノのように思うので、気をつけて取り組むことが必要。
- インフラ開発は減り、UXを大事にする開発が増えるから。
- 探索的要素がある限り、アジャイルでしかあり得ない。予測型プロジェクトの考え方で見れば、ただの炎上プロジェクトに堕して見える。





# 「手戻りを受け入れる余裕」の組織のNPSは中立者が増える

n=105

26

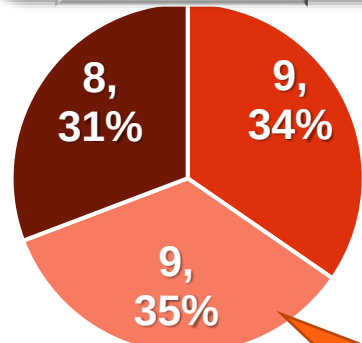
(批判者)

- そのPJの特性による
- 正しいアジャイルの方法に基づいての実施でないと成立しない。ウォーターフォールで行うべき「面倒くさいこと」をやらなくていいという幻想と誤解の元では必ず失敗する。
- 非常に有益な手法だが、スキルセットを持ち合わせたチームビルディングを社内でするのには難しいと思うため
- 友人や同僚にお勧めできるほど、私自身が勉強ができていないため。また、それぞれの事情を考えると、進めるのは難しい。
- 良いと思うが、風土がない

0% 10%

ウォーターフォール型で、  
厳密なゲート管理を  
行っている

NPS=3



推奨者

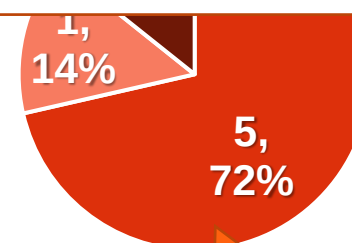
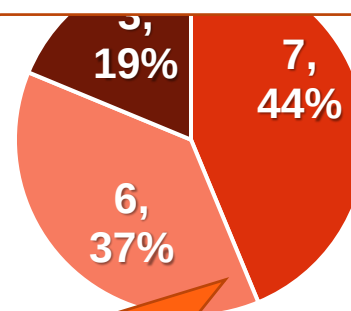
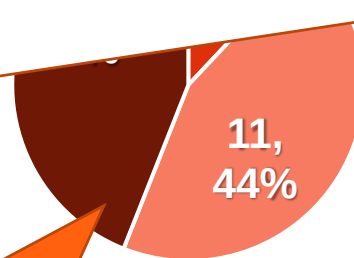
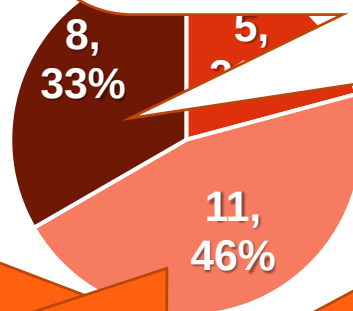
中立者

批判者

従来型の取り組みの組織では、既に変化対応を考慮している場合に、推奨者が少ない

特定のアジャイルアプローチを適用している組織では、中立者が多い

自分で手法を開発している組織の推奨者の比率が高い



その他

NPS=1

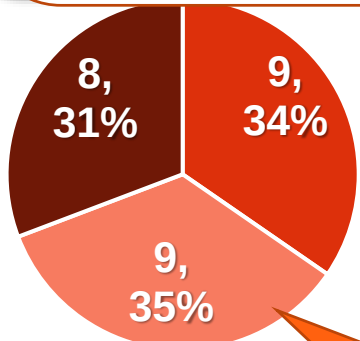


# 「状況に応じて柔軟に変更」の組織では、推奨者が少なくなる

n=105

(批判者)

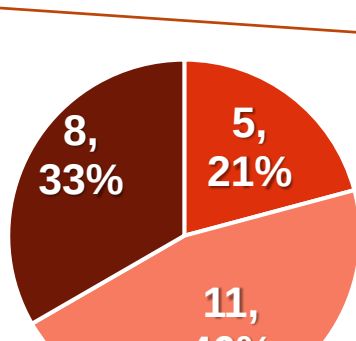
- ・ アジャイルが有効となるのはケースバイケースと考えているため。
- ・ アジャイル開発のマインドセットを顧客と開発側の両方が理解していることが重要なため、薦めるには幾つかのクリアすべき条件があるため。
- ・ テーマが適していて、然るべき人材が確保されているという稀な条件が成立していれば勧める。
- ・ 試す価値はあるので
- ・ 全員がプロセスを頭に入れる必要がありそこだけ



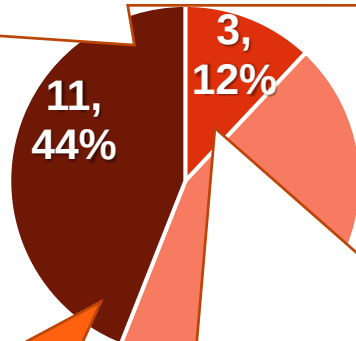
推奨者

中立者

批判者



従来型の取り組みは、既に変化対応をしている場合に、推奨者



(推奨者)

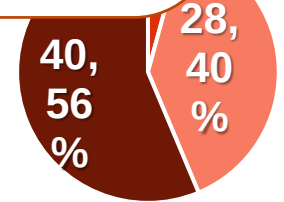
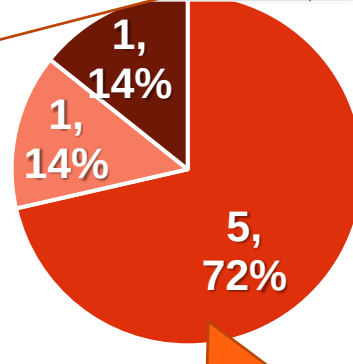
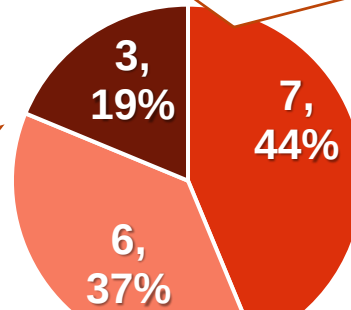
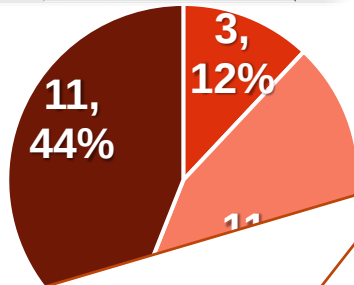
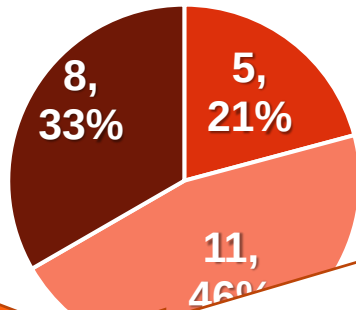
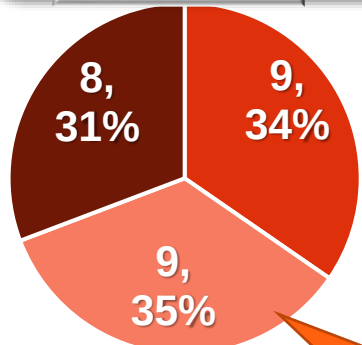
- ・ 真にユーザが求める製品を開発するためには、変化に対応できるアジャイルが適していると感じているため。

現在のソフトウェア開発が、「柔軟に変更」できているので、困っていない。

# 「特定のアジャイル・アプローチ」の組織は推奨者も中立者も多い

## (推奨者)

- ウォーターフォールでできる自信があるプロジェクト以外、アジャイルを薦める
- スクラム以外の手法の有効性が感じられない。ウォーターフォールのメリットは一つもない。どんな大規模、規制があっても。適切にスクラムが運用できれば、問題ない。
- ビジネスアジリティという分野を開拓中で、そのコアはアジャイルの思想であるため
- 一度はやってるみるべきだと思うため
- 単なる指示待ちにならず、優先順位に基づいて自分たちで作業を計画/実施するようになるから。



その他

NPS=1

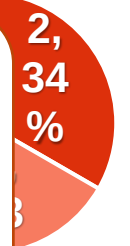
推奨者

中立者

批判者

## (批判者)

- ビジネスサイドの人間のかかわり方や巻き込みが弱いとプロジェクト運営が円滑に回らず、継続的に協力してもらう為の働きかけに負荷がかかるため
- 友人や同僚の開発プロジェクトによるのと、自分が人に薦められるほど、アジャイル開発を理解できていないため。



# 「自分の組織の手法を開発している」の組織のNPSは、抜群に高い

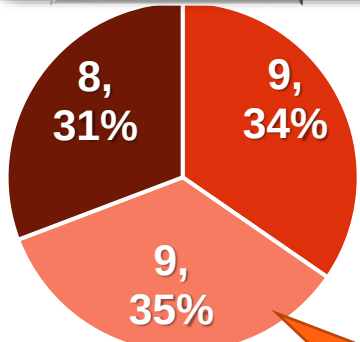
n=105

(推奨者)

- アジャイルマニフェストの価値観に完全に反する人に出会ったことがないため
- いいチームで、いいサービス・プロダクトを作るための一番の近道だと考えているので（他の手法でもできなくはないけど、より大変）
- 当たり前のことだけど、それを否定する人がもし居れば、お勧めする。

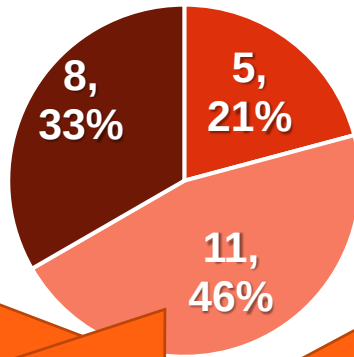
厳密なワーク管理を行っている

NPS=3



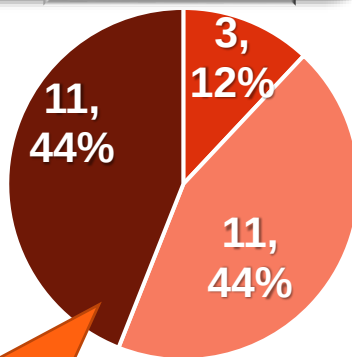
を受け入れる余裕を見込んでいる

NPS=-12



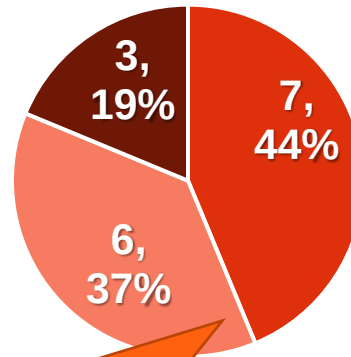
に変更することを前提にしている

NPS=-32



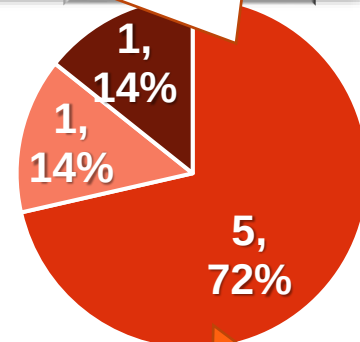
ファイル用いている

NPS=25

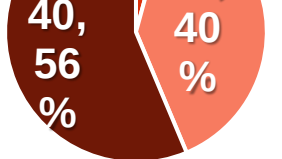


発して

NPS=8

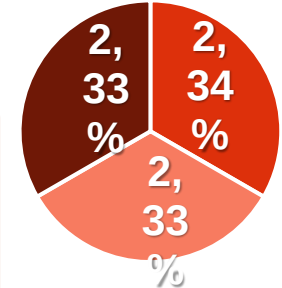


NPS=-37



その他

NPS=1



推奨者

中立者

批判者

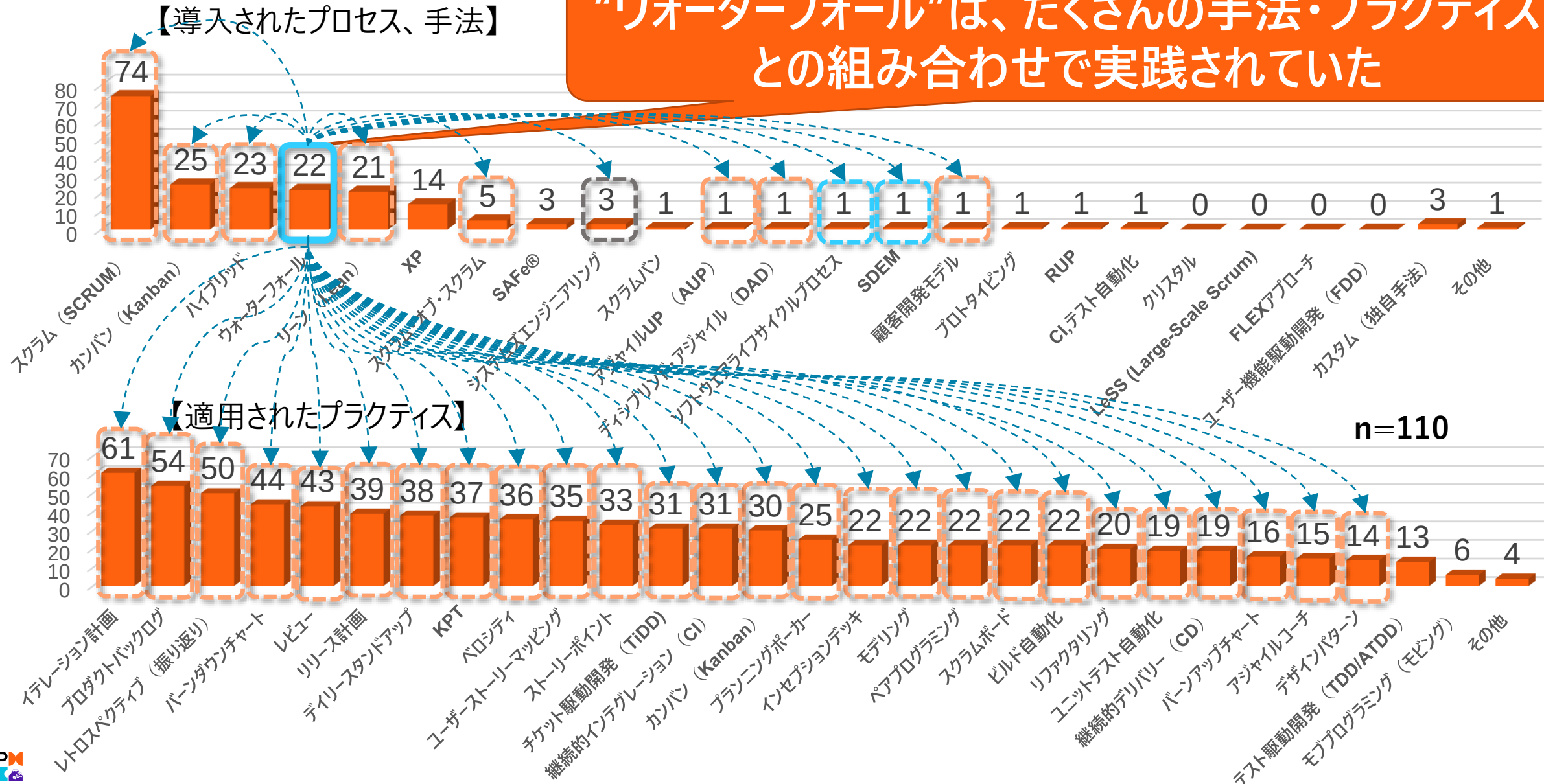
従来型の取り組みの組織では、既に変化対応を考慮している場合に、推奨者が少ない

特定のアジャイルアプローチを適用している組織では、中立者が多い

自分で手法を開発している組織の推奨者の比率が高い

# アジャイル対象業務のプロファイル | 導入されたプロセス・手法、プラクティス

“ウォーターフォール”は、たくさんの手法・プラクティスとの組み合わせで実践されていた



## 推奨者の理由（全体）

### アジャイルマニフェスト の考え方を薦める

- エンジニアの能力が向上するので薦める (3)
- 小さい単位でフィードバックを受けたいので薦める
- 顧客中心の方法論なので薦める (2)
- 変化に対応できる開発手法なので薦める (3)
- 当たり前のことから薦める (3)
- 一度はやってみるべきだと思うので薦める
- スクラムだけが有効なので薦める
- ビジネスアジリティのコアはアジャイルなので薦める

### アジャイルソフトウェア開発宣言

プロセスやツールよりも**個人と対話**を、

包括的なドキュメントよりも**動くソフトウェア**を、

契約交渉よりも**顧客との協調**を、

計画に従うことよりも**変化への対応**を、



## 批判者の理由（全体）

- 開発担当のスキルに問題あり (5)
- ウォーターフォールとの比較による
- ステークホルダーに問題あり (4)
- メンバーの負荷が高い
- 実現可能性は低い
- 現実的とは思えない。
- 内部統制の枠組みが崩壊
- 正しくないアジャイルが実施される
- プロジェクトに参加していない  
上席による承認が必要
- 薦めるのはケースバイケース (4)

アジャイルマニフェストを否定しているのではなく、その「実践」に課題を感じている

アジャイルプラ  
クティスは、  
「銀の弾丸で  
はない」  
ということ

アジャイルソフトウェア開発宣言

プロセスやツールよりも**個人と対話**を、  
包括的なドキュメントよりも**動くソフトウェア**を、  
契約交渉よりも**顧客との協調**を、  
計画に従うことよりも**変化への対応**を、

薦める場合と、  
薦めない場合がある

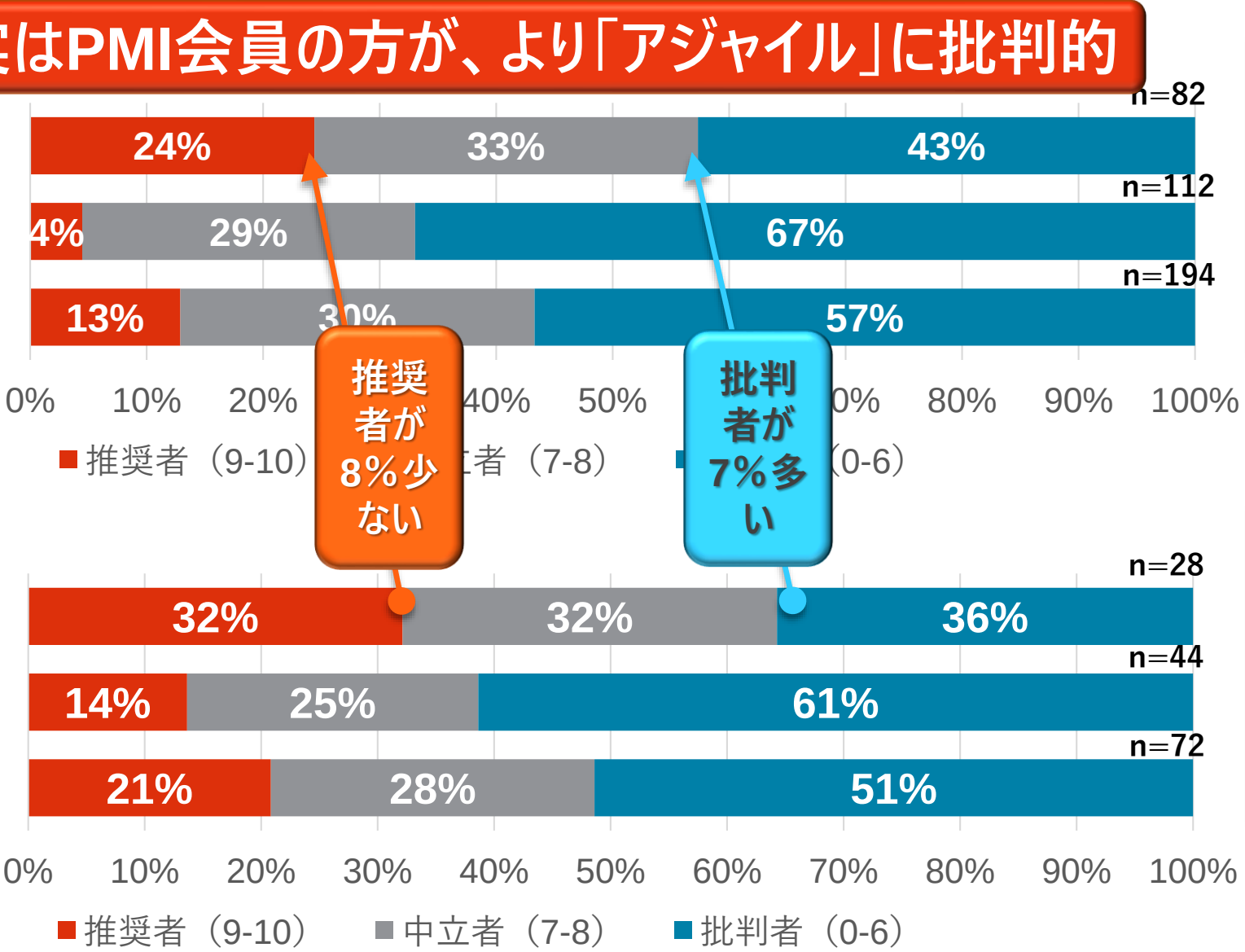


NPS（推奨者の割合 - 批判者の割合）（PMI会員と会員以外）

【PMI 会員】 **実はPMI会員の方が、より「アジャイル」に批判的**

「実践」の  
課題を  
より感じて  
いるのかも

【PMI 会員以外】



PMI 会員 (n=194)

NPS

- 19

- 63

- 44

PMI 会員以外 (n=266)

NPS

- 15

- 58

- 40

会員以外 (n=72)

NPS

- 4

- 47

- 30

閑話休題……ところで皆さん、本当にウォーターフォールですか？

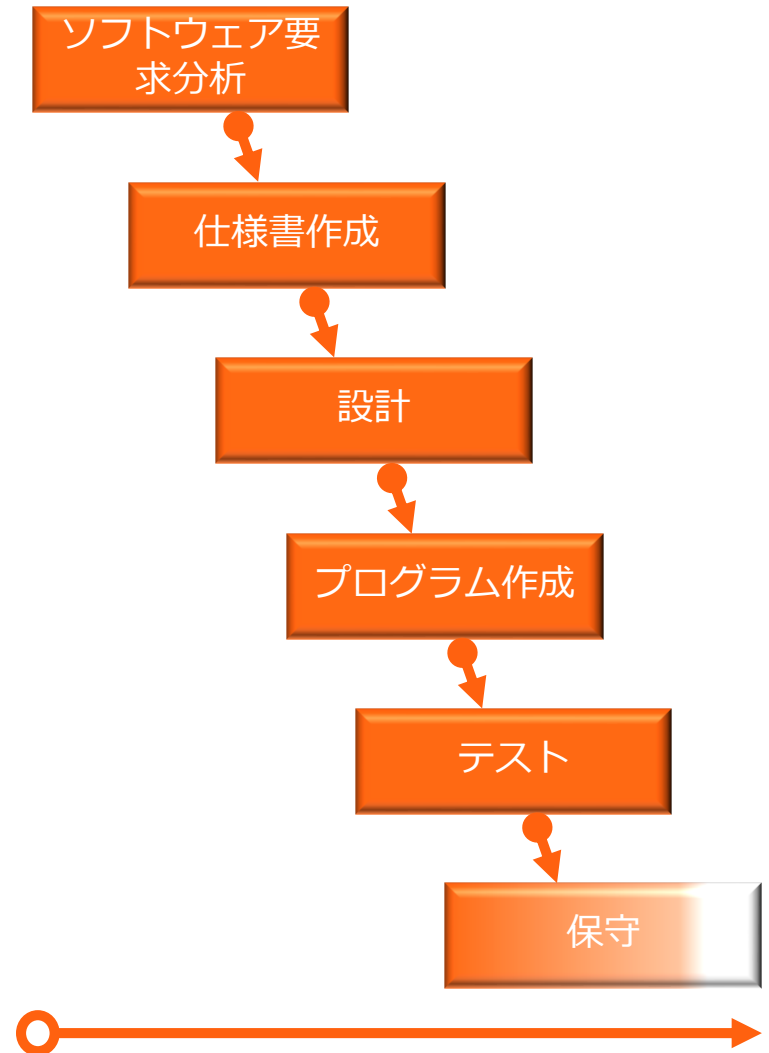




# 1988年の教科書には、こんな風を書いてありました。：ウォーターフォールモデル

## 「ウォーターフォール」とか「ビッグバンテスト」は やってはいけないと教わった

- ✓ このモデルは、単純な性格を持つ小規模なシステムが大勢を占めていた時代に開発された・・・（中略）。したがって、このモデルは、小規模なシステムの場合に比べて開発サイクルの上流工程に未知の部分がはるかに多い大規模システムの開発に合うように拡張するのが難しい。
- ✓ このモデルは、コンピュータによる支援技術がまだ未発達の時点で開発されたため、手作業を基本としており、コンピュータの利用には適さない。
- ✓ 単純なシステムの作成に向くこのモデルを大規模なシステムの開発用に拡張しようとすれば、メリットよりもデメリットの方が大きくなる可能性が高い。



# 今の教科書にも、こんな風に書いてあります：ウォーターフォールモデル

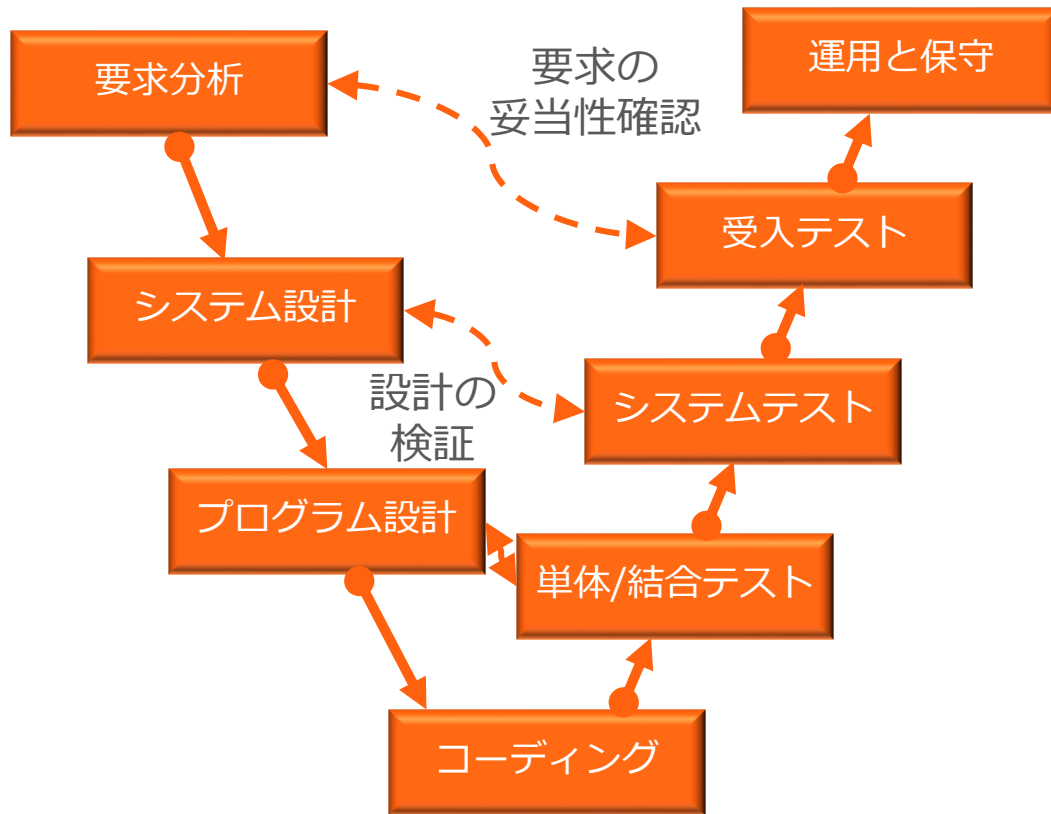
## 共通フレーム2013

- ・ ウォータフォールモデルは、 それぞれの作業ステップを一回だけ通して行うというのが特徴である。...  
このモデルを適用する際のリスクは次のようなものがある。
    - ・ 要求の定義が不十分であるおそれ
    - ・ すべての作業を一回の実施だけで済ませるには、システムの規模が大きすぎるおそれ
    - ・ 予想している技術が急に变化してしまうおそれ
    - ・ 要求が急に变化してしまうおそれ
    - ・ 現時点では、要員又は資金が限られているおそれ
    - ・ 製品をその開発の途中で動かしてみることができないおそれ
- 大規模開発に向かない、変化に弱く、要求定義が不十分になりやすい開発手順

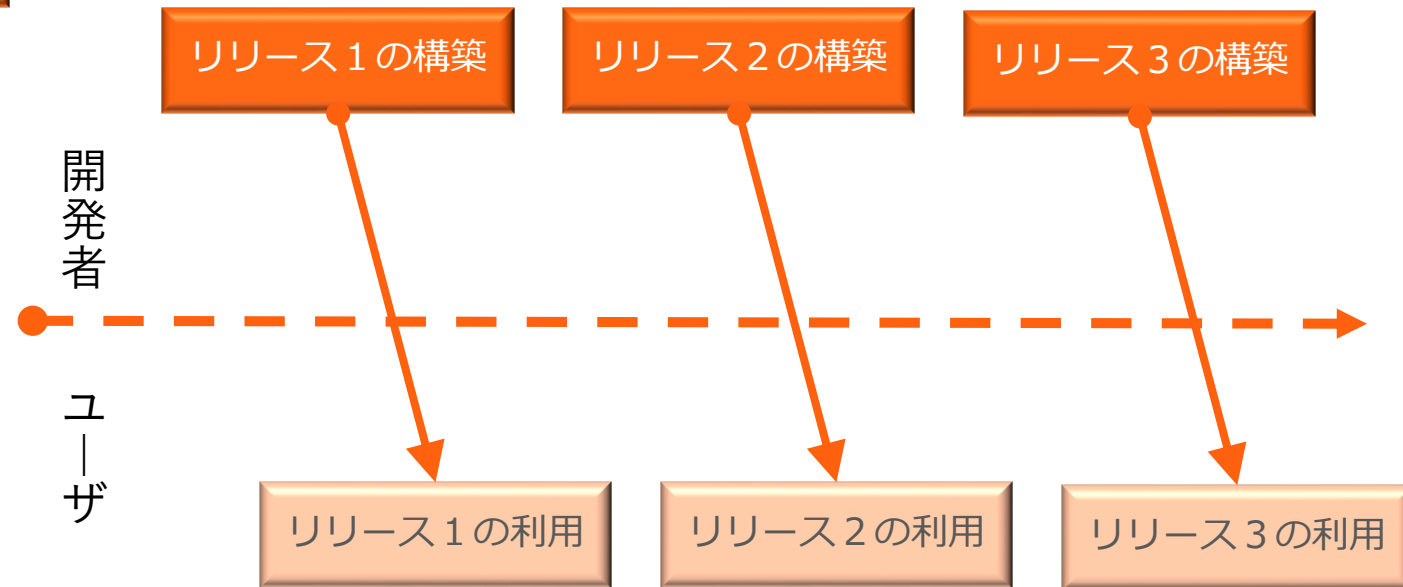
# 皆さん、本当にウォーターフォールですか？

皆さんの開発モデルは、もっと大きく複雑で、こんな要素も入っているのでは...

Vモデル



段階的开发モデル



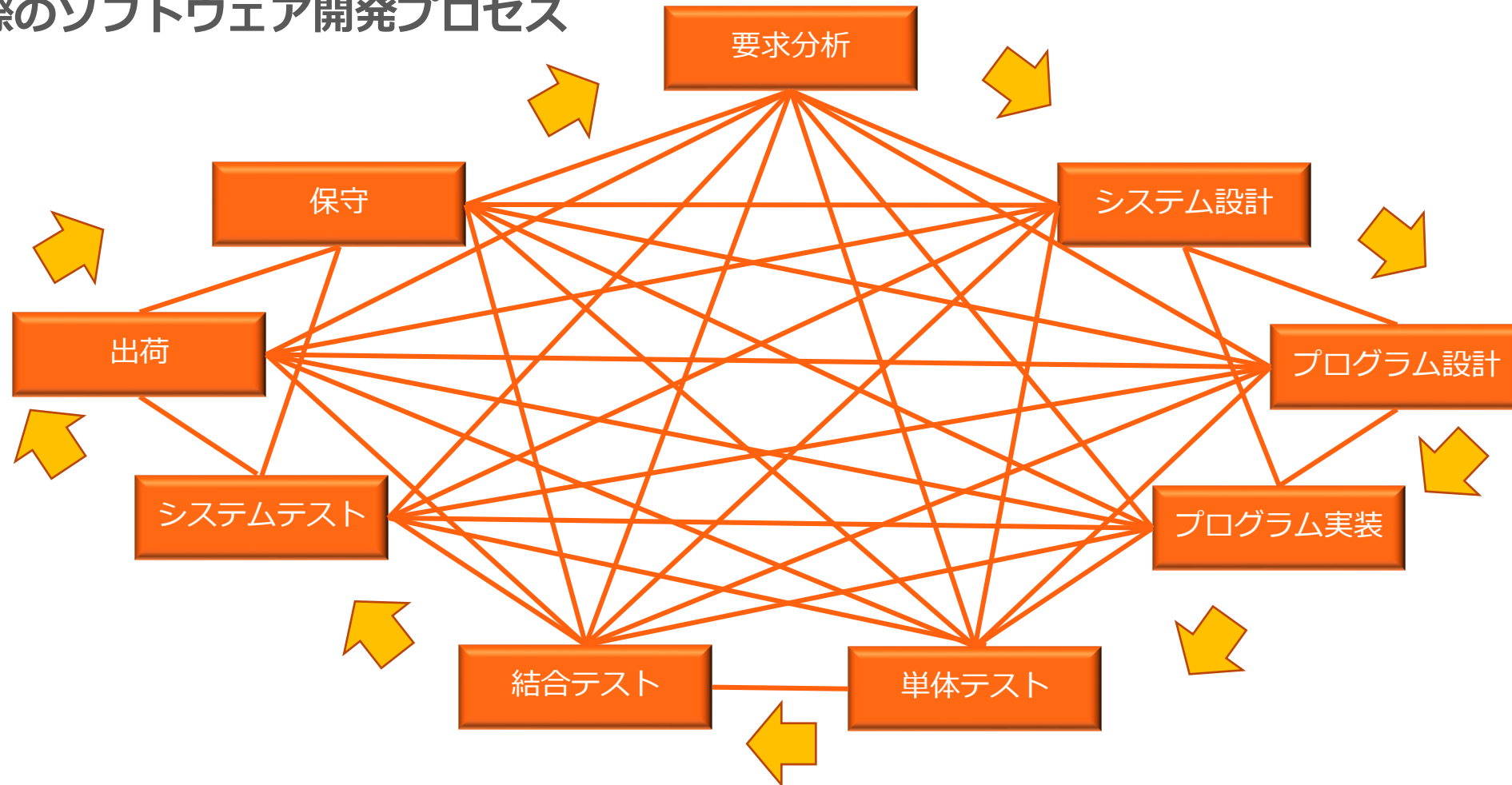
伝統的手法による実際のプロジェクトのイメージ：この「滝」には枝分かれも合流も還流もある



実際の開発では、この複雑な組み合わせの、多数の組み合わせになる

「ソフトウェア開発」はそもそも複雑、これをどう工夫して進めるかが課題

実際のソフトウェア開発プロセス



# ウォーターフォールの起源：1970, Dr. Winston W. Royce

## MANAGING THE DEVELOPMENT OF LARGE SOFTWARE

大規模ソフトウェア開発の管理についての私の個人的見解を説明する。私は過去9年間に様々な任務を経験してきたが、その大部分はミッションの計画、指揮および飛行後解析のため等の宇宙船のソフトウェアパッケージ開発に関係している。これらの任務において、私は運用状態に到達することに関して、時間通りに、そしてコストの範囲内という点で、成功とは異なる経験をした。私はこの自分の経験に対し、**改善すべき意見**をもっており、これを発表する。

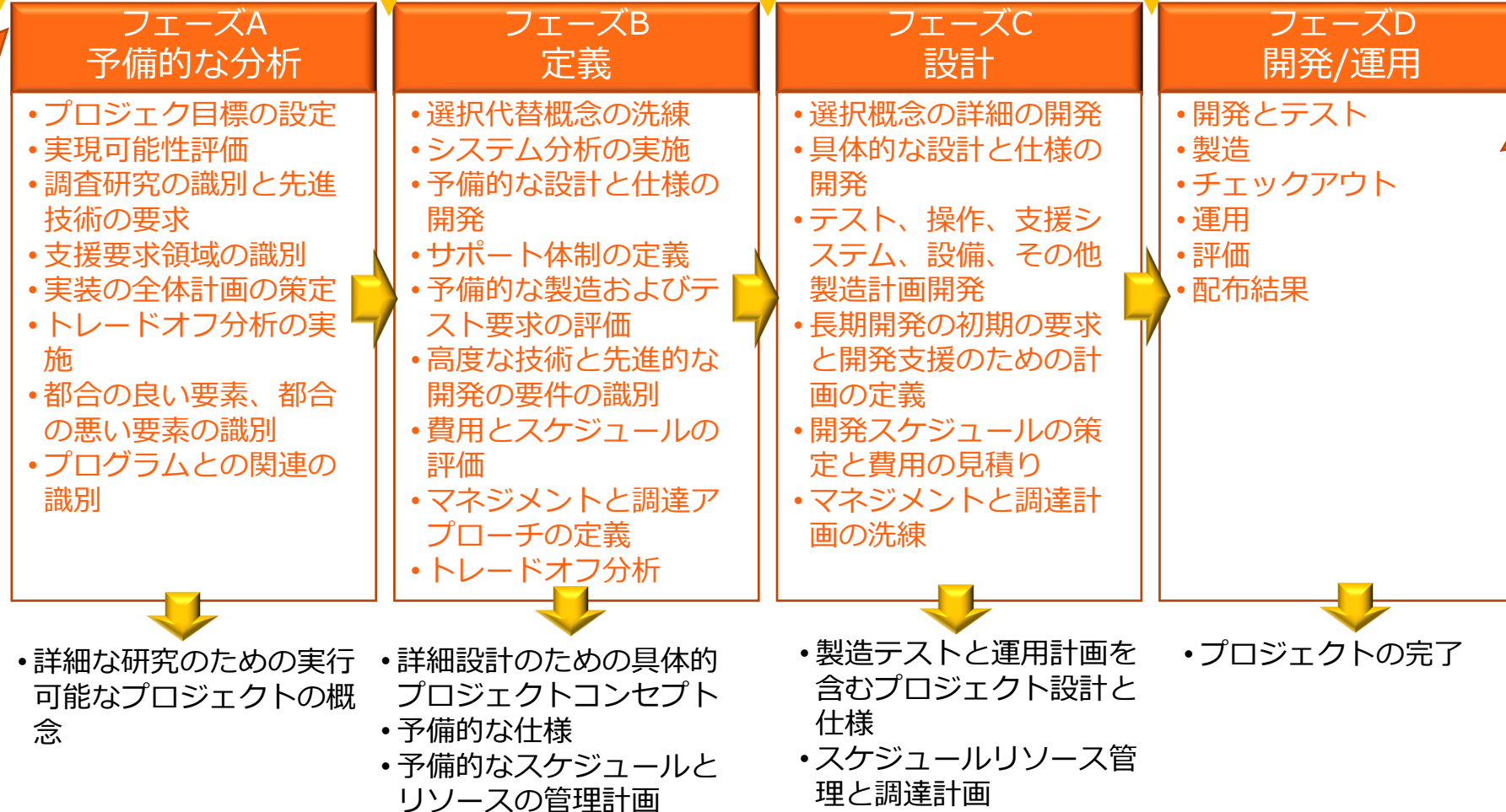
# PHASED PROJECT PLANNING GUIDELINES

1968, NASA

Royceがうまくいかないと指摘した当時のNASAの開発標準

主要なマネジメント判断

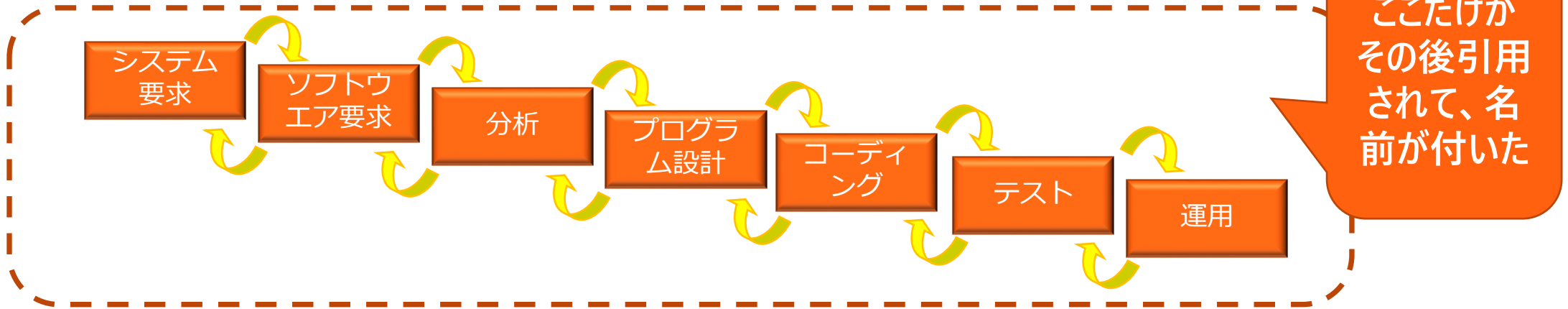
「フェーズド  
アプローチ」といっ  
ていた



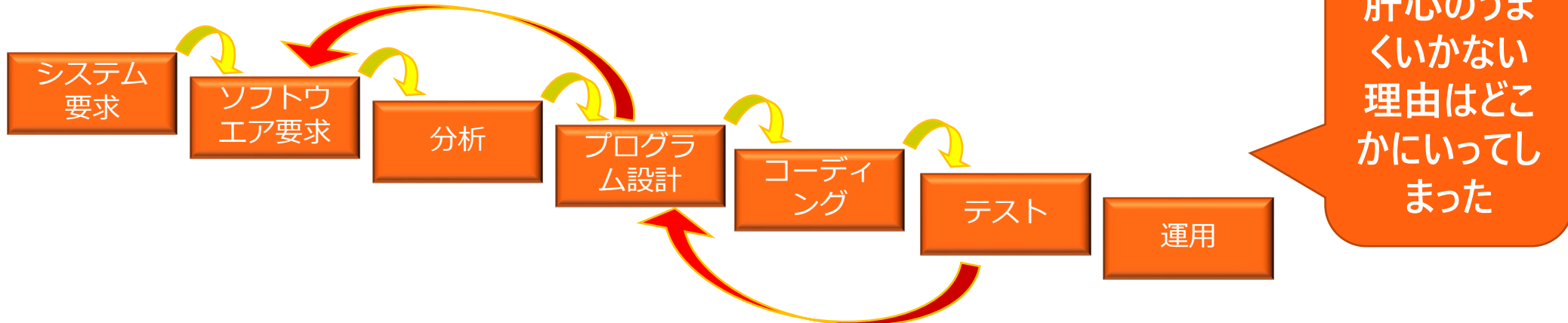
分析・定義・設計を  
これだけ丁寧  
にやっても前の  
フェーズに  
戻ってしまう

# 1970, Dr. Winston W. Royce

さまざまな段階の間の反復的な相互作用は連続したステップに限定されることが期待されている

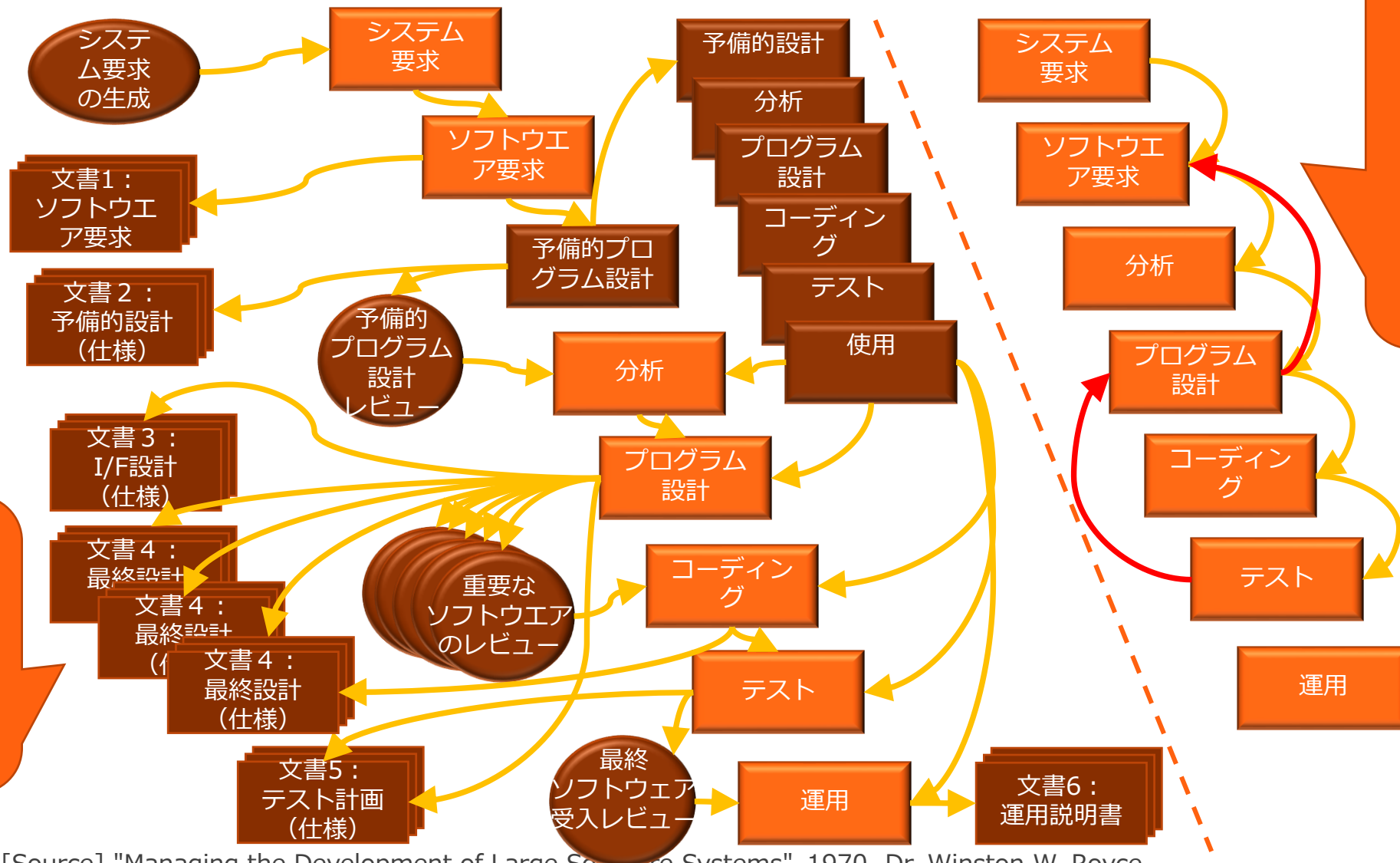


残念ながら、図のプロセスに示すように反復は連続するステップに限定されることはない





# 1970, Dr. Winston W. Royce



Royceの提案モデルは、「一度試しに作ってみてから」開発するというもの

厳密なドキュメンテーションとレビューを要求

## Royceさんが改善すべきと提案した対象（ダメなやつ）を指す

### 5.4 Life cycle concepts - 5.4.2 Life cycle model for the software system (仮訳)

- ソフトウェア開発段階の1つとしてよく引用されているのは、引き出し、要件、設計、構築、およびテスト-予測モデルまたは「ウォーターフォール」モデルです。段階が逐次的であると考えるので、各段階は次の段階に進む前に正しい結果を生み出すことを要求されます。実際には、要件がよくわかっていて初期コスト見積もりが正確でない限り、これを達成するのは非常に困難です。**ウォーターフォールを実行する際に、計画された段階のいずれにも適切に収まらないような大規模なリワークを実行するリスクがあるため、おそらく予算に収まらないでしょう。**

注1：ライフサイクルプロセスモデルの初期の分析者として一般的に知られているWinston Royceは、「ウォーターフォール」（彼が使用しなかった用語）よりも改良のための修正段階の必要性を説明しました。残念ながら、改良のための修正段階は一般の理解の中では「ウォーターフォール」モデルから削除されています。

こんな本があります：1990年 アジャイルは、まだなかった頃です

# Wicked Problems, Righteous Solutions

A Catalogue of Modern Software Engineering Paradigms

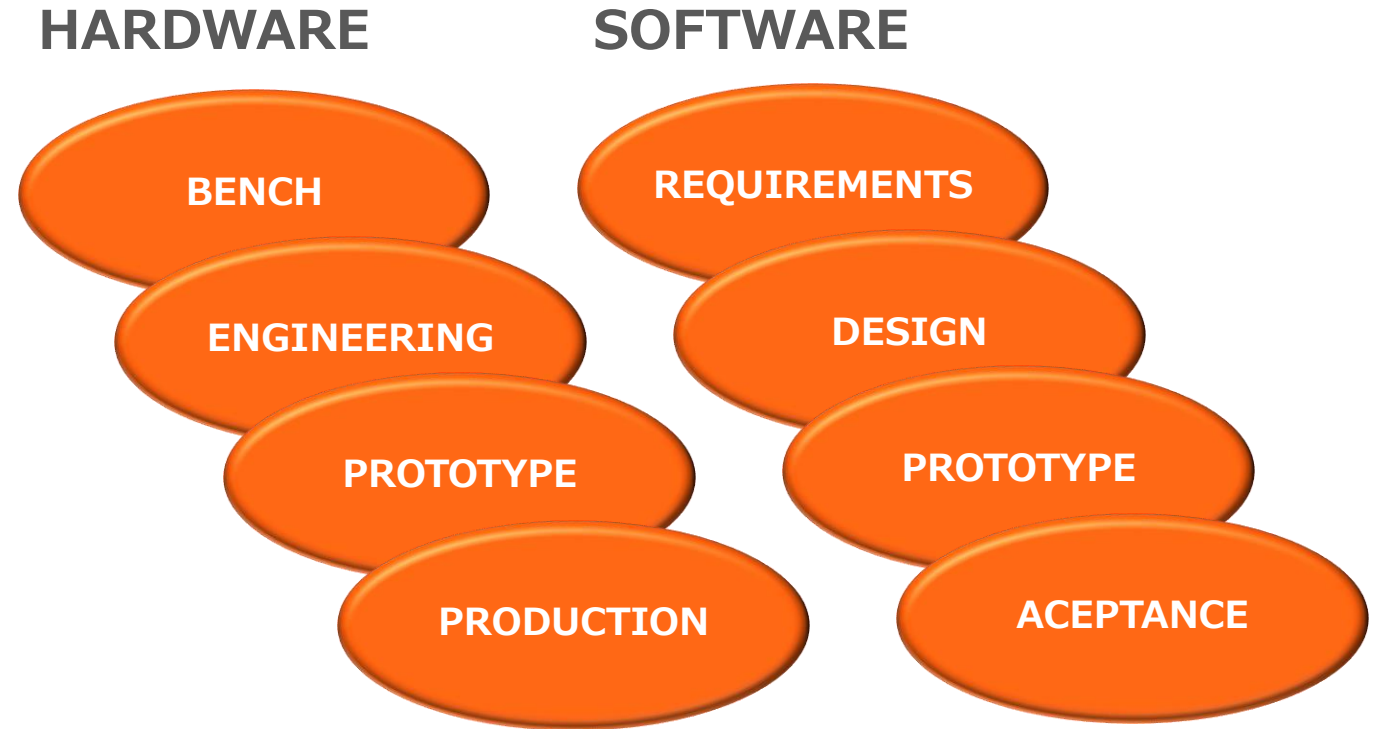
Peter DeGrace and Leslie Hulet Stahl

1990, Prentice-Hall, Inc.

- CHAPTER SEVEN The ALL-at-Once Model
  - Team Approaches --- Sashimi and Scrum

# サシミとスクラム

日本人は早い段階でウォーターフォールモデルを採用し、問題の多くを発見した。さらに、製品開発においては、速度と柔軟性が従来の高品質、低コスト、製品の差別化と同じくらい重要であるとされた。彼らはそれを自分のスタイルに適応させ、ステップ数を4に減らしたが、アクティビティは削除しなかった。4つのフェーズはオーバーラップされ、彼らはそれをサシミと呼び、重なり合うフェーズ間に密接なコミュニケーションと統合、密接な協力を意図している。



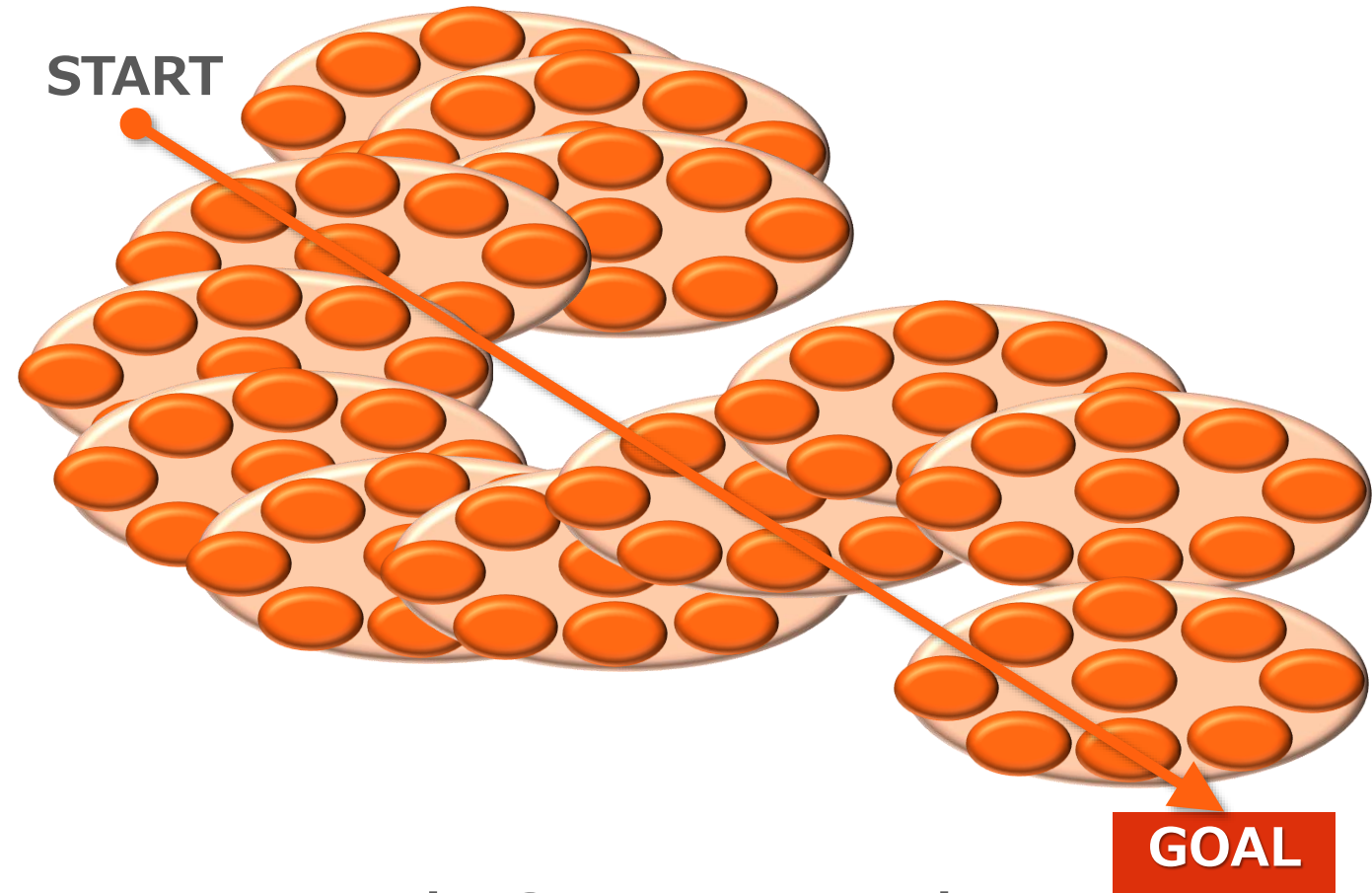
**The Sashimi Approach**

# サシミとスクラム

アジャイルが生まれる前、日本はウォーターフォールの課題を既に解決していた

他の企業はさらに一歩進め、歩数を4から1に減らし、それにラグビーから名前を付けた。ラグビーでは、チームの全員が他のみんなと一緒に「ボールをフィールドに着ける」ために行動する。このチーム運営はスクラムと呼ばれる。

しばらくすると魔法が起こる。最初はほとんど目立たないが、すぐにチームは動き始める。それぞれの作業の議論について他のすべての人がそれを学び、それに貢献し始める。チームのメンバーの仕事は、いつでも全員の力で行われる。



The Scrum Approach

この柔軟さを、何故失ってしまったのでしょうか？





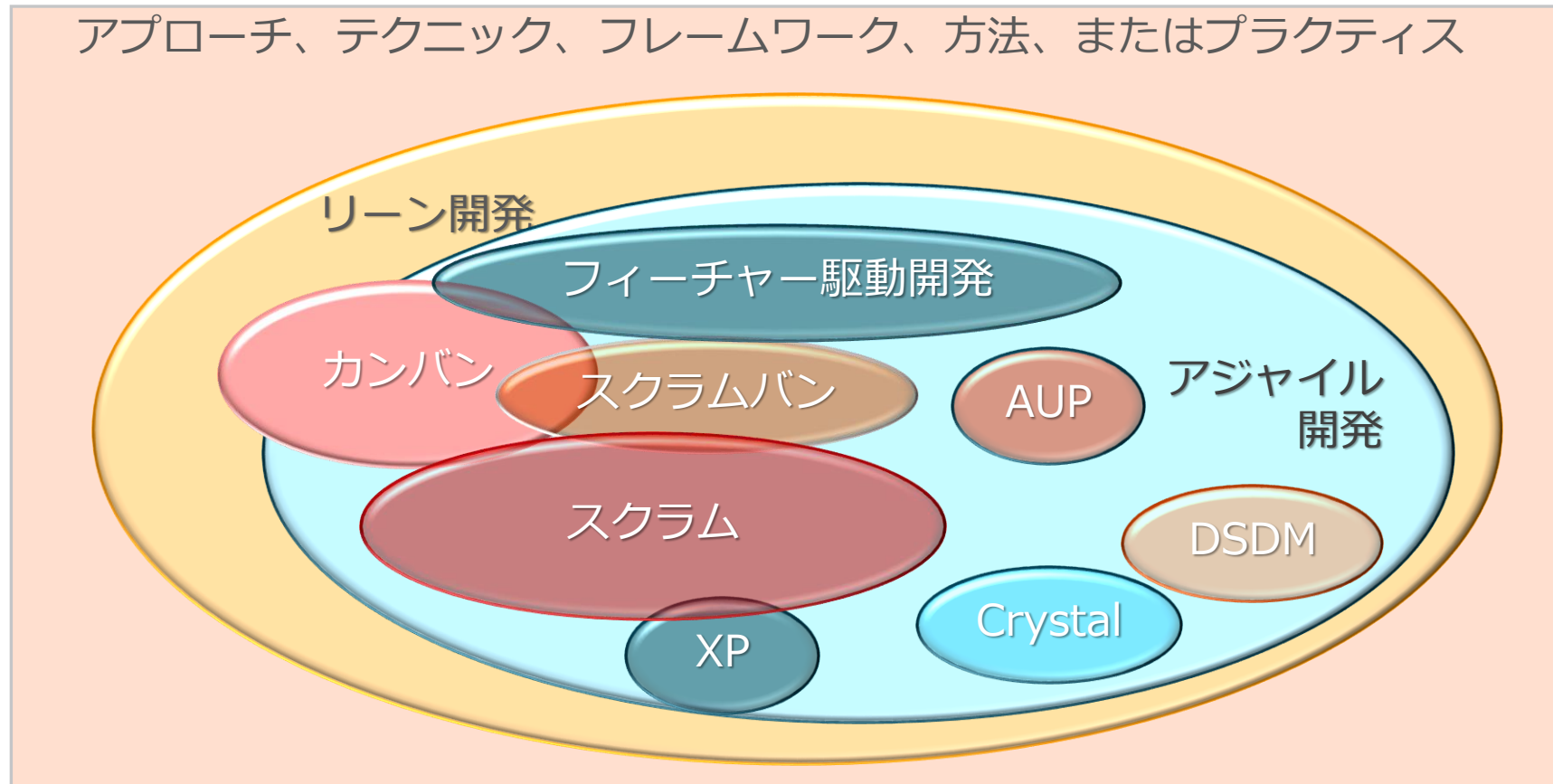
# アンケートから見た課題の整理



# アジャイル等、手法の「多様性」の理解

Agileの持ち味は、  
「多様性」

アジャイル開発は、さまざまなフレームワークとメソッドをカバーする包括的な用語





共通しているのは...

分かりやすいが、  
最大公約数

## アジャイルソフトウェア開発宣言

私たちは、ソフトウェア開発の実践あるいは実践を手助けをする活動を通じて、よりよい開発方法を見つけだそうとしている。この活動を通して、私たちは以下の価値に至った。

プロセスやツールよりも**個人と対話**を、  
包括的なドキュメントよりも**動くソフトウェア**を、  
契約交渉よりも**顧客との協調**を、  
計画に従うことよりも**変化への対応**を、

価値とする。すなわち、左記のことがらに価値があることを認めながらも、私たちは右記のことがらにより価値をおく。

© 2001, 上記の著者たち

この宣言は、この注意書きも含めた形で**全文を含めることを条件に自由にコピーしてよい。**

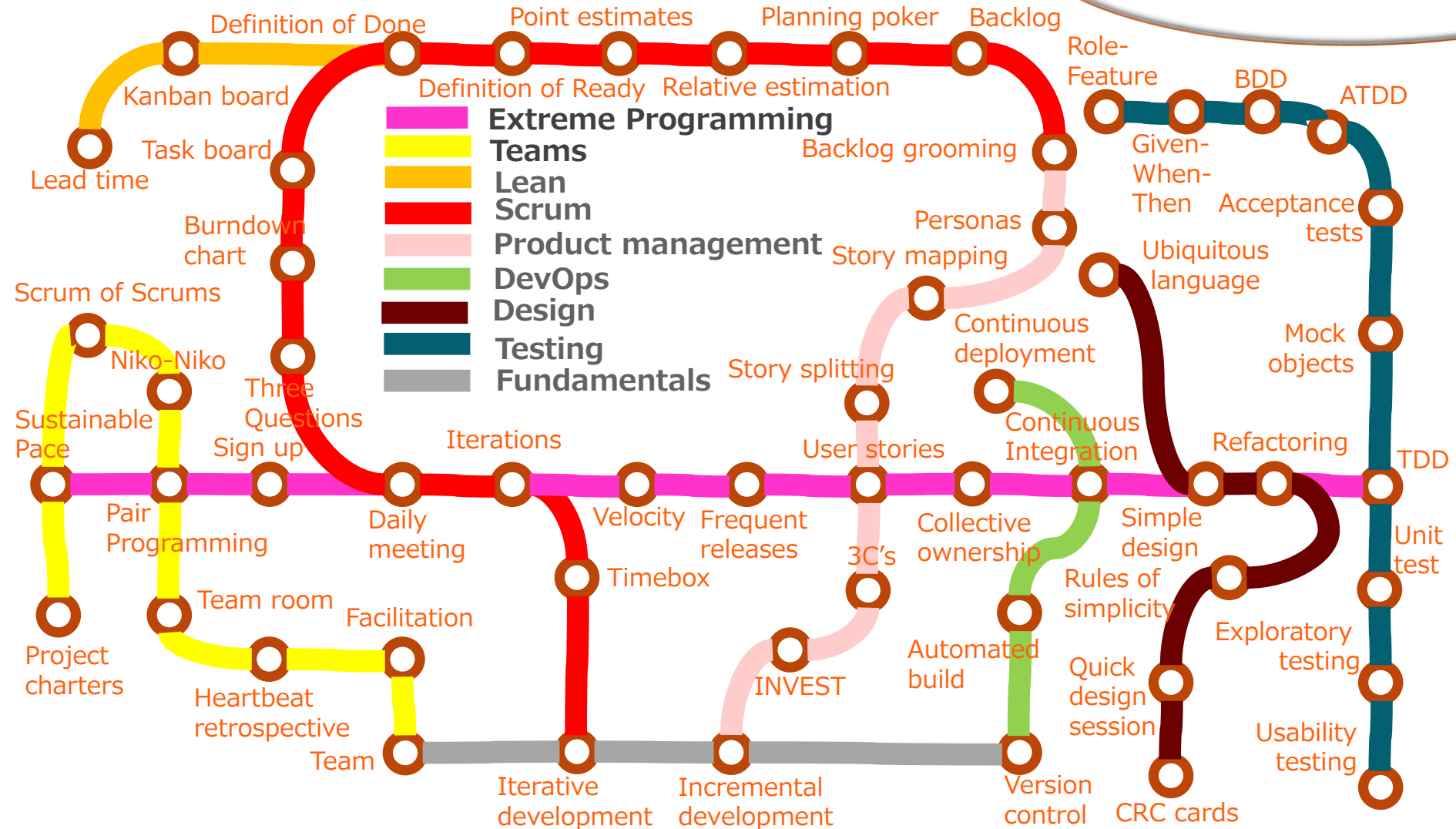
[Source] Kent Beck他17名の著者たち, 2001, <http://agilemanifesto.org/iso/ja/manifesto.html>  
2017/09/18 access)

## そしてたくさんのプラクティス

# Subway Map to Agile Practice

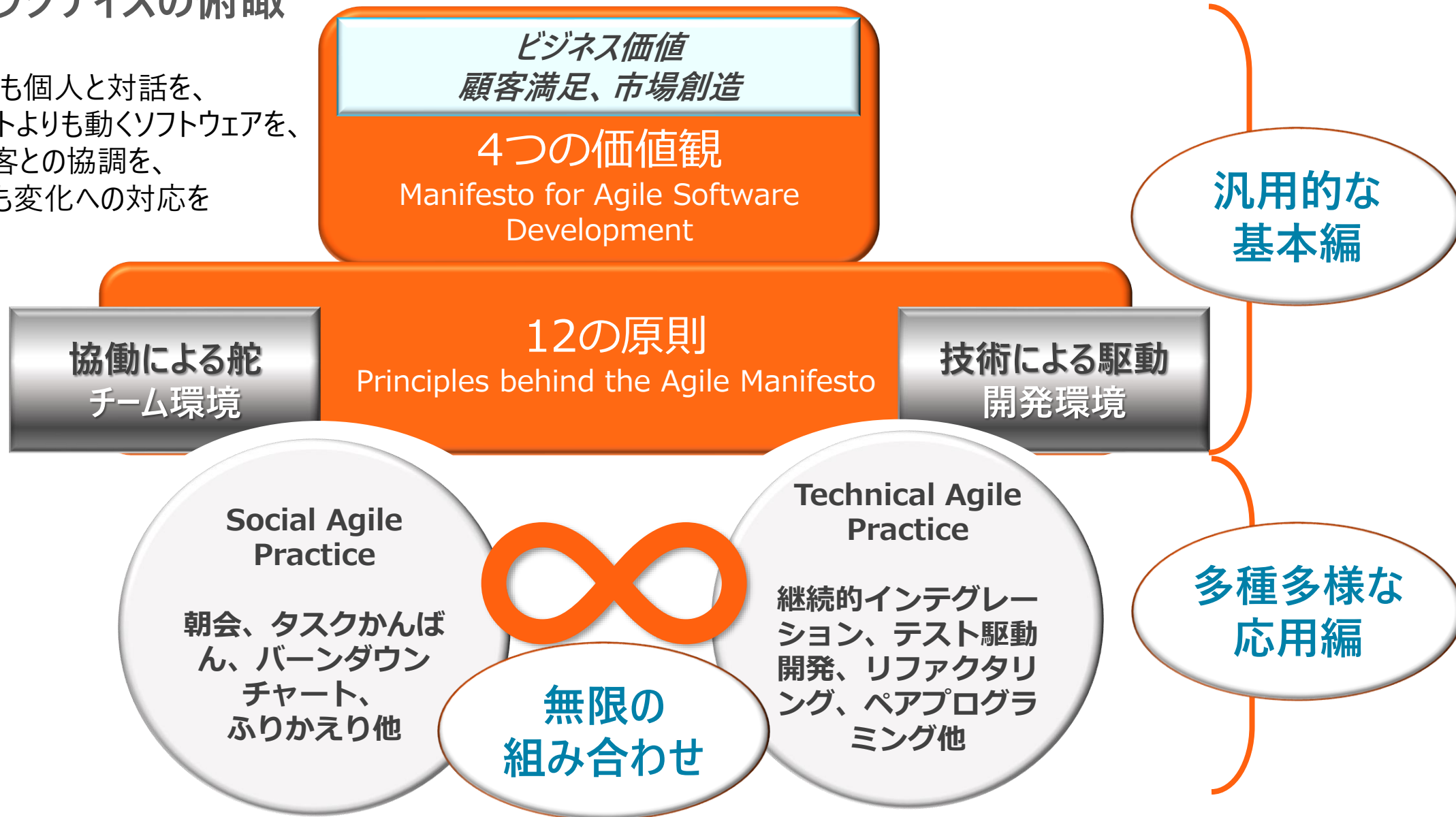
## この多様性を 楽しめるか

Lines represent practices from the various Agile “tribes” or an



# アジャイルプラクティスの俯瞰

プロセスやツールよりも個人と対話を、  
包括的なドキュメントよりも動くソフトウェアを、  
契約交渉よりも顧客との協調を、  
計画に従うことよりも変化への対応を

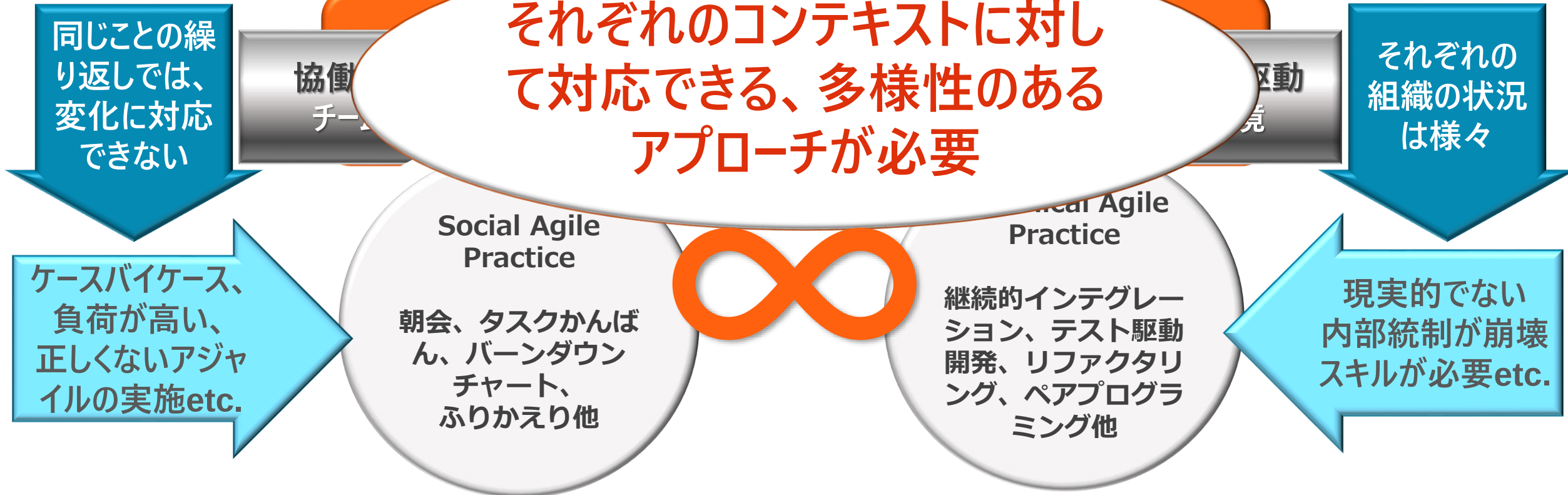
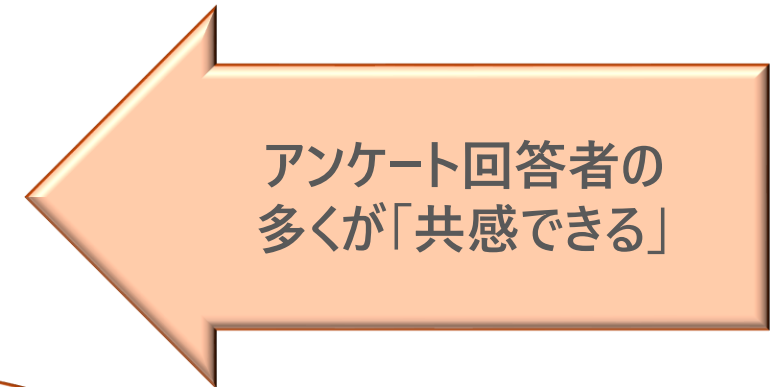


[Source]平鍋健児. 野中郁次郎 アジャイル開発とスクラム- 顧客・技術・経営をつなく協調的ソフトウェアマネジメント (Kindle版). 翔泳社, 2013, 位置No.1077 図4-14 を参考に作成



# アンケートから見た課題

プロセスやツールよりも個人と対話を、  
包括的なドキュメントよりも動くソフトウェアを、  
契約交渉よりも顧客との協調を、  
計画に従うことよりも変化への対応を



[Source]平鍋健児. 野中郁次郎 アジャイル開発とスクラム- 顧客・技術・経営をつなく協調的ソフトウェアマネジメント (Kindle版). 翔泳社, 2013, 位置No.1077 図4-14 を参考に作成

多様性のあるアプローチの実現を目指す



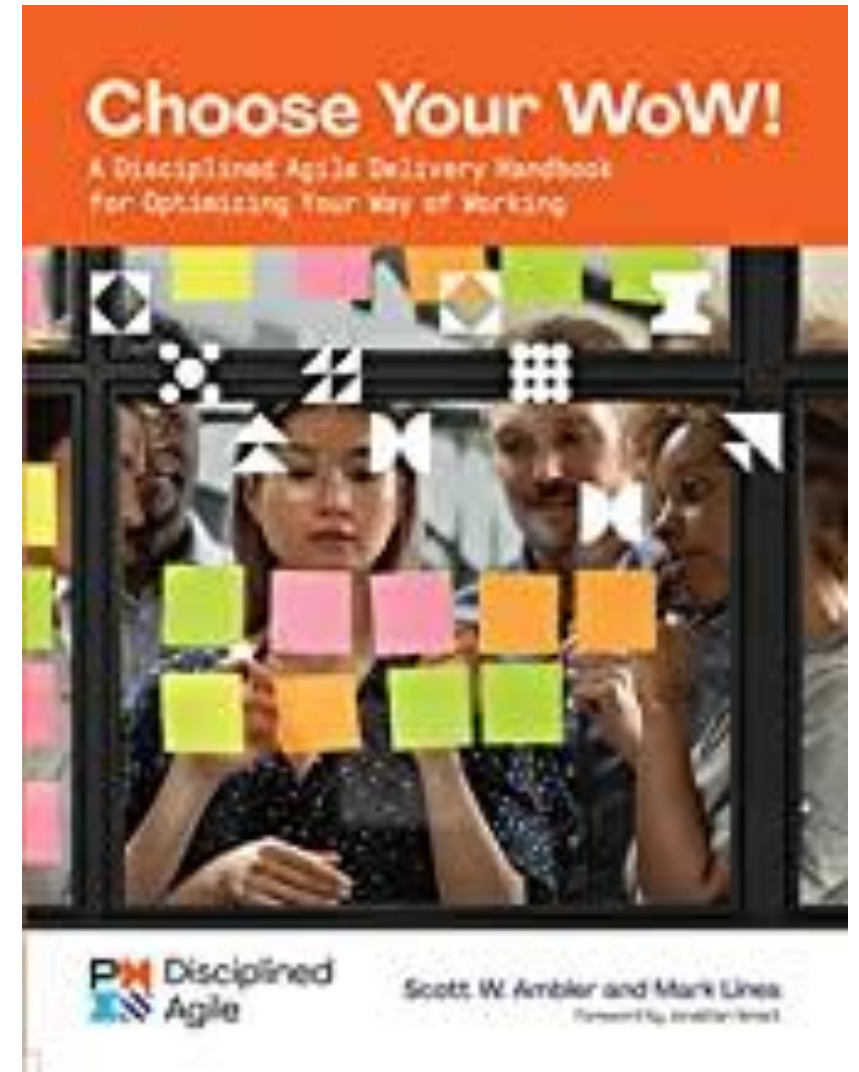


ディシプリンド・アジャイル・デリバリー (DAD)

# PMI会員はタダで 読める Choose Your WoW!

A Disciplined Agile Delivery Handbook  
for Optimizing Your Way of Working

**WoW; Way of Work** （仕事の仕方）



# なぜ、この本を読む必要があるのか

ソフトウェア  
開発の複雑さ

ルーチン化した  
アジャイル

優れた開発  
技術者の不足

私たちが妨げているいくつかの事柄

## 1. Product development is complex.

- ソフトウェアや製品開発の要件、アーキテクチャ、テスト、デザイン、プログラミング、マネジメント、デプロイ、ガバナンス、およびその他の無数のアプローチ方法

## 2. Agile industrial complex (AIC：アジャイル産業複合体).

- チームに対して組織全体と同じルーチン化を指図するフレームワーク

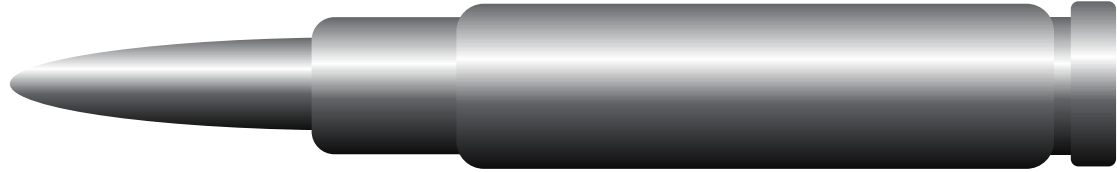
## 3. Agile growth greatly exceeded the supply of experienced coaches.

- 何日かのトレーニングではなくて、優れた人としてのスキルと、指導しているその道の何年かの経験を持つ人の不足



# なぜ、この本を読む必要があるのか

銀の弾丸は  
ない



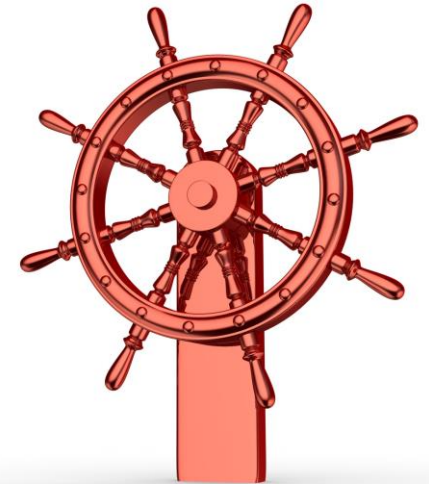
## 座礁する原因

- False promises.
  - アジャイルの採用により10倍の生産性向上、スクラムを使用すると半分の時間で2倍の作業ができると主張しているタイトルの本
- More silver bullets.
  - 2日間のトレーニングで「認定マスター」になり、4日間のトレーニング後に「プログラムコンサルタントになる」
- Process populism.
  - 業界アナリスト会社の評判や競合他社の状況に依存したリーダーシップの判断

(参考) 禿茶瓶問題； <https://youtube.com/watch?v=OA3oZpGhneQ>

# なぜ、この本を読む必要があるのか

組織に合った  
選択を行う

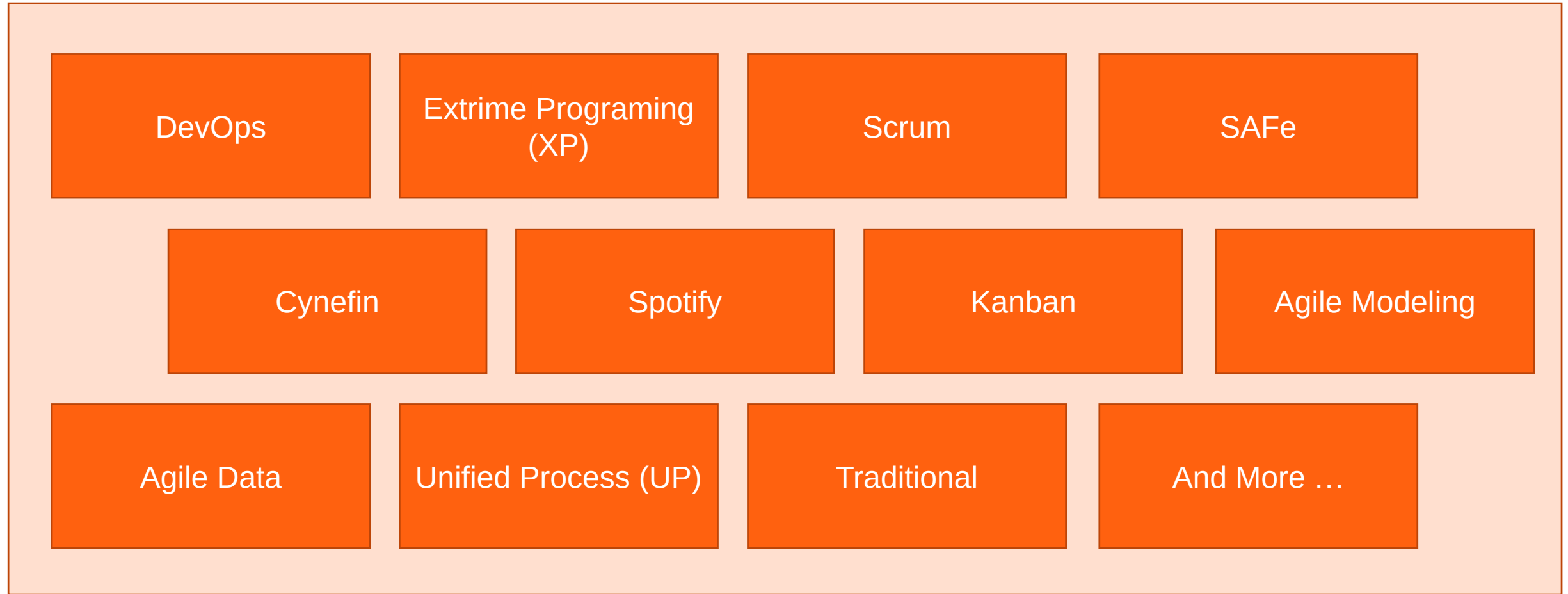


## 帆の中の風

- It embraces your uniqueness.
  - すべてのチームはユニークであり、直面している状況はユニークであることを前提とする
- It embraces the complexity you face.
  - 複雑なプロセス改善の取り組みを助けるのに役立つが、「銀の弾丸」ではない
- It provides explicit choices.
  - 組織の全体的な方向性を反映した仕事の仕方（WoW）を選択できる
- It provides agnostic（不可知論） advice.
  - 単一のフレームワークまたはメソッドに限定せず、種々のテクニックの強みと弱みを理解

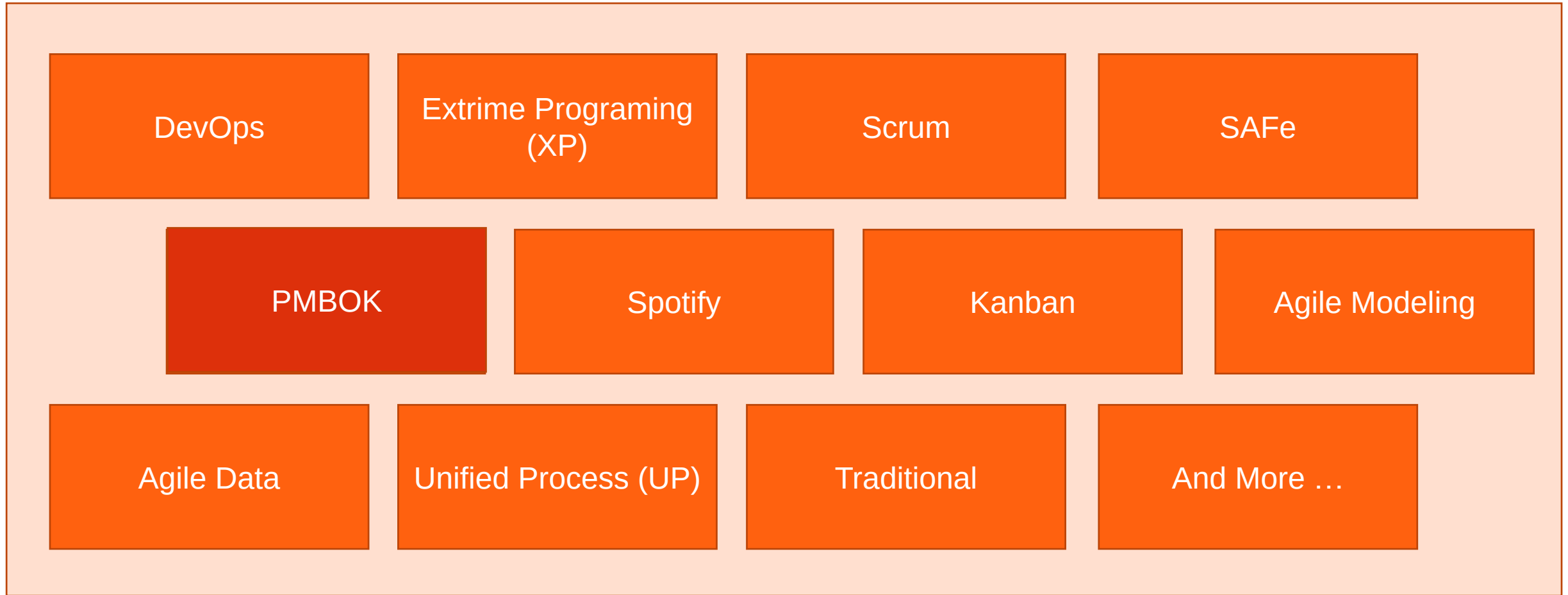
# DADは、なんでもあり

## Disciplined Agile Consortium版



# DADは、なんでもあり（PMBOKも。。。）

PMI版では...



# コンテキストに合わせて選ぶ

## コンテキスト化されたプロセスのリファレンス

- ✓ ライフサイクル
  - ✓ Agile, Continuous Delivery: Agile, Exploratory
  - ✓ Lean, Continuous Delivery: Lean, Program
- ✓ プロセスの目標
- ✓ 実践/戦略

# ソフトウェア開発のライフサイクル



# DADのライフサイクル



Agile



Continuous Delivery : Agile



Exploratory



Lean



Continuous Delivery : Lean



Program

# ソフトウェア・ライフサイクル・プロセスの国際規格：ISO/IEC/IEEE 12207:2017

## 30のプロセス

### 6.1 合意プロセス

取得プロセス(6.1.1)

供給プロセス(6.1.2)

### 6.2 組織プロジェクト イネーブリング プロセス

ライフサイクルモデル  
管理プロセス (6.2.1)

インフラストラクチャ  
管理プロセス (6.2.2)

ポートフォリオ  
管理プロセス (6.2.3)

人的資源管理プロセス  
(6.2.4)

品質管理プロセス (6.2.5)

知識管理プロセス (6.2.6)

プロジェクト計画プロセス  
(6.3.1)

プロジェクトアセスメント  
及び制御プロセス (6.3.2)

意思決定プロセス (6.3.3)

### 6.3 テクニカル管理プロセス

リスク管理プロセス  
(6.3.4)

構成管理プロセス (6.3.5)

情報管理プロセス (6.3.6)

測定プロセス (6.3.7)

品質保証プロセス(6.3.8)

### 6.4 テクニカルプロセス

ビジネスまたはミッション  
分析プロセス(6.4.1)

利害関係者ニーズ及び  
要求定義プロセス (6.4.2)

システム/ソフトウェア  
要求定義プロセス (6.4.3)

アーキテクチャ定義プロセ  
ス (6.4.4)

設計定義プロセス (6.4.5)

システム分析プロセス  
(6.4.6)

実装プロセス (6.4.7)

結合プロセス (6.4.8)

検証プロセス (6.4.9)

移行プロセス(6.4.10)

妥当性確認プロセス  
(6.4.11)

運用プロセス(6.4.12)

保守プロセス (6.4.13)

廃棄プロセス(6.4.14)



# ソフトウェア・ライフサイクル・プロセスの国際規格：合意プロセス

6.1合意プロセス

取得プロセス(6.1.1)

供給プロセス(6.1.2)

組織は、システムの生産者であり利用者でもある。製品又はサービスに対して、（取得者として行動する）一つの組織は、（供給者として行動する）別の組織に仕事を課すことができる。これは、合意を用いて達成される。

6.2組織プロジェクト  
イネーブリング  
プロセス

ライフサイクルモデル  
管理プロセス (6.2.1)

インフラストラクチャ  
管理プロセス (6.2.2)

ポートフォリオ  
管理プロセス (6.2.3)

人的資源管理プロセス  
(6.2.4)

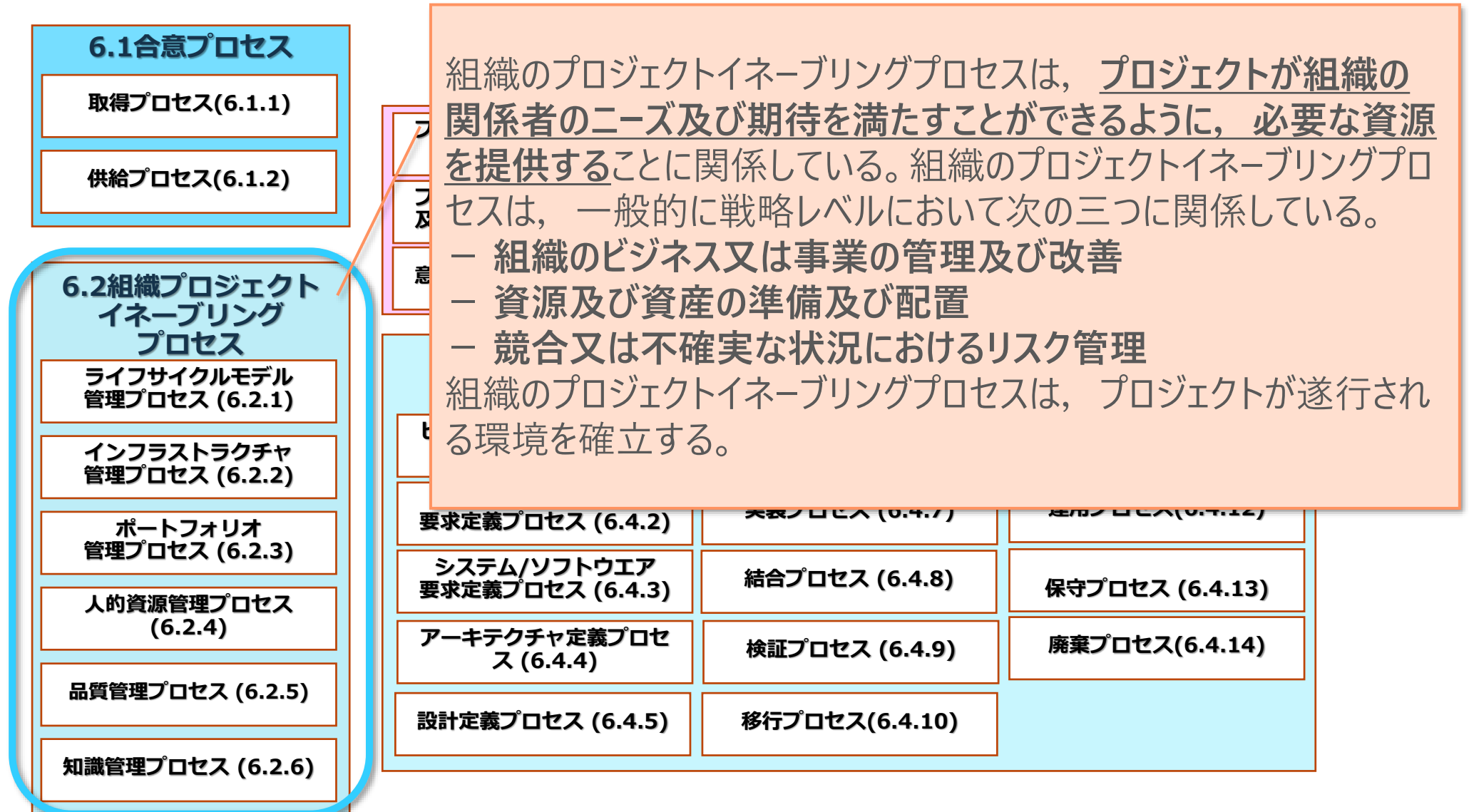
品質管理プロセス (6.2.5)

知識管理プロセス (6.2.6)

6.4  
テクニカルプロセス

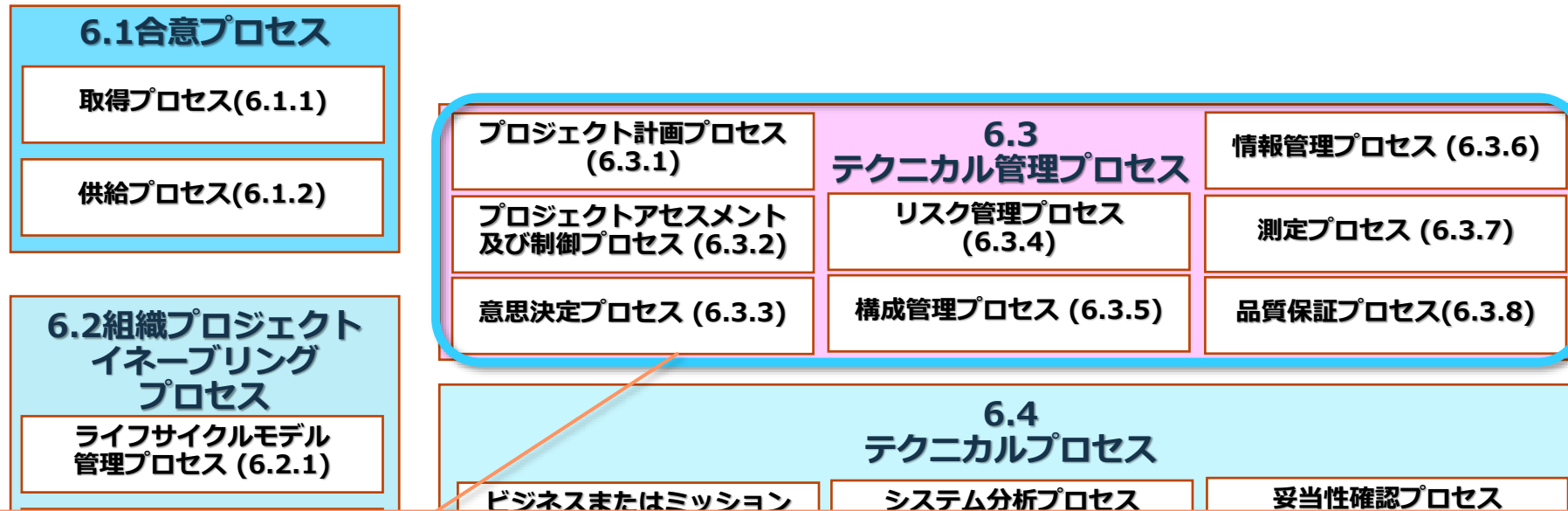
|                                 |                       |                       |
|---------------------------------|-----------------------|-----------------------|
| ビジネスまたはミッション<br>分析プロセス(6.4.1)   | システム分析プロセス<br>(6.4.6) | 妥当性確認プロセス<br>(6.4.11) |
| 利害関係者ニーズ及び<br>要求定義プロセス (6.4.2)  | 実装プロセス (6.4.7)        | 運用プロセス(6.4.12)        |
| システム/ソフトウェア<br>要求定義プロセス (6.4.3) | 結合プロセス (6.4.8)        | 保守プロセス (6.4.13)       |
| アーキテクチャ定義プロセ<br>ス (6.4.4)       | 検証プロセス (6.4.9)        | 廃棄プロセス(6.4.14)        |
| 設計定義プロセス (6.4.5)                | 移行プロセス(6.4.10)        |                       |

# ソフトウェア・ライフサイクル・プロセスの国際規格：組織プロジェクトイネーブリングプロセス



[Source] ISO/IEC/IEEE 12207:2017, Systems and software engineering — Software life cycle processes, 2017をJISX 0160:2012を参考に仮訳

# ソフトウェア・ライフサイクル・プロセスの国際規格：テクニカル管理プロセス



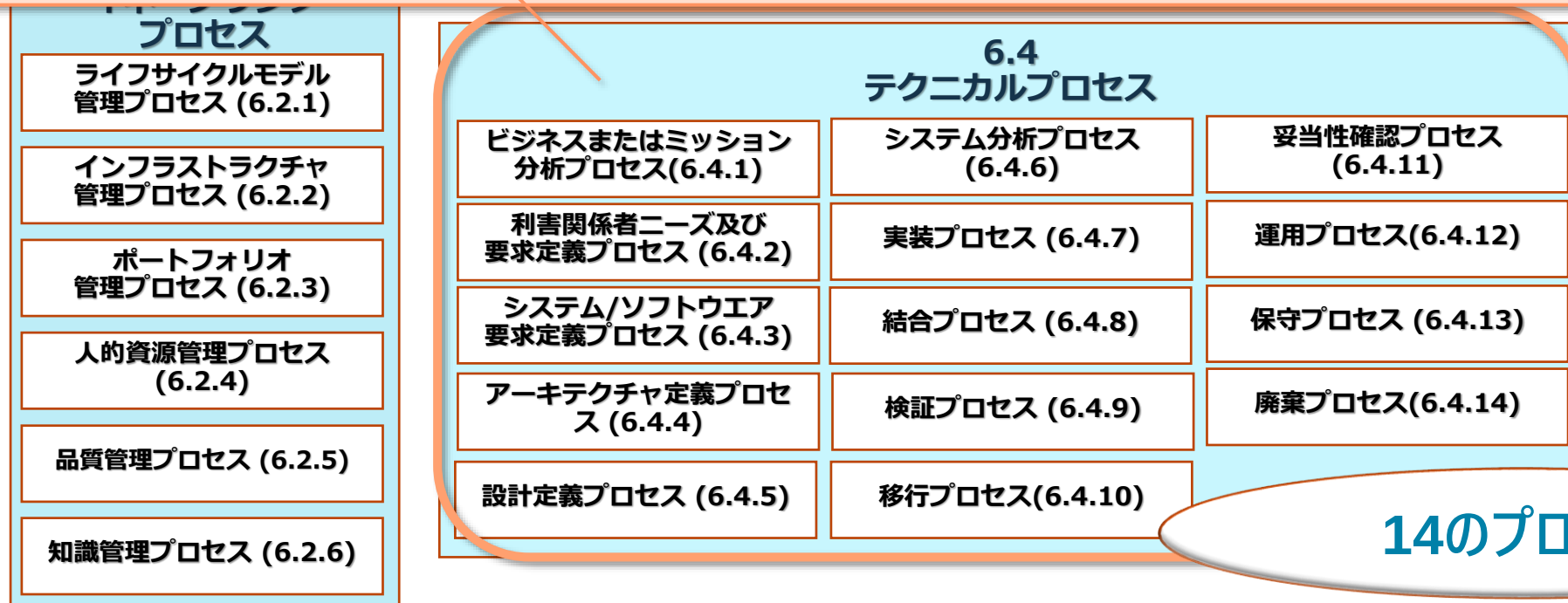
テクニカルマネジメントプロセスは、組織の管理者によって割り当てられた資源及び資産を管理すること，並びに一つ以上の組織が行った合意を果たすために資源及び資産を適用することに関係している。

テクニカルマネジメントプロセスは、プロジェクトの技術面の作業に関係するが，特に次に関係する。

- － コスト，期間の長さ及び達成に関して計画すること
- － 計画及び達成度を示す実績基準を遵守することを確実なものにするよう助けるために，活動をチェックすること
- － 進捗及び達成における未達成の部分を取り戻す是正処置の識別及び選択すること

# ソフトウェア・ライフサイクル・プロセスの国際規格：テクニカルプロセス

テクニカルプロセスは、ライフサイクルを通して技術的活動に関係する。テクニカルプロセスは、利害関係者のニーズを製品及びサービスに変換する。テクニカルプロセスは、その製品を適用するか、又はそのサービスを運用することによって、必要なときに必要な所で、現時点から将来にわたって持続的に、利害関係者要求事項を満たし顧客満足を獲得できるようにする。テクニカルプロセスは、モデルとして表現されたシステムであるか、完成品のシステムであるかにかかわらず、システムを開発して使用するために適用する。テクニカルプロセスは、システム構造の階層のどのレベルにおいても、及びライフサイクルのどの段階においても適用する。



[Source] ISO/IEC/IEEE12207:2017, Systems and software engineering — Software life cycle processes, 2017をJISX0160:2012を参考に仮訳

©2020 PMI Japan Chapter. All Rights Reserved.

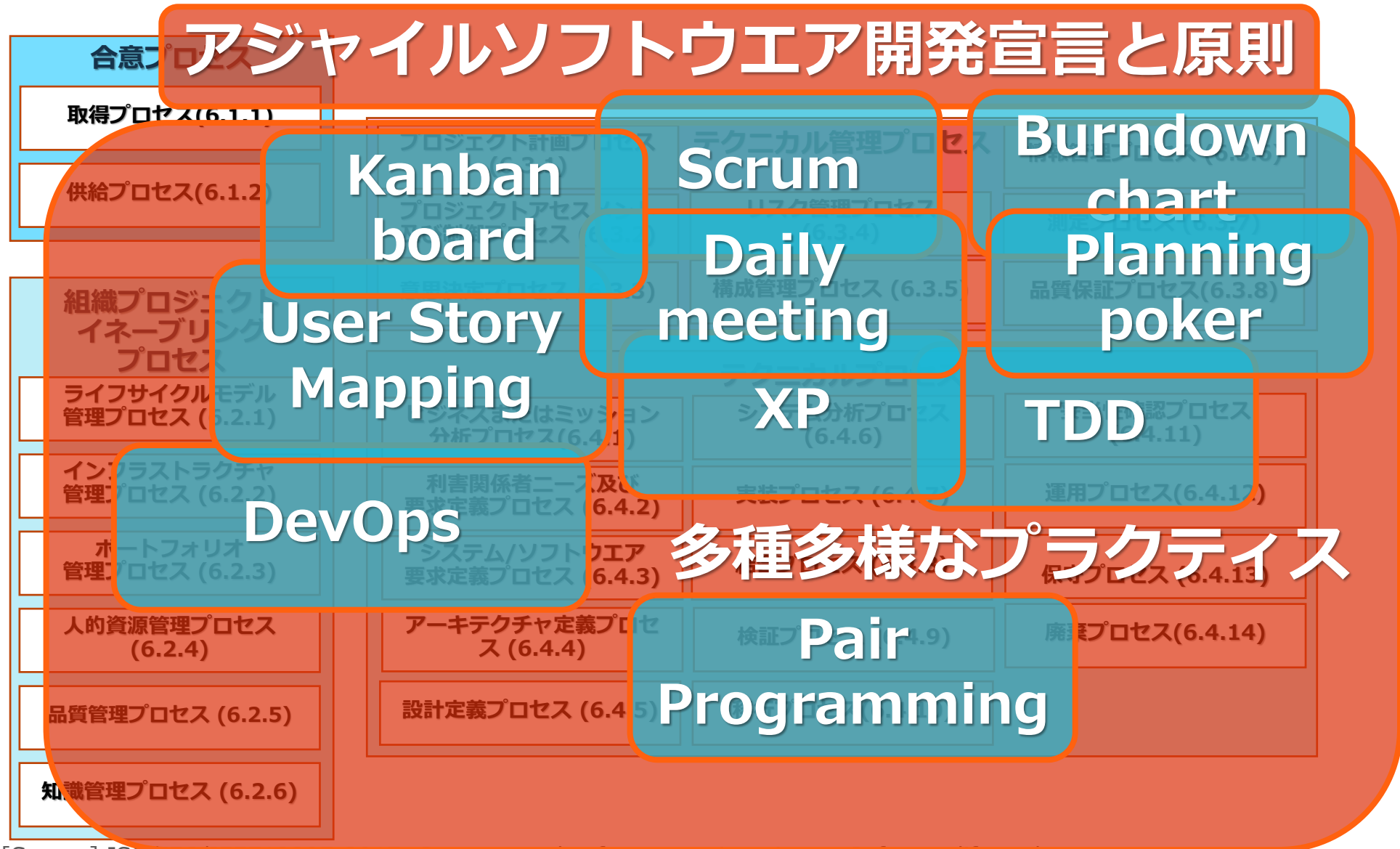
# ソフトウェア・ライフサイクル・プロセスの国際規格：アジャイルとは？

## 5.4 Life cycle concepts - 5.4.2 Life cycle model for the software system (仮訳)

- 「アジャイル」メソッドは、実際にはさまざまなモデルに適用できる。それぞれのアジャイルメソッド（**Agile methods**）は、進化的なライフサイクルモデルを実行するという点で共通だが、**さまざまな段階（stages）で別のライフサイクルモデルを使用する**。これらのメソッドに共通しているのは、要件への変更を含む変更が予想される環境において、動くソフトウェアを高速開発するために、継続的な検査とコラボレーションを重視する点である。

「アジャイル開発」という開発モデルはなく、  
さまざまなライフサイクルモデル（テーラリング）

# ソフトウェア・ライフサイクル・プロセスの国際規格：アジャイルプラクティスを俯瞰してみると

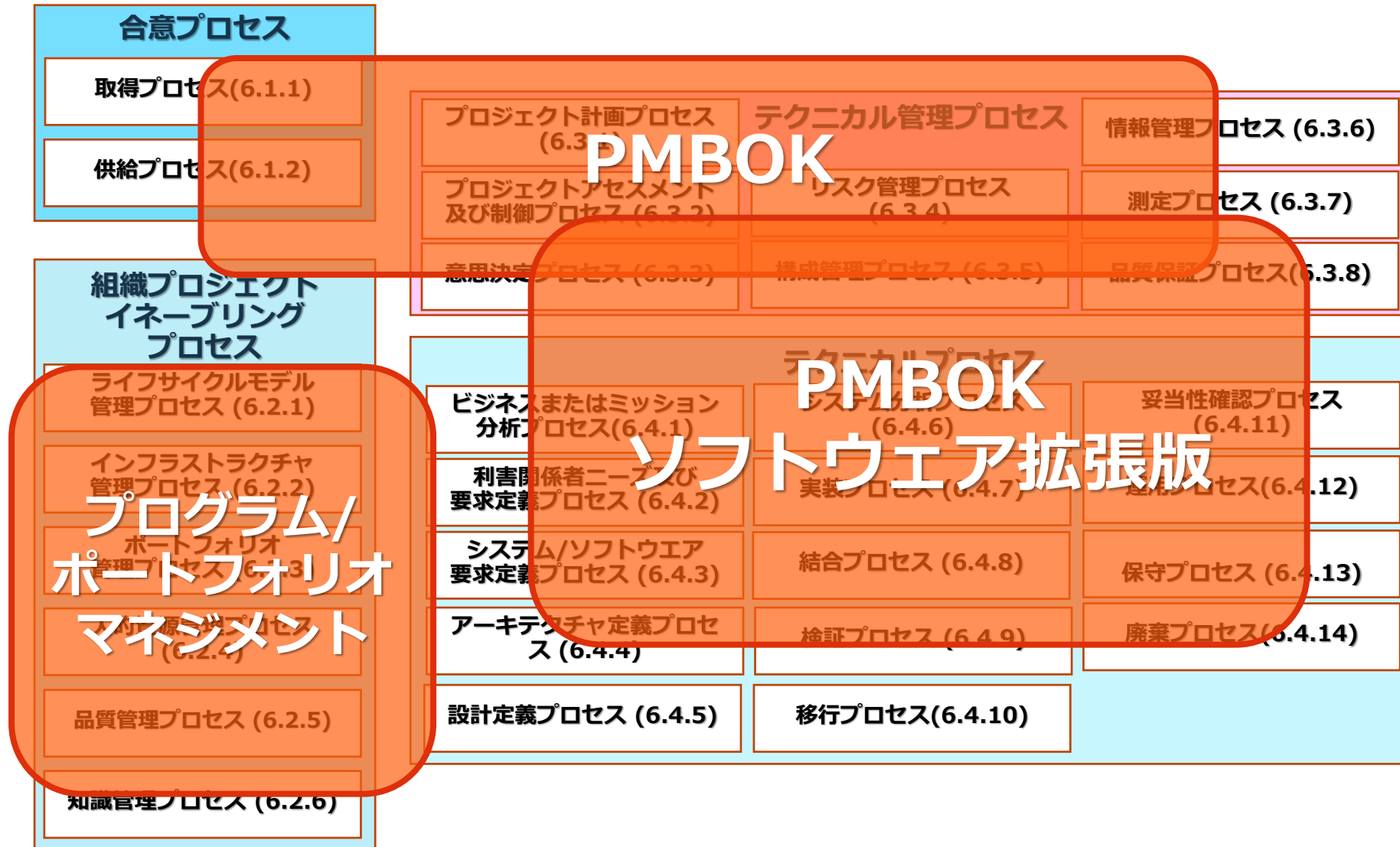


[Source] ISO/IEC/IEEE 12207:2017, Systems and software engineering — Software life cycle processes, 2017をJISX0160:2012を参考に仮訳

©2020 PMI Japan Chapter. All Rights Reserved.



# ソフトウェア・ライフサイクル・プロセスの国際規格：プロジェクトマネジメントを見てみると



[Source] ISO/IEC/IEEE12207:2017, Systems and software engineering — Software life cycle processes, 2017をJISX0160:2012を参考に仮訳

©2020 PMI Japan Chapter. All Rights Reserved.

# プロセスの目標と実践/戦略





## アジャイルソフトウェア開発宣言

私たちは、ソフトウェア開発の実践あるいは実践を手助けをする活動を通じて、よりよい開発方法を見つけだそうとしている。この活動を通して、私たちは以下の価値に至った。

プロセスやツールよりも**個人と対話**を、  
包括的なドキュメントよりも**動くソフトウェア**を、  
契約交渉よりも**顧客との協調**を、  
計画に従うことよりも**変化への対応**を、

価値とする。すなわち、左記のことがらに価値があることを認めながらも、私たちは右記のことがらにより価値をおく。

© 2001, 上記の著者たち

この宣言は、この注意書きも含めた形で**全文を含めることを条件に自由にコピーしてよい。**

[Source] Kent Beck他17名の著者たち, 2001, <http://agilemanifesto.org/iso/ja/manifesto.html>  
(2017/09/18 access)

# 大野耐一の名著『トヨタ生産方式-脱規模の経営を目指して』

## アジャイルソフトウェア開発宣言は、トヨタ生産方式の考え方の一部

現場主義；プロセスやツールよりも**個人と対話**を、

- ・ 手をとって教え込まなければならず、これが監督者に対する信頼感につながる。作業者と作業者のつなぎの工程は「助け合い」ができるようにつくってある。「人間の和」が育つ。

現物主義；包括的なドキュメントよりも**動くソフトウェア**を、

- ・ どこにも手本はない。何が起るかやってみないとわからないことが多く、毎日毎日が緊張の連続であった。やってみては直し、またやってみる。また直す。

ジャストインタイム；契約交渉よりも**顧客との協調**を、

- ・ トヨタ式は「最終の市場では、お客さんが一人、一人、違った車を一台ずつ買うのであるから、生産の場においても一台、一台つくる。部品をつくる段階においても、一個、一個つくっていく。

朝令暮改；計画に従うことよりも**変化への対応**を、

- ・ 状況が変われば、やり方も変えていくのは当然であるし、また変化に対応できるよう現場の体質をつくり上げていくこと、自分自身の頭脳を柔軟に保つことこそ大切。

# プロセスの目標と実践/戦略

## 「あなたの組織の生産方式」（例えば4つの価値観 + $\alpha$ ）

現場主義；（プロセスやツールよりも個人と対話を）

- ・ チームの信頼感、助け合い、人間の和

現物主義；（包括的なドキュメントよりも動くソフトウェア）

- ・ 何が起こるかやってみないとわからないことへの取り組みの推奨（やってみては直し、またやる。また直す。）

ジャスト・イン・タイム；（契約交渉よりも顧客との協調）

- ・ お客さま一人一人をどう大切にしていくのか

朝令暮改；（計画に従うことよりも変化への対応）

- ・ 状況に応じた臨機の計画見直し。変化に対応できる現場の体質。頭脳を柔軟に保つ。

自動化；

### 「トヨタ生産方式」からの更なるヒント

- ・ 機械に人間の知恵を付ける。「人間の動きを、いかにコンベアの付いた働きにするか」。「目で見える管理」を行き届かせ、生産現場の弱点を浮き彫りにし、強化策を講じる。
- ・ 古来、日本には「臭いものにはフタをする」という諺がある。問題が管理監督者の知らないところで繕われていては、改善が進まない。



# ライト、ついてますか

長いトンネルの前、走行中に停電しても大丈夫なように注意標識

**注意 前方にトンネルがあります ライトをつけてください**

トンネルの先の眺めのよい高台 …… みんな駐車場に車を止めて眺めを堪能した  
車に戻ると…「ライトつけっぱなしでバッテリー上がっちゃった…」

トンネルの出口に、「ライトを消せ」の注意標識が欲しい。…でも夜に走っていたら、消すと危ない

もし今が昼間でライトがついているならライトを消せ  
もし今暗くてライトが消えているならライトをつけよ  
もし今が昼間でライトが消えているならライトを消したままとせよ  
もし今暗くてライトがついているならライトをついたままとせよ

[Source] ドナルド・C・ゴース, G.M.ワインバーグ, ライト、ついてますか—問題発見の人間学. 共立出版, 1987, pp.99-104



# ライト、ついてますか

もし人々の頭の中のライトが  
ついているなら、  
ちょっと思い出させてやる方が、  
ごちゃごちゃいうより有効なのだ

ソフトウェア開発においても、  
アジャイルかウォーターフォールかではなく、  
「コンテキストに応じたアプローチ」が大事



ライトついてますか？

# ライト、ついてますね？

今回のアンケートでは、

「アジャイルソフトウェア開発宣言」は、広く共感を得られることが確認できました。

また、それぞれの組織では既に、「プロジェクトマネジメント・アプローチ」の種類にかかわらず、  
いろいろなプラクティスに取り組んでいました。

その実践の結果として、「アジャイルプラクティスの適用はケース・バイ・ケース」という、  
重要な指摘をいただきました。

ソフトウェア開発には「銀の弾丸」はありません。自分たちの組織やその状況に合った

**「コンテキストに応じた多様性のあるアプローチ」**

に引き続きチャレンジしていきましょう。

# PMI日本支部 アジャイル研究会からの お知らせ



# アジャイル研究会の仲間になりませんか？

グローバルでプロジェクトマネジメントをリードするPMIおよびPMI日本支部の立場で、アジャイルの普及、啓発を行っています。そのため、アジャイルの最も基本となるアジャイルマニフェスト（アジャイルの価値と原則）の考えに基づき、PMBOK®ガイドとアジャイルの関連性を重視し、議論や情報収集を行い研鑽しています。また、外部関連団体とも連携を図り、PMI内外に対して積極的に情報発信も行っていきます。

- 定例会での意見交換・議論(毎月第三木曜日19:00~21:00にZoom開催)
  - アジャイルプロジェクトマネジメントに関する議論と情報交換
  - アジャイルに関する勉強会などの実施
  - アジャイルに関する内外団体との連携
  - アジャイルに関する意識調査（アンケート）の実施・分析・提言
- 応募資格 PMI日本支部会員で、年間を通じて積極的に活動いただける方。

詳しくは  
こちらへ



# 自己紹介

成田 和弘 : Kazuhiro Narita



銀行員；勘定系開発、IT投資戦略、基盤開発、システム統合から  
監査まで（所属：三菱UFJトラストシステム(株) 品質管理部）

アジャイル研究会、プロジェクトマネジメント研究会、ビジネス  
アナリシス研究会

他団体；システム監査学会（理事）、日本クラウドセキュリティ  
アライアンス(運営委員)、ISACA東京支部 など

（R）研究論文）トヨタ生産方式のシステム開発および監査への実装（システム監査 第32巻第  
1号）（<https://www.sysaudit.gr.jp/gakkaishi/ronbun/201906/narita.pdf>）他

趣味 ；絵画（2018年日本水彩展入選、2019年 千葉県展入選）、  
「からすみ」「梅干し」作り

資格 ；CIA、CISA、情報処理技術者試験（システム監査技術者、システムアナリスト、プロ  
ジェクトマネージャー、情報セキュリティアドミニストレータ、ITストラテジスト 他）



魚ヲ売ル (2018年日本水彩展)



# END



この資料の内容は発表者個人の見解です。  
発表者の所属会社・組織等とは関係ありません。