

提供



Google Cloud

Accelerate State of DevOps Report 2023



Premiere スポンサー



Deloitte.

Qarik

v. 2023-12

目次

序章

概要	03
コンセプトと尺度	06

1章

他社との比較	10
--------------	----

2章

ユーザーを重視する	17
-----------------	----

3章

技術的能力からパフォーマンスを 予測する	20
-------------------------------	----

4章

ドキュメントが基礎となる	27
--------------------	----

5章

信頼性がパフォーマンスを 高める	31
---------------------------	----

6章

柔軟なインフラストラクチャが 成功の鍵	38
------------------------------	----

7章

組織文化への投資なしには 何事もうまくいかない	45
----------------------------------	----

8章

個人の属性が重要になる 状況と理由	51
----------------------------	----

あとがき

最後に	57
謝辞	58
著者	59
方法論	62
ユーザー属性と企業特性	72
モデル	81
関連情報	91
付録	92

引用文献はすべて2023年9月27日に参照

概要

DORA の研究プログラムは、パフォーマンスの高いテクノロジー主導型組織の能力と尺度について 10 年近くにわたって調査を行ってきました。その中で、さまざまな業界のあらゆる規模の組織に所属する、36,000 人超のプロフェッショナルの方々にお話を伺いました。皆様のご協力に感謝いたします。

DORA は、業務の進め方（つまり能力）と成果（組織全体とそこに所属する人員に関係する、有意義な業績）の関係を探っています。本調査は厳密な統計的評価を利用しており、プラットフォームに依存しません（[方法論](#)をご覧ください）。

こうして得られた知見から、リーダーや実務者の皆様に、インパクトを生み出すためのヒントをつかんでいただければ幸いです。

今年の調査では次に示す 3 つの主な成果と、その達成に寄与する能力について調べました。

- **組織パフォーマンス** - 組織は収益を上げるだけでなく、顧客のために、また、その周囲に広がるコミュニティのために価値を生み出す必要があります。
- **チームパフォーマンス** - アプリチームやサービスチームが価値を生み、イノベーションを起こし、協力する能力が必要です。
- **従業員のウェルビーイング** - 組織やチームで採用する戦略は、従業員にとって有益なものである必要があります。つまり、燃え尽き症候群を減らし、仕事への満足度を高め、価値ある成果を生み出す能力（生産性）を高めることが求められます。

本調査では、それ自体が目的のように語られることの多い、手段やパフォーマンスの尺度についても調べました。

- **ソフトウェアデリバリーパフォーマンス** - チームがテクノロジーシステムを安全、迅速、効率的に変更できること。
- **運用パフォーマンス** - サービスが信頼性の高いユーザーエクスペリエンスを提供すること。



主な調査結果



健全な組織文化を確立する

組織文化は、技術力を築き上げ、技術的パフォーマンスを高め、組織のパフォーマンス目標を達成し、従業員の成功を支援するために不可欠です。創造的な文化のあるチームは、そうでないチームと比べて組織パフォーマンスが 30% 高くなっています。



ユーザーを念頭に置いて構築する

ユーザーを重視することで、この調査の対象である技術、プロセス、組織文化のすべての能力にわたり、改善のヒントを得て実現していくことができます。望みどおり迅速かつ効果的にデプロイできたとしても、ユーザーのことを考えていなければ意味がありません。ユーザーを重視するチームは、そうでないチームと比べて組織パフォーマンスが 40% 高くなっています。



コードレビューを迅速化してソフトウェアデリバリーパフォーマンスを向上させる

コードレビューのスピードアップは、ソフトウェアデリバリーパフォーマンスを向上させる方法として非常に効果的です。コードレビューを迅速化したチームは、そうでないチームと比べてソフトウェアデリバリーパフォーマンスが 50% 高くなっています。



質の高いドキュメントで技術的能力を増幅する

質の高いドキュメントは、技術的能力が組織パフォーマンスにもたらす影響を増幅します。たとえばトラックベース開発では、質の高いドキュメントが用意されている場合、質の低いドキュメントしかない場合に比べて、組織パフォーマンスに与える影響が 12.8 倍になると推定されています。



クラウドでインフラストラクチャの柔軟性を高める

クラウドコンピューティングが有益なのは、インフラストラクチャが柔軟になるからです。たとえば、パブリッククラウドを使用した場合、クラウドを使用しなかった場合と比べてインフラストラクチャの柔軟性が 22% 高くなります。この柔軟性により、インフラストラクチャの柔軟性が低い場合と比べて、組織パフォーマンスを 30% 高めることができます。クラウドを最大限有効に活用するために重要なのは、クラウドの差別化につながる特性や機能、つまりインフラストラクチャの柔軟性を利用することです。



デリバリースピード、運用パフォーマンス、ユーザー重視のバランスをとる

組織パフォーマンスを最大限に引き出すには、優れたソフトウェアデリバリーパフォーマンスと優れた運用パフォーマンスの両方が必要です。ユーザーを重視しながらこの 2 つのバランスを維持することで、従業員のウェルビーイングを向上させつつ、組織として最高の結果を出すことができます。



仕事を公平に配分する

過小評価されている（十分に代表されていない、公平に声が反映されていない）グループに属することを自認している人や、女性、ジェンダーを自己決定している人は、燃え尽き症候群に陥る可能性が高いことがわかっています。その原因として、構造的、環境的な複数の要因が考えられます。反復作業を多く引き受けている回答者は燃え尽き症候群の程度が大きくなる傾向があり、過小評価グループのメンバーはより多くの反復作業を引き受ける傾向があるという結果は予想どおりでした。過小評価グループの回答者は、そうではない回答者と比べて燃え尽き症候群の報告が 24% 多く、反復作業の量が 29% 多くなっています。女性や、ジェンダーを自己決定している人々は、男性と比べて 40% 多くの反復作業を行っています。

DORA の分析情報をお客様の コンテキストに適用

「継続的な改善」の考え方と手法を採用しているチームは、成功する可能性が高くなります¹。DORA は、お客様独自の改善の取り組みに貢献することができます。

本調査を最大限に活用するには、ご自身のチームとユーザーの観点から考えてみてください。たとえば先述のように、コードレビューを迅速化したチームは、そうでないチームと比べてソフトウェア デリバリー パフォーマンスが 50% 高くなっています。しかし、コードレビューが速くてもシステムの他の部分にスピードの制約がある場合、ソフトウェア デリバリー パフォーマンスが向上する可能性は低くなります。本調査を現状に当てはめて考えるには、実務者が今の業務の進め方について話し合う必要があります。こうした会話は、共感や協力を促進し、各参加者の動機について理解を深めることにつながります。

改善の取り組みに終わりはありません。システムのボトルネックを見つけて対処し、プロセスを繰り返します。他のアプリケーション、組織、業界を見るのではなく、同じアプリケーションを長期的に見て比較することが最も重要です。

指標と測定

指標とダッシュボードは、チームが進捗状況をモニタリングし、軌道修正するために役立ちます。

実務者とリーダーは、組織パフォーマンス、チームパフォーマンス、ウェルビーイングのために努力しています。しかし、ソフトウェアを提供することが目標ではないのと同様に、測定は目標ではありません。

パフォーマンス指標に固執すると、効果的でない行動につながるおそれがあります。成功を果たすためには、能力と学習に投資する方がよいでしょう。よく学ぶチームほど大きな改善ができるものです。

1 人では改善できません

互いの経験から学びましょう。DORA コミュニティ サイト (<https://dora.community>) は、改善の取り組みについて共有し学ぶことができるフォーラムです。

¹2022 Accelerate State of DevOps Report.
<https://dora.dev/research/2022/dora-report/2022-dora-accelerate-state-of-devops-report.pdf#page=7>



コンセプトと尺度

このセクションでは、DORA が測定しようとしているコンセプトについて説明します¹。これは本レポートとモデルの基礎をなしている要素です。以降のセクションには、本レポートを作成するための材料となった要素が含まれています。そのため、私たち著者がこうしたコンセプトが指すものを明確にし、その説明の方法に一貫性を持たせることが重要です。掲載している表は、読者や著者も含め、全員で共有する用語を明確にし、その根拠を示すためのものです。

本レポートのコンセプトの多くは多面的であるため、うまく捉えるために複数の指標を使用することがよくあります。こうしたコンセプトをうまく捉えることができているかどうかを評価する方法として、探索的因子分析と確認的因子分析があります。プロセスの詳細については、[方法論](#)をご覧ください。測定方法を評価した後、0 から 10 の範囲でスコアをつけました（コンセプトが完全に欠如しているものが 0、コンセプトが最大限に存在しているものが 10）。このようにすることで、コンセプトがどのように機能するのかについて語る時の表現方法が標準化されると私たちは考えており、これは年をまたいでデータを比較する際に役立ちます。

各コンセプトには以下のような情報が添えられています。

- 意味を伝え、本章を参照する際に見つけやすくするためのアイコン。
- サンプルにおける当該コンセプトの平均スコア（平均）。
- 四分位範囲（IQR）の境界。データの中位 50% が存在する 2 つの数値（25 パーセンタイル、75 パーセンタイル）を示すことで、回答の広がり具合が伝わりやすくなるはずです。
- データセットの中位の値（中央値）。平均値と大幅に異なる場合、データに偏りがあることを示している可能性があります。
- コンセプトとその測定方法の説明。

¹分析に使用したアンケートの質問は <https://dora.dev> で公開されています

主な成果

主な成果とは、人員、チーム、組織が目指しているであろう到達点（組織パフォーマンスなど）、または避けようとしているであろう到達点（燃え尽き症候群など）のことで、そのためこの尺度は、人員が自身やチーム、組織を評価するための重要な方法になると考えられます。

組織パフォーマンス			チームパフォーマンス			ソフトウェアデリバリーパフォーマンス		
平均	IQR	中央値	平均	IQR	中央値	平均	IQR	中央値
6.3	5~8	6.3	7.6	6.6~9	8	6.3	5.1~7.8	6.4
パフォーマンスの高い組織は、主要なプロダクトやサービスで顧客が多く、利益が高く、相対的なマーケットシェアが大きくなります。			パフォーマンスの高いチームでは、変化への適応、相互の信頼、働き方の効率化、イノベーションの実現、コラボレーションが見られます。			次の4つの指標でソフトウェアデリバリーのスピードと安定性を測定します。 <ul style="list-style-type: none">• デプロイの頻度• 変更のリードタイム• 変更時の障害率• デプロイ失敗時の復旧までの時間		
運用パフォーマンス			信頼性目標					
平均	IQR	中央値	平均	IQR	中央値			
6.2	5~7.5	6.3	7	5~7.5	7.5			
サービスがユーザーの期待にどの程度応えられるか（可用性やパフォーマンスなどの尺度を含む）。			サービスが、可用性、パフォーマンス、正確性などの尺度について、示した目標をどの程度達成しているか。					

ウェルビーイングは、燃え尽き症候群、生産性、仕事の満足度の組み合わせを指します。

燃え尽き症候群			生産性			仕事の満足度		
平均	IQR	中央値	平均	IQR	中央値	平均	IQR	中央値
4.1	2~6	4	7.5	6.7~8.8	7.9	6.08	5.7~7.1	7.1
仕事による心理的、肉体的負担のほか、自身の仕事の価値や意味をどのように評価するか。燃え尽き症候群は悲観的な考えや不信感につながります ² 。			生産性の高い人員は、自身のスキルに合った仕事をし、価値を生み、効率的に働きます。			あらゆることを考慮し、自身の仕事を全体的にどのように感じているかを評価するよう回答者に求める、単一項目の質問 ³ 。		

² Maslach C, Leiter MP. Understanding the burnout experience: recent research and its implications for psychiatry (英語). World Psychiatry.2016 Jun;15 (2) : 103~11. doi: 10.1002/wps.20311.PMID: 27265691; PMCID: PMC4911781.

³ Warr, P., Cook, J., & Wall, T. "Scales for the measurement of some work attitudes and aspects of psychological well-being." Journal of Occupational Psychology, 52(2), 1979.129-148. <https://doi.org/10.1111/j.2044-8325.1979.tb00448.x>

プロセスと技術的能力

チームや組織で生じる可能性のある活動、手法、状態のいずれかです。言い換えると、チームが行うことやチームのあり方です。

 AIの寄与	 疎結合アーキテクチャ	 継続的インテグレーション																		
<table border="1"><thead><tr><th>平均</th><th>IQR</th><th>中央値</th></tr></thead><tbody><tr><td>3.3</td><td>0.3~6.3</td><td>2.4</td></tr></tbody></table>	平均	IQR	中央値	3.3	0.3~6.3	2.4	<table border="1"><thead><tr><th>平均</th><th>IQR</th><th>中央値</th></tr></thead><tbody><tr><td>6.4</td><td>4.7~8.3</td><td>6.7</td></tr></tbody></table>	平均	IQR	中央値	6.4	4.7~8.3	6.7	<table border="1"><thead><tr><th>平均</th><th>IQR</th><th>中央値</th></tr></thead><tbody><tr><td>6.9</td><td>5~8.9</td><td>7.8</td></tr></tbody></table>	平均	IQR	中央値	6.9	5~8.9	7.8
平均	IQR	中央値																		
3.3	0.3~6.3	2.4																		
平均	IQR	中央値																		
6.4	4.7~8.3	6.7																		
平均	IQR	中央値																		
6.9	5~8.9	7.8																		
さまざまな技術タスクに寄与するAIの役割の重要性。	独立して記述、テスト、デプロイできるソフトウェア。	ソフトウェアの変更を自動的にビルドし、テストする手法。																		
 継続的デリバリー	 コードレビューのスピード	 ドキュメント																		
<table border="1"><thead><tr><th>平均</th><th>IQR</th><th>中央値</th></tr></thead><tbody><tr><td>7.0</td><td>5.7~8.7</td><td>7.3</td></tr></tbody></table>	平均	IQR	中央値	7.0	5.7~8.7	7.3	<table border="1"><thead><tr><th>平均</th><th>IQR</th><th>中央値</th></tr></thead><tbody><tr><td>6.5</td><td>6~8</td><td>6</td></tr></tbody></table>	平均	IQR	中央値	6.5	6~8	6	<table border="1"><thead><tr><th>平均</th><th>IQR</th><th>中央値</th></tr></thead><tbody><tr><td>5.8</td><td>3.8~7.9</td><td>6.25</td></tr></tbody></table>	平均	IQR	中央値	5.8	3.8~7.9	6.25
平均	IQR	中央値																		
7.0	5.7~8.7	7.3																		
平均	IQR	中央値																		
6.5	6~8	6																		
平均	IQR	中央値																		
5.8	3.8~7.9	6.25																		
新機能、構成変更、バグの修正、テストなどのあらゆる変更を、安全、迅速、サステナブルに、本番環境に適用またはユーザーに届ける能力 ⁴ 。	pull リクエストからコード変更レビューまでにかかる時間を評価する単一項目。	組織の人員が作成し日常業務で使用する書面の品質。																		
 信頼性の手法	 トランクベース開発	 柔軟なインフラストラクチャ																		
<table border="1"><thead><tr><th>平均</th><th>IQR</th><th>中央値</th></tr></thead><tbody><tr><td>5.9</td><td>3.9~8.3</td><td>6.1</td></tr></tbody></table>	平均	IQR	中央値	5.9	3.9~8.3	6.1	<table border="1"><thead><tr><th>平均</th><th>IQR</th><th>中央値</th></tr></thead><tbody><tr><td>5.6</td><td>3.9~7.8</td><td>5.6</td></tr></tbody></table>	平均	IQR	中央値	5.6	3.9~7.8	5.6	<table border="1"><thead><tr><th>平均</th><th>IQR</th><th>中央値</th></tr></thead><tbody><tr><td>6.6</td><td>5~8.3</td><td>7.3</td></tr></tbody></table>	平均	IQR	中央値	6.6	5~8.3	7.3
平均	IQR	中央値																		
5.9	3.9~8.3	6.1																		
平均	IQR	中央値																		
5.6	3.9~7.8	5.6																		
平均	IQR	中央値																		
6.6	5~8.3	7.3																		
チームがサービスの運用パフォーマンスを改善するために行う活動と手法。	バージョン管理システムのメインコードブランチに定期的に統合される、小規模で頻繁な変更を行う手法。	弾力性があり、アクセスしやすく、測定可能で、スケーラブルなインフラストラクチャ ⁵ 。																		

⁴“What is Continuous Delivery” <https://continuousdelivery.com/>

⁵米国国立標準技術研究所 (2018) NIST The NIST Definition of Cloud Computing. <https://csrc.nist.gov/pubs/sp/800/145/final>

組織文化の要素

組織文化を定義することは簡単ではありませんが、一案として、職場において行き渡っている様式(柔軟性など)、一般的となっている志向(ユーザー中心など)、雰囲気(組織の安定性など)と表現できます。

 仕事の配分	 柔軟性	 雇用保障																		
<table border="1"><thead><tr><th>平均</th><th>IQR</th><th>中央値</th></tr></thead><tbody><tr><td>5.8</td><td>3.8~7.9</td><td>5.8</td></tr></tbody></table>	平均	IQR	中央値	5.8	3.8~7.9	5.8	<table border="1"><thead><tr><th>平均</th><th>IQR</th><th>中央値</th></tr></thead><tbody><tr><td>7.7</td><td>6.6~8.9</td><td>8.3</td></tr></tbody></table>	平均	IQR	中央値	7.7	6.6~8.9	8.3	<table border="1"><thead><tr><th>平均</th><th>IQR</th><th>中央値</th></tr></thead><tbody><tr><td>5.9</td><td>3.3~8.3</td><td>6.7</td></tr></tbody></table>	平均	IQR	中央値	5.9	3.3~8.3	6.7
平均	IQR	中央値																		
5.8	3.8~7.9	5.8																		
平均	IQR	中央値																		
7.7	6.6~8.9	8.3																		
平均	IQR	中央値																		
5.9	3.3~8.3	6.7																		
従業員がチーム内で公平にタスクを配分できる正式なプロセス。	いつ、どこで、どのようにタスクに取り組むか ⁶ 。	雇用保障が心配になる頻度を尋ねる単一項目の尺度。スコアが高いほど心配が少ないことを示します。																		
 組織の安定性	 知識の共有	 ユーザー中心																		
<table border="1"><thead><tr><th>平均</th><th>IQR</th><th>中央値</th></tr></thead><tbody><tr><td>7.2</td><td>6.7~8.3</td><td>8.3</td></tr></tbody></table>	平均	IQR	中央値	7.2	6.7~8.3	8.3	<table border="1"><thead><tr><th>平均</th><th>IQR</th><th>中央値</th></tr></thead><tbody><tr><td>6.4</td><td>5.0~8.3</td><td>6.7</td></tr></tbody></table>	平均	IQR	中央値	6.4	5.0~8.3	6.7	<table border="1"><thead><tr><th>平均</th><th>IQR</th><th>中央値</th></tr></thead><tbody><tr><td>7.8</td><td>5.6~8.3</td><td>7.8</td></tr></tbody></table>	平均	IQR	中央値	7.8	5.6~8.3	7.8
平均	IQR	中央値																		
7.2	6.7~8.3	8.3																		
平均	IQR	中央値																		
6.4	5.0~8.3	6.7																		
平均	IQR	中央値																		
7.8	5.6~8.3	7.8																		
従業員にとって職場環境がどの程度安定しているか、または不安定かを尋ねる単一項目の尺度。	アイデアや情報が組織全体に広がる仕組み。質問に答えたチームメンバーは、他のメンバーにも情報を共有します。答えを待つ必要がありません ⁷ 。	ユーザーのニーズや目標を理解して取り入れることで、プロダクトやサービスを向上させること ⁸ 。																		
 Westrum の組織類型																				
<table border="1"><thead><tr><th>平均</th><th>IQR</th><th>中央値</th></tr></thead><tbody><tr><td>7.3</td><td>6.1~8.6</td><td>7.8</td></tr></tbody></table>	平均	IQR	中央値	7.3	6.1~8.6	7.8														
平均	IQR	中央値																		
7.3	6.1~8.6	7.8																		
組織が問題や機会にどのように対応する傾向があるか。組織文化には、創造的、官僚的、不健全の3種類があります ⁹ 。																				

⁶ Shifrin, Nicole V., Jesse S. Michel. "Flexible work arrangements and employee health: A meta-analytic review." *Work & Stress* 36, no. 1, 2022.60-85

⁷ "2022 Developer Survey" <https://survey.stackoverflow.co/2022#overview>

⁸ Kersten, Mik. *Project to Product: How to survive and thrive in the age of digital disruption with the flow framework* (IT Revolution, 2018), 54. <https://itrevolution.com/product/project-to-product/>

⁹ Westrum R. "A typology of organisational cultures." *BMJ Quality & Safety*, 2004. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1765804/>

他社との比較

グッドハートの法則: 測定が目標になると、それは適切な測定ではなくなる¹。

要点

パフォーマンス改善の第一歩は、アプリケーションの現在のソフトウェア デリバリー パフォーマンス、運用パフォーマンス、ユーザー中心度のベースラインを設定することです。こうした尺度はチームの取り組みを評価するために役立ち、時間とともに状況がどのように変化しているのかを示す良いシグナルになります。

しかし、尺度はチームを改善するための手段ではありません。このベースラインに基づき、人員、プロセス、技術的能力にわたって幅広くチームの強さを評価し、進歩の妨げになっているものを特定することが重要です²。次に、チームには調整、テスト、見直しを行う時間と空間が必要です。このプロセスを繰り返すことで、チームに継続的改善の考え方と手法が定着します。

¹ Strathern, Marilyn (1997). "Improving ratings": audit in the British University system". *European Review*. John Wiley & Sons. 5 (3): 305–321. doi:10.1002/(SICI)1234-981X(199707)5:3<305::AID-EURO184>3.0.CO;2-4.S2CID 145644958.

² 本レポートおよび「能力カタログ」(<https://dora.dev/devops-capabilities/>)に記載されているリソースを参照。

³ Forsgren, N., Storey, M.-A., et. al. "The SPACE of Developer Productivity: There's more to it than you think." 2021. <https://queue.acm.org/detail.cfm?id=3454124>

こうした比較をする際は以下のような間違いを犯しやすいため、注意する必要があります。

- **ふさわしくない比較:** これらのクラスタのみに基づいてアプリケーションを比較しても、役に立ちません。アプリケーションごとのコンテキストが切り捨てられ、改善という目標のうえで有害になり得るからです。
- **指標を目標に設定する:** グッドハートの法則を無視し、「どのアプリケーションも年末までにエリートのパフォーマンスを示さなければならない」というような大雑把な宣言をすると、チームが指標をごまかそうとする可能性が高まります。
- **1つの指標ですべてを賄う:** 複雑なシステムを「重要な1つの指標」で測定しようとせず、複数の指標を組み合わせで理解を深めましょう³。
- **指標の範囲が狭い:** 最も意味のあるものではなく最も測定しやすいものを測定しがちです。
- **業界を盾に改善を渋る:** たとえば、規制の厳しい業界のチームが、現状を打破しない理由として規制を利用することがあります。

継続的改善の考え方と手法を推進する調査結果やアドバイスの詳細については、「[変革の方法](https://dora.dev/devops-capabilities/cultural/devops-culture-transform/)」(dora.dev/devops-capabilities/cultural/devops-culture-transform/)をご覧ください。

はじめに

アプリケーションに共通する傾向を把握するために、DORA は1つ以上のクラスタ分析を毎年行っています。比較の方法を理解するうえで、こうした分析を利用されることをおすすめします。ただし、比較に固執しないようにしましょう。比較は、コンテキストが決して一致しない異なるアプリケーション間で行うのではなく、同じアプリケーションで長期にわたって行うのが最善です。

チームはユーザーのためにソフトウェアを構築します。サービスの信頼性と有用性を最終的に判断するのはユーザーです。ユーザーのニーズを重視するチームほど、適切なものを構築できます。さらに、ユーザーを重視したソリューションをソフトウェア デリバリー パフォーマンス、運用パフォーマンスと組み合わせれば、そのチームは適切な方法で構築できるということになります。

ユーザーのニーズを重視する
チームは、適切なものを、
適切に構築します。

結果

ソフトウェア デリバリー パフォーマンス

ソフトウェア デリバリー パフォーマンスを評価するために、以下の尺度を使用します。

- **変更のリードタイム** - コードの変更を commit してからデプロイするまでの時間。
- **デプロイの頻度** - 変更を本番環境に push する頻度。
- **変更時の障害率** - デプロイにより障害が発生し、すぐに対処する必要が生じる頻度。
- **デプロイ失敗時の復旧までの時間** - デプロイの失敗時に復旧にかかる時間。

この4つの尺度すべてを改善するための一般的なアプローチは、アプリケーションの変更のバッチサイズを小さくすることです⁴。変更が小規模であれば、デリバリー プロセスを推測しやすくなり、実行しやすくなります。障害が発生した場合の復旧も容易です。チームはデリバリー プロセスを迅速かつ安定的なものにするために、各変更を可能な限り小さくする必要があります。このようにすることは、変更の速度と安定性の両方に寄与します。

⁴多くの場合、1つの機能を多数の変更に分割し、別々に提供できます。

ソフトウェア デリバリー パフォーマンスの尺度は、アプリケーションやサービスに加えられた変更を評価します。

今年、ソフトウェア デリバリー パフォーマンスの尺度を改良しました。変更点について詳しくは、[付録](#)の「ソフトウェア デリバリー パフォーマンスの測定方法の改良」をご覧ください。

ソフトウェア デリバリー パフォーマンスに関する今年のアンケート回答者の状況は次のとおりです。

パフォーマンスレベル	デプロイの頻度	変更のリードタイム	変更時の障害率	デプロイ失敗時の復旧までの時間	全回答者に占める割合
エリート	オンデマンド	1日未満	5%	1時間未満	18%
高	1日1回から週1回の間	1日から1週間の間	10%	1日未満	31%
中	週1回から月1回の間	1週間から1か月の間	15%	1日から1週間の間	33%
低	週1回から月1回の間	1週間から1か月の間	64%	1か月から6か月の間	17%



運用パフォーマンス

サービスで以下のことが発生する頻度を回答者に尋ねることで、運用パフォーマンスを評価しました。

- エンドユーザーがシステムの信頼性に不満を抱いているという報告を受ける。
- 利用できない、動作が想定よりも遅い、正しく動作しない。

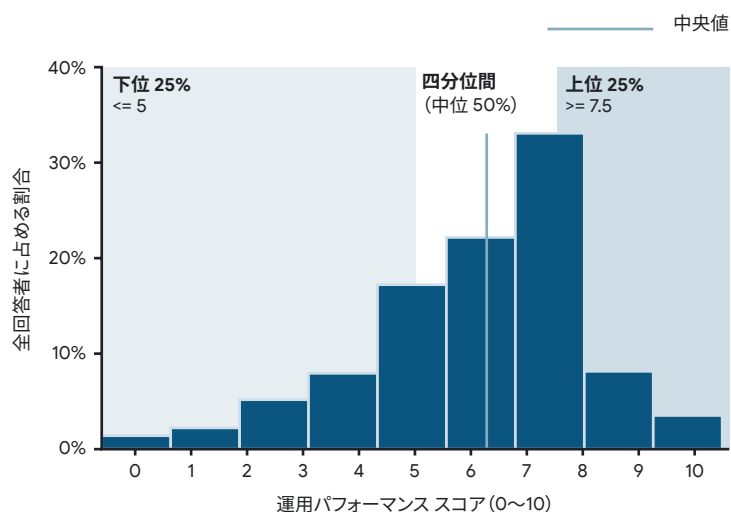
運用パフォーマンスから組織パフォーマンスをどのように予測するのかについては、[5章 - 信頼性がパフォーマンスを高める](#)をご覧ください。

ユーザー中心度

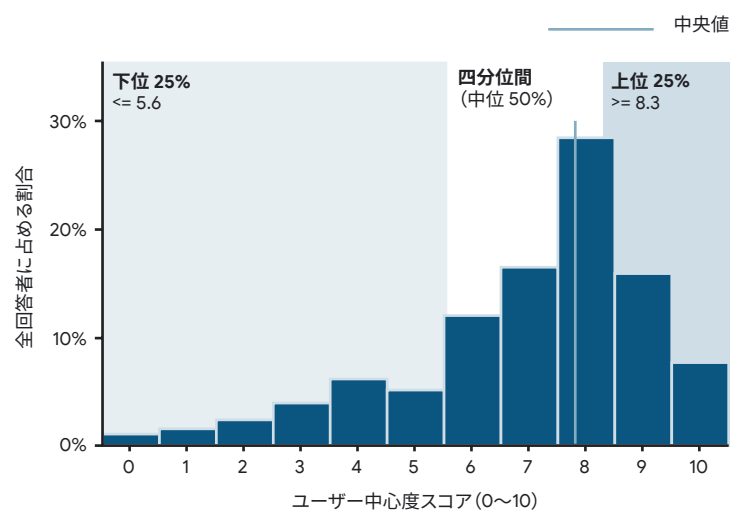
ユーザー中心のアプリケーションやサービスは、エンドユーザーを念頭に置いて構築されています。このようなプロダクトを構築するには、ユーザーのニーズをよく理解して、プロダクトのロードマップに組み込む必要があります。以下の各項にどの程度当てはまるかを回答者に尋ねることで、ユーザー中心かどうかを評価しました。

- ユーザーが達成したがっていることをチームが明確に理解している。
- チームの成功は、組織やアプリケーションユーザーに提供する価値によって評価される。
- 仕様(要件の計画など)は継続的に見直され、ユーザーシグナルに応じて再度優先順位付けされる。

運用パフォーマンスに関する今年のアンケート回答者の状況は次のとおりです。



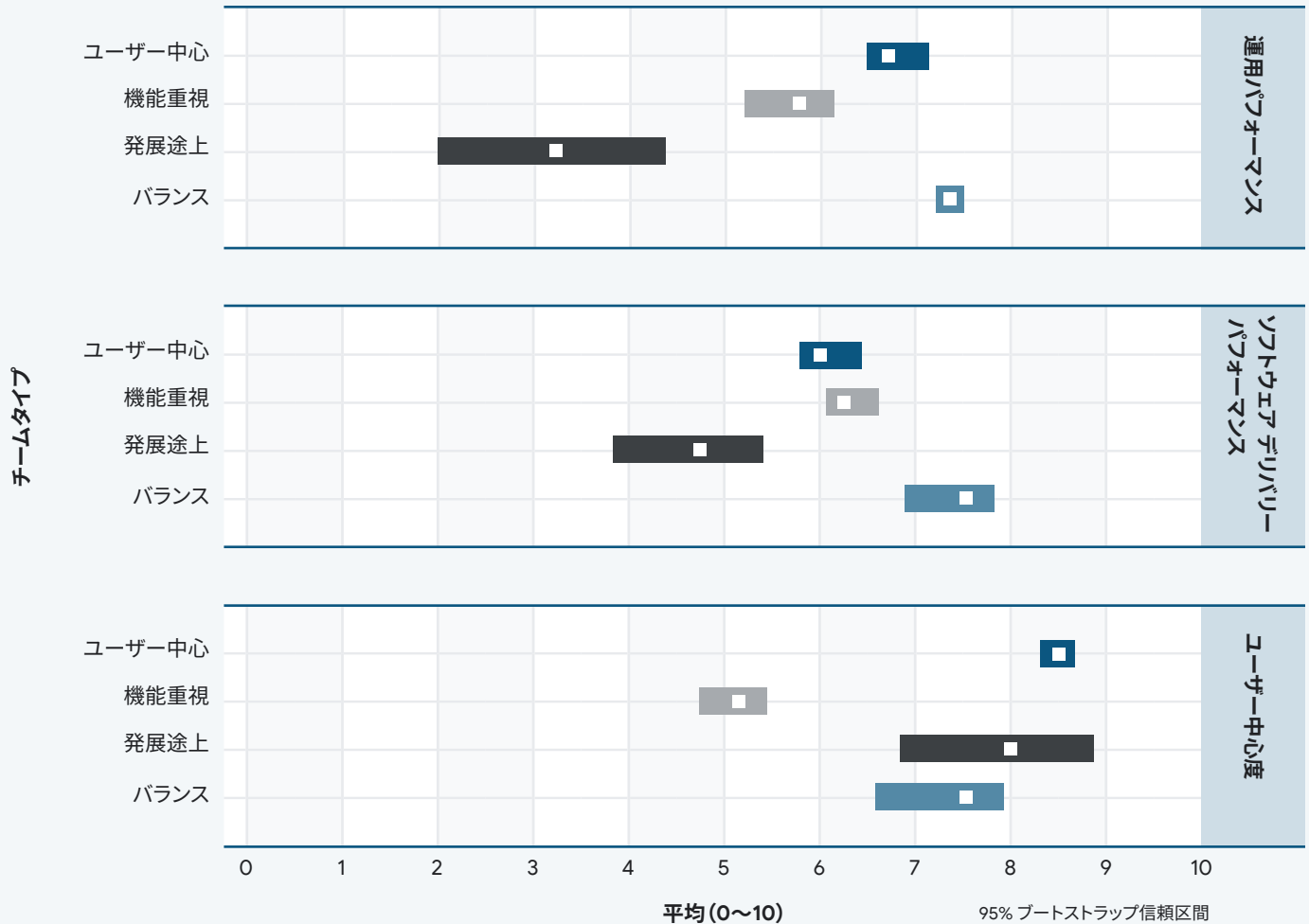
ユーザー中心度に関する今年のアンケート回答者の状況は次のとおりです。



チームタイプのクラスタリング

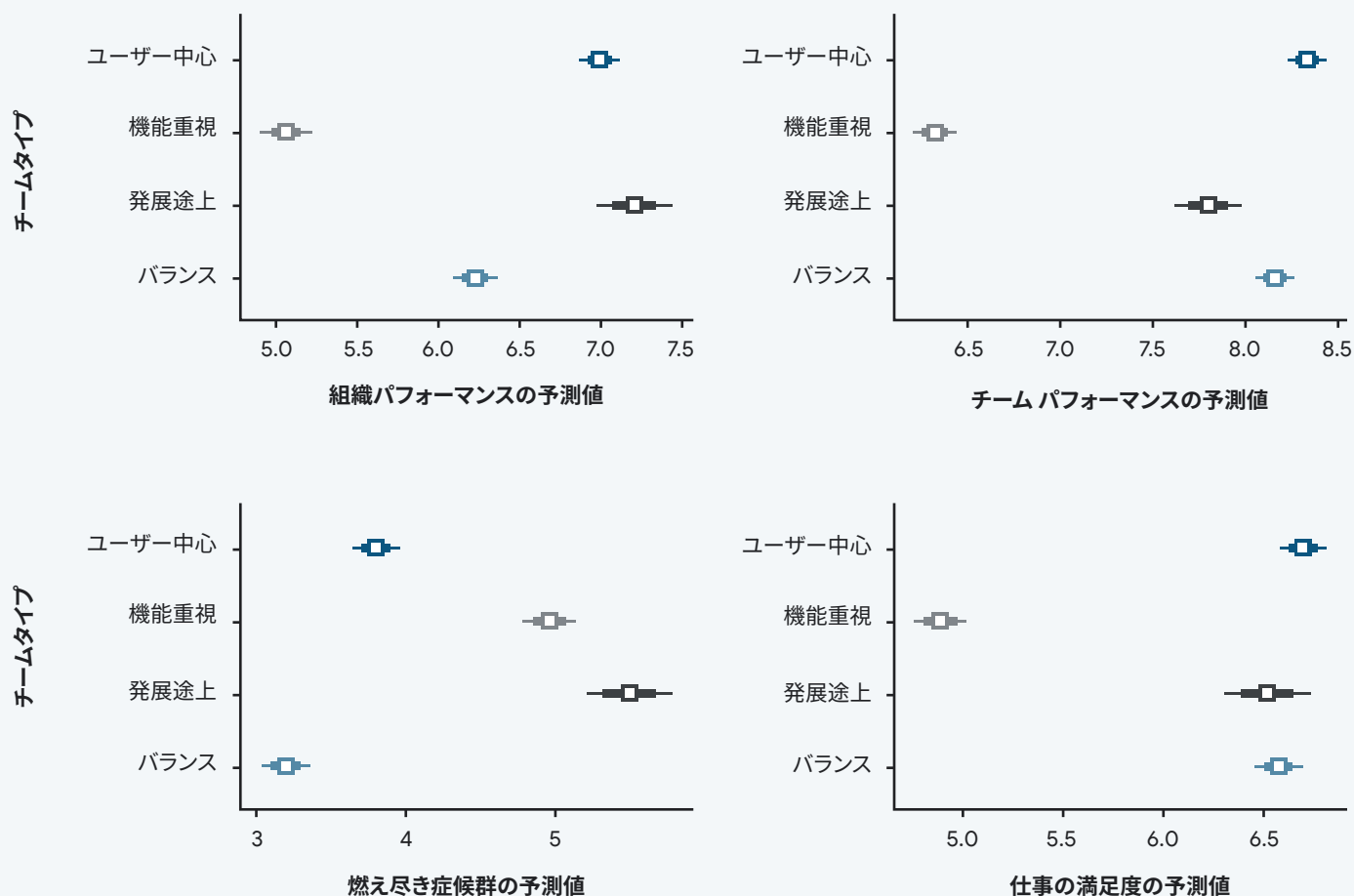
ソフトウェア デリバリー パフォーマンス、運用パフォーマンス、ユーザー中心度を単位として比較すると、チームが4種類に分かれます。こうしたチームタイプは、基となっているすべての尺度と同様に、アプリケーションレベルまたはサービスレベルです。

チームタイプを、ユーザー中心、機能重視、発展途上、バランスと名付けました。



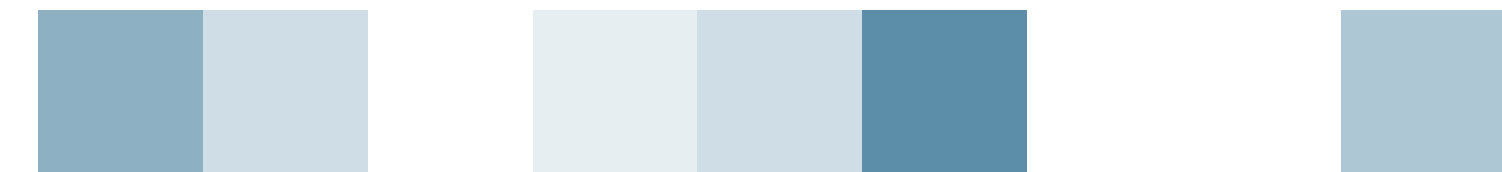
これまでに見てきたパフォーマンス指標は、組織パフォーマンス、チームパフォーマンス、チームの個人のウェルビーイングを変えるために組織やチームが調整できるダイヤルだと考えてください。

チームタイプごとに予測されるパフォーマンス成果のグラフを以下に示します。



* ドットはチームタイプ平均の点推定値を示しています。太線区間はシミュレーションの 66% が該当。細線区間はシミュレーションの 89% が該当。

チームタイプごとに、回答者のかなりの割合を占める固有の特徴があります。また、成果はそれぞれ異なります。おそらくチームがいずれかのタイプに完全に一致することはありません。また、該当するチームタイプが長期にわたって不変であるとは限りません。



結果からわかること



ユーザー中心チームタイプ

このチームタイプはユーザーのニーズを最も重視します。これが優れたソフトウェア デリバリー パフォーマンスや優れた運用パフォーマンスと相まって、最も高いレベルの組織パフォーマンスが予測されます。しかし、燃え尽き症候群がバランスチームより少し多くなっています。これらのチームで燃え尽き症候群を減らすには、ソフトウェア デリバリー パフォーマンスや運用パフォーマンスを改善することをおすすめします。



機能重視チームタイプ

このチームタイプは機能の提供を優先します。機能の提供を重視しすぎると、ユーザー中心度や運用パフォーマンスの数値の低さに表れているように、チームの注意がそれによってユーザーのニーズを満たせなくなる可能性があります。このようなチームタイプでは、燃え尽き症候群のレベルが非常に高く、仕事の満足度、チーム パフォーマンス、組織パフォーマンスが最も低くなっています。従業員は、機能だけでなく価値の提供にも価値を見出します。機能重視チームでは、提供する機能から価値をより多く引き出す手段として、ユーザーのニーズを熟考することをおすすめします。



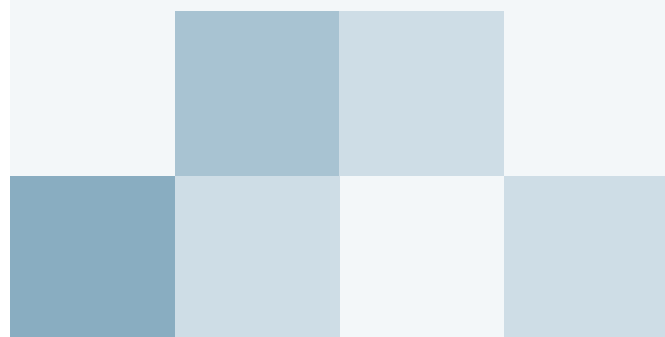
発展途上チームタイプ

このチームタイプは、アプリケーション ユーザーのニーズを重視して優れた組織パフォーマンスを達成します。しかし、プロダクトと市場の適合性や技術的能力の面で発展途上である場合があります。このようなチームは小規模な組織によく見られます。ソフトウェア デリバリー パフォーマンスと運用パフォーマンスは低く、こうしたアプリケーションに取り組んでいるチームは、バランスタイプやユーザー中心タイプのチームよりも燃え尽き症候群のレベルが高いと報告されています。このようなチームには、負担の大きなプロセスや手間のかかるタスクがあるかもしれません。これらを自動化することで、ソフトウェア デリバリー パフォーマンスと運用パフォーマンスを改善できる可能性があります。



バランス チームタイプ

このチームタイプは、バランスの取れたサステナブルなアプローチをしています。サステナブルな方法で技術を利用し、優れた組織パフォーマンス、チーム パフォーマンス、仕事の満足度を達成しています。このようなチームは、燃え尽き症候群のレベルが最も低いとも報告されています。また、3つの尺度すべてにおいて優れたパフォーマンスを示すように能力が調整されています。組織パフォーマンスをさらに向上させるには、ユーザー中心度を高めるとよいでしょう。



2章

ユーザー重視度から組織パフォーマンスを予測する

要点

組織は、開発者の生産性と組織パフォーマンスを改善しようとして、最新の技術や管理のトレンドにとらわれることがあります。DORA の調査では、ユーザー中心のアプローチでアプリケーションやサービスを構築しているかどうか、組織全体のパフォーマンスを予測するうえで特に重要な要素であることが明らかになっています。パフォーマンスを改善するために、ユーザーについて理解を深め、繰り返し調整を行い、フィードバックを取り入れましょう。

はじめに

DevOps ムーブメントは、より良いユーザー価値を提供するために、開発チームと運用チームの間の連携を向上させる方法として始まりました。この協力体制は早い段階で成功し、DevOps のアイデアと機能はこの2つの部署を越えて拡大していきました。現在、パフォーマンスの高い技術主導型組織は、組織の目標を達成するためにはすべてのチームにわたる連携が重要であることを認識しています。

私たちは、ユーザー重視に関して、次の3つの重要な特性を調査しました。

- チームがユーザーのニーズをどれくらい理解しているか。
- ユーザーのニーズを満たすためにチームがどれくらい足並みを揃えているか。
- 仕事の優先順位付けに際し、ユーザー フィードバックをどれくらい活用しているか。

ユーザー重視傾向の
強いチームは

40%

組織パフォーマンスが
高い

結果

ソフトウェア開発におけるユーザー中心のアプローチによって、パフォーマンスが有意に改善することがわかりました。ユーザーを第一に考えることで、組織はさまざまな恩恵を受けられます。ユーザー フィードバックは、チームがプロジェクトを優先順位付けするためにも、ユーザーのニーズを満たすプロダクトやサービスを生み出すためにも役立ちます。このアプローチは、ユーザー エクスペリエンスの向上、ユーザー満足度の向上、収益の増加につながります。

ユーザー中心の影響		
組織パフォーマンス		大幅に向上
チーム パフォーマンス		大幅に向上
ソフトウェア デリバリー パフォーマンス		わずかに向上
運用パフォーマンス		大幅に向上
トランクベース開発		大幅に向上
信頼性の手法		大幅に向上
継続的インテグレーション		大幅に向上
継続的デリバリー		大幅に向上
疎結合アーキテクチャ		大幅に向上
燃え尽き症候群		わずかに減少*
仕事の満足度		大幅に向上
生産性		大幅に向上

* 燃え尽き症候群の減少は良い兆候です

結果からわかること

ユーザーのニーズを重視しているかどうかは、組織全体のパフォーマンスを予測するうえで特に重要な要素です。ユーザー重視度を高めるには、インセンティブ、連携、業務の進め方を適切なものにする必要があります。ユーザーを重視することで、この調査の対象である技術、プロセス、組織文化のすべての能力にわたり、改善のヒントを得て実現していくことができます。

ここでは、こうした結果が組織全体のさまざまなチームにどのような影響を与えるかを探ります。

プロダクトの開発チームとデリバリー チーム
ユーザーを重視することで、プロダクトの開発チームとデリバリー チームはユーザーに適したものを（うまくいけばサステナブルな方法で）構築できます。バランス型のチームはまさにそうです。ユーザーのニーズを重視することで、デリバリー パフォーマンス、運用パフォーマンス、組織パフォーマンスにわたって強みを発揮しています。このようなチームのメンバーは、ユーザーのニーズを明確に理解することや、ユーザーフィードバックに基づいて計画を調整できる能力によってメリットを得ています。

機能重視型のチームは、組織パフォーマンスで最上位になれませんでした。このようなチームはデリバリーパフォーマンスを過度に優先し、組織パフォーマンスと組織の従業員のウェルビーイングの両方に弊害が出ています。

運用チーム
運用パフォーマンスを重視するチームは、CPU使用率などのシステム指標の最適化に懸命に取り組んでいるかもしれませんが、ユーザーがサービスに望んでいるものを理解していないと、動作が遅いというユーザー報告が頻繁に寄せられることになるで

しょう。ユーザーが重視するサービスレベル指標を特定し、典型的なユーザーを満足させるサービスレベル目標を定めるなど、サイト信頼性エンジニアリング (SRE) の手法は、運用チームがさらにユーザー中心の考え方を具体化するために役立ちます。



プラットフォーム エンジニアリング チーム

プラットフォーム エンジニアリング チームは、プラットフォームの構築にあたり「構築すればユーザーがついてくる」というアプローチをとることがあります。そうではなく、開発者をプラットフォームのユーザーとして扱うアプローチをとれば、成功しやすくなるでしょう。焦点をこのように移すためには、開発者が現在どのように仕事を進めているのかを把握し、問題の生じやすい部分をうまく特定して排除する必要があります。プラットフォーム構築の取り組みが良い成果につながっているかどうかをモニタリングするシグナルとして、ソフトウェア デリバリー パフォーマンスと運用パフォーマンスの尺度を使用できます。



リーダー

リーダーは、ユーザーに価値を提供したチームに報酬を与えるインセンティブの仕組みを設けることで、ユーザー重視が成功する環境を整えることができます。このような仕組みがないと、チームは単に提供した機能の数やサービス停止の減少を測定しているのだと感じる可能性があります。DORA は、変革型リーダーシップの役割を調査し、この能力を向上させたいと考えているリーダーに助言を行っています¹。詳細については、「DevOps の能力: 変革型リーダーシップ」(<https://dora.dev/devops-capabilities/cultural/transformational-leadership>) をご覧ください。

ユーザーを重視すると

20%

仕事の満足度が
高くなる

始めるためのリソース

ユーザー重視の能力を伸ばすことは、成功の重要な原動力です。2023 年の調査結果は、無駄のないプロダクト管理能力からソフトウェア デリバリー パフォーマンスと組織パフォーマンスを予測できるという 2018 年の調査結果を裏付けるものです²。

お客様のフィードバック³、バリュー ストリームでの作業の可視性⁴、小さいバッチ単位の作業⁵、チームでの試験運用⁶など、ユーザー重視の能力を取り入れてチームのパフォーマンスを強化しましょう。

¹ 2017 State of DevOps Report. <https://dora.dev/publications/pdf/state-of-devops-2017.pdf>, 12-19

² 2018 Accelerate: State of DevOps Report: ニュー エコノミーに向けた戦略. <https://dora.dev/publications/pdf/state-of-devops-2018.pdf>, 49-51

³ 「お客様のフィードバック」 <https://dora.dev/devops-capabilities/process/customer-feedback/>

⁴ 「バリュー ストリームでの作業の可視性」 <https://dora.dev/devops-capabilities/process/work-visibility-in-value-stream/>

⁵ 「小さいバッチ単位の作業」 <https://dora.dev/devops-capabilities/process/working-in-small-batches/>

⁶ 「チームでの試験運用」 <https://dora.dev/devops-capabilities/process/team-experimentation/>

技術的能力から パフォーマンスを予測する

要点

継続的インテグレーション、疎結合アーキテクチャ、コードレビューの迅速化にリソースと労力を投じれば、組織パフォーマンス、チームパフォーマンス、ソフトウェアデリバリーパフォーマンス、運用パフォーマンスの向上など、数多くの有益な成果につながります。その際、アプリケーションやサービスに取り組む個人のウェルビーイングを損なうことはなく、多くの場合はなんらかの利益をもたらします。

はじめに

「概要」では、調査した技術的能力と、それが種々のパフォーマンスやウェルビーイングの尺度にどのような影響を与えたのかを説明しました。DORAの中心的な要素は常に、さまざまなプロセスや技術的能力からパフォーマンスがどの程度予測されるのかを調査し、定量化することでした。

今年、以下の技術的能力からパフォーマンスがどのように予測されるのかを調査しました。

- AI
- トランクベース開発
- 疎結合アーキテクチャ
- 継続的インテグレーション
- 迅速なコードレビュー

これらから、以下のパフォーマンス尺度がどのように予測されたのかを確認しました。

- チームパフォーマンス
- 組織パフォーマンス
- ソフトウェアデリバリーパフォーマンス
- 運用パフォーマンス

さらに、こうした能力と複数の指標の関係をテストし、働いている人にどのような影響があるかを明らかにしました。

- 燃え尽き症候群
- 生産性
- 仕事の満足度

結果

調査した技術的能力とプロセスは、主なパフォーマンス尺度に対し、さまざまではありますが全体的に好影響を与えています。

技術的能力とプロセス	チームパフォーマンスへの影響	組織パフォーマンスへの影響	ソフトウェアデリバリーパフォーマンスへの影響	運用パフォーマンスへの影響
AI	⊖ 影響なし	↑ わずかに向上	↓ わずかに低下	↓↓ 大幅に低下
継続的インテグレーション	↑ わずかに向上	↑ わずかに向上	↑ わずかに向上	⊖ 影響なし
コードレビューのスピード	↑ わずかに向上	⊖ 影響なし	↑↑ 大幅に向上	↑↑ 大幅に向上
疎結合アーキテクチャ	↑↑ 大幅に向上	↑↑ 大幅に向上	↑ わずかに向上	↑↑ 大幅に向上
トランクベース開発	↑ わずかに向上	↑ わずかに向上	↑ わずかに向上	↓ わずかに低下

疎結合のチーム、あるいは疎結合アーキテクチャのあるチームは、他のチームを巻き込まずにシステムに大幅な変更を加えることができます。そのため迅速に行動できるようになります。対象分野の専門家がチームの近くにいれば、変更の影響をより詳しく理解しているため、コードを迅速にレビューしてもらうことができます。疎結合の設計では、他のチームがボトルネックになることなく、チームでテスト、ビルド、デプロイを行うことができます。

疎結合アーキテクチャでは、変更の影響が小さいとはいえ、チームの他の開発者が行っていることとの競合が発生しないようにする必要があります。小さいバッチで作業するチームでは競合の可能性が減り、commit ごとにソフトウェアがビルドされ、自動テストがトリガーされて、フィードバックが開発者へ迅速に提供されます。

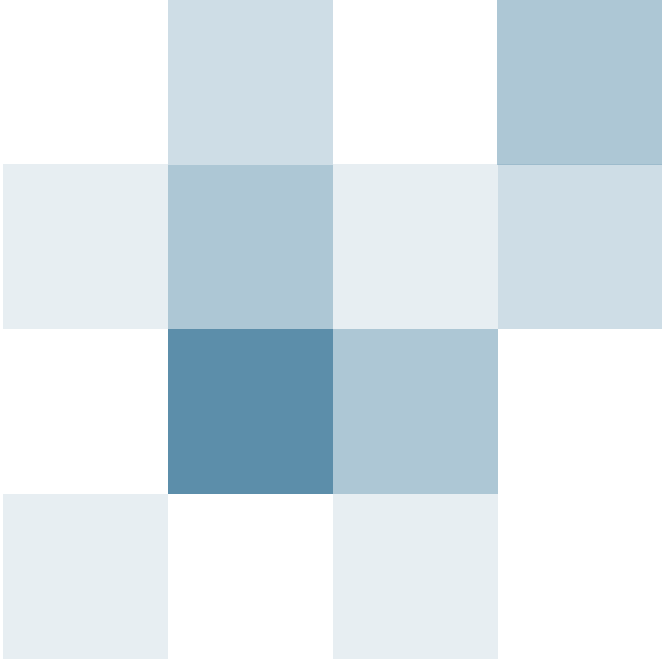
コードレビューにかかる時間が短いチームは、そうでないチームと比べてソフトウェアデリバリーパフォーマンスが50%高くなっています。効率的なコードレビュープロセスは、コードの改善、知識の伝達、コードの所有権の共有、チームの当事者意識、透明性につながります。

コードレビューがボトルネックになっていませんか？コードレビュー プロセスと、それが変更のリードタイムに与える影響を評価することで、改善の機会を探ることができます。次のことを検討してください。

- ピア コードレビューがプロセスに組み込まれているか
- コードの完成からレビューまでの期間
- コードレビューの平均バッチサイズ
- レビューに関与しているチームの数
- レビューに関与している地域の数
- チームがコードレビューの提案に基づいてコード品質の自動化を改善しているか

コードの完成からレビューまでの期間が長いと、開発者の作業効率や提供するソフトウェアの品質に悪影響が及ぶことがわかっています。地理的に離れている複数のチームが関与すると、期間が長くなり、プロセスへの関わりが浅くなり、費用が増えます¹。

2022 年に、技術的能力には相乗作用があることが判明しました。コードレビューのスピードを改善すると、コードの保守性、学習文化（知識の伝達）、創造的な文化の構築など、いくつかの技術的能力の向上につながる可能性があります²。



コードレビューの迅速化は疎結合チームのメリットの一つであり、ソフトウェア デリバリー パフォーマンスと運用パフォーマンスの大幅な改善につながります。コードレビューの効率を上げる方法は複数あります。レビューするコードがチームのアーキテクチャの範囲にしか影響しない場合、レビュー担当者はコードがシステムに与える影響をよく理解できます。コードレビューの規模が小さいほど、レビュー担当者が変更の影響を理解しやすくなります。小さいバッチで作業すると、チームのフィードバック サイクル、効率、集中力が向上します³。ペア プログラミングでは、現在のアーキテクチャや統合の手法に関係なく、コードレビューにかかる時間を短縮できます⁴。

さらに、このような能力やプロセスは、働いている人のウェルビーイングに悪影響を与えません。むしろ、ほとんどは個人のウェルビーイングを向上させると予測されています。

¹ Investigating the effectiveness of peer code review in distributed software development based on objective and subjective data. <https://jserd.springeropen.com/articles/10.1186/s40411-018-0058-0>

² Expectations, Outcomes, and Challenges of Modern Code Review. <https://dl.acm.org/doi/10.5555/2486788.2486882>

³ 「小さいバッチ単位の作業」 <https://dora.dev/devops-capabilities/process/working-in-small-batches/>

⁴ “On Pair Programming.” <https://martinfowler.com/articles/on-pair-programming.html>

技術的能力とプロセス	燃え尽き症候群への影響*	仕事の満足度への影響	生産性への影響
AI	↓ わずかに減少	↑ わずかに向上	↑ わずかに向上
継続的インテグレーション	○ 影響なし	↑ わずかに向上	○ 影響なし
コードレビューのスピード	⇄ 大幅に減少	↑ わずかに向上	↑ わずかに向上
疎結合アーキテクチャ	⇄ 大幅に減少	⇄ 大幅に向上	⇄ 大幅に向上
トランクベース開発	⇄ 大幅に増加	○ 影響なし	○ 影響なし

* 燃え尽き症候群はカラーパターンが反対になっています。燃え尽き症候群は減少する方が良いからです。

疎結合アーキテクチャ、継続的インテグレーション、効率的なコードレビューを利用することで、チームはウェルビーイングを維持し、ときには向上させながら、組織の成果を高めることができます。

ユーザーに価値を提供する信頼性の高いシステムを改善し維持するための自由裁量がチームにあれば、仕事の満足度、チームパフォーマンス、ソフトウェアデリバリーパフォーマンスが改善されます。アーキテクチャは、チームがユーザーを重視してソフトウェアデリバリーを改善するにあたり、重要な役割を果たします。小規模な構成から始めてユーザーを重視することで、チームはトランクベース開発、疎結合アーキテクチャ、継続的インテグレーション、継続的デリバリー、SREで大幅な改善を達成しました。技術的能力を向上させるために、チームでの試験運用と継続的改善の機会を設けましょう⁵。

⁵「チームでの試験運用」<https://dora.dev/devops-capabilities/process/team-experimentation>



継続的デリバリーのメリット

著者: Dave Farley

継続的デリバリー (CD) の基本原理は、ソフトウェアが常にリリース可能な状態になるように作業することです。そのためには質の高い作業をする必要があります。そうすれば、問題を発見したときに簡単に修正できるため、リリース可能な状態に素早く戻すことができます。

ソフトウェアをリリース可能な絶好の状態に保つには、迅速なフィードバックを確立し、障害からすぐに復旧するように取り組む必要があります。

今年のレポートを読んだ方は、この考え方に覚えがあると思います。**安定性**の指標 (変更時の障害率、デプロイ失敗時の復旧までの時間) はすべて品質に関するものであり、**スループット**の指標 (変更のリードタイム、デプロイの頻度) はすべて、フィードバックと、問題のを見つけやすさに関するものです。

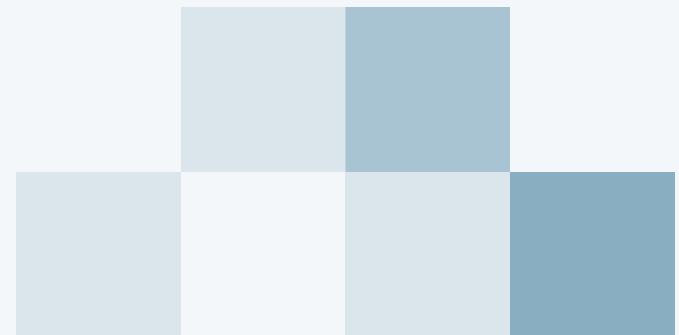
CD を実践すれば、**安定性とスループット**のスコアが高くなります。**安定性とスループット**のスコアが高い場合、その高いスコアの背景にはおそらく CD の実践があると考えられます。

今年の分析では、各能力を媒介する要素、つまりメディエーターを探すことで、能力がどのようにパフォーマンスを向上させているのかも調べています。CD (あらゆる変更をオンデマンドで迅速、安全、サステナブルにリリースする機能) は、多くの技術的能力の重要なメディエーターです。つまり、これらの能力が機能するのは、CD を可能にする環境がこれらの能力によって整えられるからです。そして、CD の実践は、これらの能力から優れたソフトウェア デリバリー パフォーマンスを予測できるメカニズムを実現します。

リリース可能性はソフトウェア開発全般で満たすべき重要な基準であり、だからこそ CD においても重視されています。リリース可能性が重要なのは、主観的ではあっても明確な、コンテキストに応じた品質の表現だからです。リリース可能性を定義する厳密さの程度は、セーフティクリティカルシステムに取り組んでいる場合と、ケーキ店のソフトウェアを作成している場合とでは異なるでしょう。しかしどちらの場合も、リリース可能性とは、コードがユーザーにリリースできる状態であり、十分に良質で、十分に安全であるというために必要と思われる作業はすべて行ったということを示します。

そのため、変更をリリース可能な状態に維持するための最適化は、システムで許容できる最低限の品質に関するコンテキスト固有の定義の状態を目指した最適化でもあります。

品質の高い迅速なフィードバックを得て、それに基づいて行動することを優先するチームは、ソフトウェア デリバリー パフォーマンスが高くなります。



継続的インテグレーション(CI)とトランクベース開発がソフトウェア デリバリー パフォーマンスに大きな影響を与えなかったことには、少々驚きました。特にCIは私にとってかなり基本的に思え、考え方を揺るがされていますが、これらの関係を解き明かすのは複雑です。たとえば、コードを統合せずにスループットで高いスコアを達成できるわけがありませんし、安定性を確認せずに安定性が高いと確信できるわけがありません。私にとってはCIが、こうしたことを確認する方法であり、ソフトウェア デリバリー パフォーマンスの重要なメディエーターとなっています。これは解釈の問題なのでしょうか。それとも、もっと深い、重要なことなのでしょう。興味をそそります。

技術的能力とプロセス	ソフトウェア デリバリー パフォーマンスへの影響	継続的デリバリーによるメディエーション*
AI	↓ わずかに低下	⊗ なし
継続的インテグレーション	↑ わずかに向上	☑ 完全
コードレビューのスピード	↑↑ 大幅に向上	☑ 部分的
疎結合アーキテクチャ	↑ わずかに向上	☑ 部分的
トランクベース開発	↑ わずかに向上	☑ 完全

* メディエーションは、影響の基礎となり得るメカニズムや経路を評価するテストです。たとえば、「ソフトウェア デリバリー パフォーマンスに対するトランクベース開発の影響は継続的デプロイ(メディエーター)を通じて生じる、という仮説をこのデータは裏付けている」と言うことができます。影響全体がメディエーターによって説明されるような場合を、完全なメディエーションといいます。影響の一部のみがメディエーターによって説明される場合は、部分的なメディエーションといいます。

組織のプロセスと能力の最適化

組織文化が成功の源になることはわかっています。では、組織文化の源になるのは何でしょうか？これは興味深い問いですが、おなじみの答えになります。つまり、場合によります。

実務者の観点では、日常業務の進め方を改善することが、リスクの共有、協力体制の強化、心理的安全の確立といった組織文化の要素に好影響を与えます。たとえば、変更をバージョン管理システムのメインブランチに定期的に統合すると、知識の共有と連携が促進されます。また、セキュリティチームが開発者とともに作業し、「コードとしてのポリシー」のために連携すると、チーム間の信頼が深まり、デプロイする変更への自信が増します。

リーダーシップの観点では、組織文化は、組織文化の重要性についての認識と教育から始まります。変革型リーダーシップ⁶は、試行錯誤と学習を奨励し、実務者に信頼と発言力を与える、誰も責めない環境を醸成するために役立ちます。エンジニアは複雑な問題を解決するためにいるのであって、単にタスクの要求に応えるためにいるわけではありません。そのためには、ビジネスに対する可視性と、行動を起こすための自律性が必要です。結局、組織文化はリーダーシップの影響を強く受けます。

理想的には、トップダウンとボトムアップという両方の観点から組織文化を捉えることで、最良の結果が得られます。

⁶「変革型リーダーシップ」 <https://dora.dev/devops-capabilities/cultural/transformational-leadership>

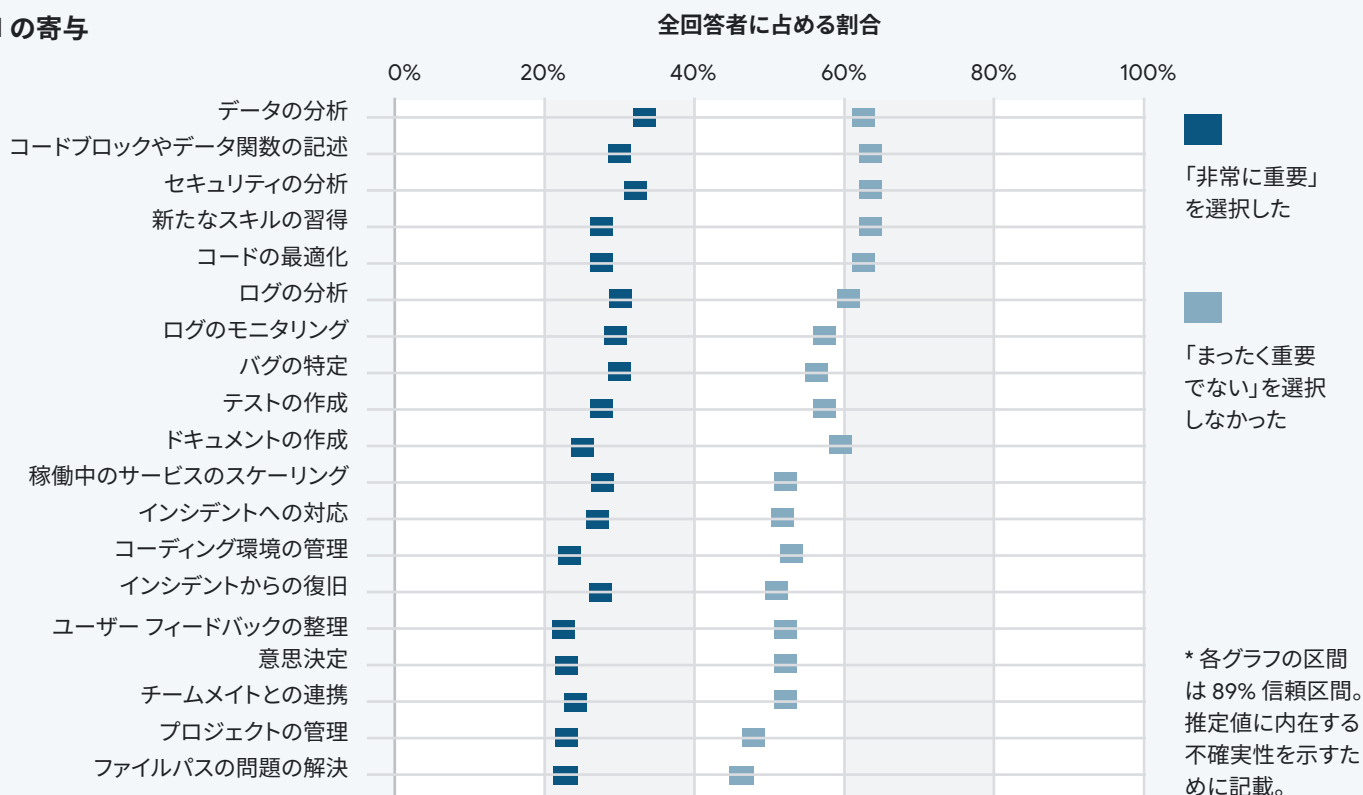
AI

アナリストやテクノロジストの中には、AI が仕事上のウェルビーイングに悪影響を与えずにソフトウェアチームのパフォーマンスを向上させる、という仮説を立てている人もいます。これまでのところ、私たちの調査結果はこれを裏付けるものではありません。DORA の調査結果では、AI は個人のウェルビーイングの尺度(燃え尽き症候群、仕事の満足度など)をわずかに改善するものの、グループレベルの成果(チーム パフォーマンス、ソフトウェア デリバリー パフォーマンスなど)には影響を与えないか、場合によっては悪影響を与えることが示されています。

この混在した調査結果は、企業の AI 搭載ツールの導入が初期段階にあることによって説明できると考えられます。おそらく一部の大企業は、さまざまな AI 搭載ツールを試用してから、広く使用するかどうかを判断していることでしょう。調査で取り上げられたタスクに、大半の人が少なくともなんらかの AI を取り入れていることからわかるように、AI を活用した開発ツールに対する期待は高まっています。これは下のグラフに表れています。しかし、AI 搭載ツールが業界全体で広く調和した形で使われるようになるまでには、まだ時間がかかると DORA は考えています。

AI の重要性 技術タスクに対する AI の寄与

主なアプリケーションまたはサービスで、以下の各タスクへの寄与という点から、AI の役割は現在のどの程度重要ですか。



今後どのように導入が拡大していくのか、組織が重視するパフォーマンス指標や成果にそれがどう影響するのかについて、DORA は強い関心を寄せています。

4章

ドキュメントが基礎となる



要点

質の高いドキュメントが基礎となります。技術的能力を実際に活かすための原動力となり、組織パフォーマンスに対する能力の影響を増幅します。ドキュメントは、チームパフォーマンス、生産性、仕事の満足度などの成果にも好影響を与えます。しかし、ドキュメントの質を高めれば全員のウェルビーイングが向上するというわけではありません。ドキュメントの質が高まるにつれて燃え尽き症候群のレベルが高まると報告する回答者もいます。

はじめに

今年は内部ドキュメント（組織内の人員が日常的に使用する、知識を文書化したもの）を詳しく取り上げ、ドキュメントが技術的能力と主な成果に与える影響を調査しています。

ドキュメントの質を測定するために、ドキュメントの信頼性、検索性、最新性、関連性の程度を測定しました。そして、ドキュメントの全体的な使用感について1つのスコアを算出しました。ドキュメントを1ページずつ評価するのではなく、全体として評価しています。

結果

ドキュメントが基礎となる: 技術的能力の原動力となり、影響を増幅する

2021年¹と2022年²の調査結果と同様に、ドキュメントの質は、調査対象のすべての技術的能力を適切に活かす原動力となっています。

次の表に示すとおり、ドキュメントの質は2022年の調査結果³と同様に、各技術的能力が組織パフォーマンスに与える影響を増幅してまいります。

技術的能力	組織パフォーマンスへの影響の増幅*
継続的インテグレーション	2.4 倍*
継続的デリバリー	2.7 倍*
トランクベース開発	12.8 倍*
疎結合アーキテクチャ	1.2 倍*
信頼性の手法	1.4 倍*
AIの寄与	1.5 倍*

* 計算式: $\frac{\text{ドキュメントの質が高い場合の技術的能力の影響}}{\text{ドキュメントの質が低い場合の技術的能力の影響}}$

¹ Accelerate State of DevOps 2021. <https://dora.dev/publications/pdf/state-of-devops-2021.pdf>

² 2022 Accelerate State of DevOps Report. <https://dora.dev/research/2022/dora-report/2022-dora-accelerate-state-of-devops-report.pdf>

³ 2022 State of DevOps Report データの詳細: ドキュメントと太陽の光の類似点. <https://cloud.google.com/blog/ja/products/devops-sre/deep-dive-into-2022-state-of-devops-report-on-documentation>

ドキュメントから生産的かつ幸福な個人と組織を予測する

質の高いドキュメントは技術的能力を向上させるだけでなく、個人のウェルビーイングにも好影響を与えることがわかりました。燃え尽き症候群が減り、仕事の満足度が高まり、生産性が上がります。この影響の一部は、質の高いドキュメントによって知識の共有が促進されるためであることがわかりました。

これはそれほど驚くことではありません。方法がわかっているならば何かを成し遂げるのは簡単になりますし、知識が共有されていけば仕事の不満も減ります。

ウェルビーイングの要素	質の高いドキュメントの影響
燃え尽き症候群	 大幅に減少
仕事の満足度	 大幅に向上
生産性	 大幅に向上

質の高いドキュメントはチームパフォーマンス、組織パフォーマンス、運用パフォーマンスにも影響を与え、主な成果を後押しします。

主な成果	質の高いドキュメントの影響
チームパフォーマンス	 大幅に向上
組織パフォーマンス	 大幅に向上
ソフトウェアデリバリーパフォーマンス	 影響なし*
運用パフォーマンス	 大幅に向上

* 質の高いドキュメントがソフトウェアデリバリーパフォーマンスに影響を与えていないのは、やはり驚きです。この性質が確認されたのは昨年に続いて2回目です。

3つの主な成果に対する好影響の裏には何があるのでしょうか。読者として、明確なドキュメントを利用することは有益です。記述プロセスが一因である可能性もあります。質の高いドキュメントを作成するには、そもそもチームがプロセスを決定する必要があります。ドキュメントを用意することで、組織全体のチームが行うべきこととその方法について明確に議論し、足並みを揃えることができます。

また、質の高いドキュメントは、人員の入れ替わりがあってもチームの知識のリポジトリとして機能します。組織全体へ、時間を超えて知識を拡大していくために役立ちます。

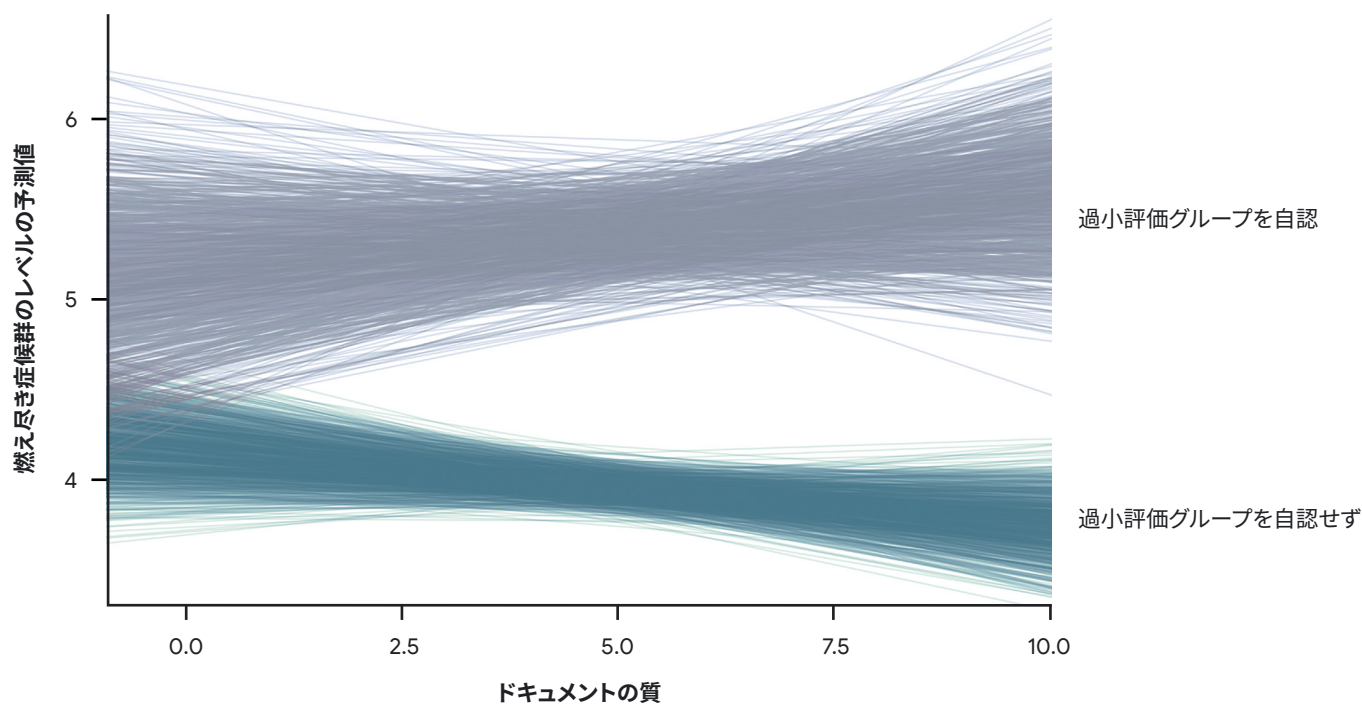
ドキュメントとウェルビーイングの低下との関連

過小評価グループを自認する回答者を調べたところ、予想外の傾向があることがわかりました。このグループでは、ドキュメントの質が燃え尽き症候群の増加と結びついていたのです。

この結果についてジェンダーも調べましたが、影響がないことに驚きました。男性や女性を自認する回答者でも、ジェンダーを自己決定する回答者でも、質の高いドキュメントで燃え尽き症候群が有意に減少しました。しかしジェンダー アイデンティティに関係なく、過小評価グループを自認する回答者では、質の高いドキュメントがある場合に燃え尽き症候群の割合が高くなりました。

下のグラフは、データに基づく予測シミュレーションです。下の集団では、回答者の大多数について、ドキュメントの質が上がるにつれて燃え尽き症候群が減っていくことがわかります。しかし上の集団では、過小評価グループを自認する回答者について、燃え尽き症候群が大幅に増えていることがわかります。

このグラフには、シミュレーション結果の線をグループごとに1,000本ずつ示しています。線が密集しているほど、データに基づく傾きの信頼性が高いということになります。



この結果は、ドキュメントの質、創造的な組織文化、チームの安定性について同様でした。過小評価グループを自認する回答者について、これらの属性が高くなるにつれ、燃え尽き症候群も増えています。ドキュメントに関して何が起きているのでしょうか。

質の高いドキュメントを作成して維持するには労力を要します。これは技術面の労力であり、技術的能力、チームの生産性、組織パフォーマンスに大きな影響を与えます。また、その重要性や影響力が一貫して認識されていない可能性がある労力であるともいえます。この労力は、過小評価グループを自認する人が極端に多く負担しているのでしょうか。そうであれば、この労力は燃え尽き症候群に対する影響を説明するために役立つのでしょうか。

ドキュメントの利用に頼ることに問題があるのでしょうか。ドキュメントの質が上がると、一部の回答者で知識の共有が増えないということなのでしょうか。あるいは、知識の共有が増えても、このグループの燃え尽き症候群につながる他の要素を打ち消すには不十分だということなのでしょうか。

ドキュメントの質を高めるだけでなく、過小評価グループを自認する回答者の燃え尽き症候群を引き起こす、あるいは燃え尽き症候群からの回復を妨げる、まったく別の何かが作用している可能性もあります。さらなる調査が必要です。

チームに所属する個人の属性は、重要な意味を持つようです。質の高いドキュメントといった職場の要素は、チームや組織全体に大きな利益をもたらします。しかし、人によっては望ましくない結果が生じることもあります。この件について詳しくは、[8章 - 個人の属性が重要になる状況と理由](#)で説明します。

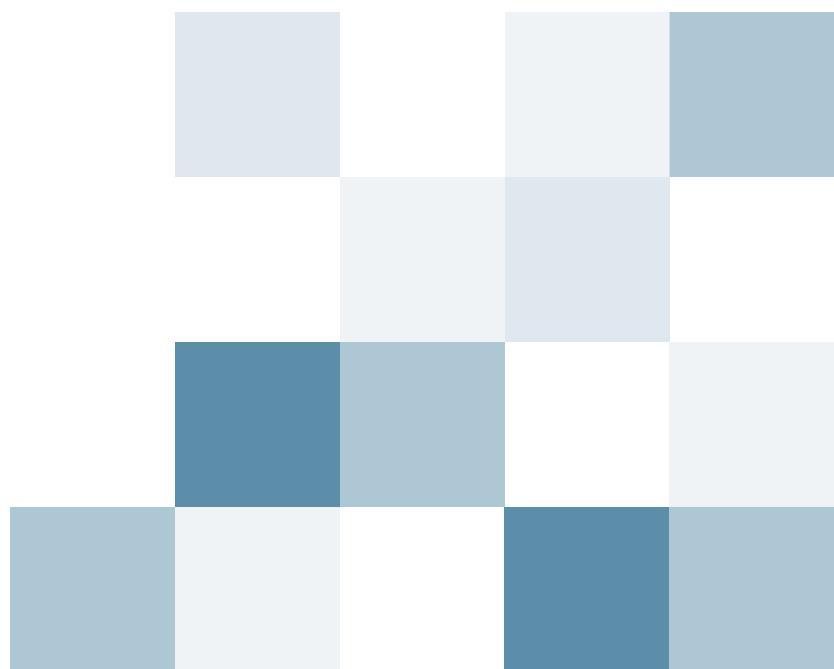
⁴ Accelerate State of DevOps 2021, 22. <https://dora.dev/publications/pdf/state-of-devops-2021.pdf#page=22>

始めるためのリソース

ドキュメントの質を高める手法については、2021年のレポートをご覧ください⁴。今年は、ドキュメントに関する作業を分担する正式なプロセスなど、仕事の配分によってドキュメントの質が大幅に高まることもわかりました。

テクニカルライティングのためのリソースやトレーニングはたくさんあります。詳しくは以下のリソースをご覧ください。

- Society for Technical Communications (stc.org)
- エンジニアのためのテクニカルライティングコース (developers.google.com/tech-writing)
- Write the docs (writethedocs.org)



信頼性がパフォーマンスを高める

要点

信頼性の手法が優れていると、運用パフォーマンス、チームパフォーマンス、組織パフォーマンスの向上が予測されます。データによると、信頼性の手法を改善することによる影響は非線形の経路をたどります。つまり、組織が能力を強化していく中で、パフォーマンスの向上が停滞しているように見えるときがあるかもしれません。それでも長い目で見れば、こうした手法に取り組み続けることで良い結果につながります。

はじめに

信頼性という用語は IT 運用の分野で広く使われています。DORA は信頼性を、可用性、パフォーマンス、正確性などの尺度に関し、定められた目標をサービスが達成している程度と定義しています。信頼性に関する成果を達成するための一般的なアプローチは SRE です。これは Google 発祥で (<https://sre.google>)、今では多くの組織で実践されています。SRE では、経験主義の学習、部門横断的な協力、自動化の積極的利用、サービスレベル目標 (SLO) などの測定技法の使用を重視します。

多くの組織が、SRE と呼ばずに信頼性の手法を使用しています。代替の用語としては、プロダクション エンジニアリング、プラットフォーム チーム、インフラストラクチャ チーム、TechOps などがあります。こうした手法が採用されている程度を可能な限り客観的に評価するために、本調査の文章には中立的で説明的な表現を使用しています。

また、信頼性エンジニアリングの成果、すなわちチームが信頼性目標を達成できる程度についてもデータを収集しています。信頼性の手法と信頼性の成果 (**組織パフォーマンス**と いいます) は、どちらも他の能力とともに予測モデルに反映されています。

信頼性の手法

回答者に、運用において重要な 3 つの側面から信頼性について考えてもらいました。1 つ目は、依存関係の緩和策があるかどうか。2 つ目は、障害のシミュレーション、実践的なフェイルオーバー、机上演習のいずれかを通じて、障害復旧計画を定期的にテストしているかどうか。3 つ目は、信頼性目標が達成できなかった場合に、改善作業、または作業の優先順位の見直しや作業の調整を実施しているかどうかです。

DORA はこれらの尺度を、「リスクの受容」や「ユーザーの幸福度の測定」などの確立された SRE の原則に沿おうとするチームの精神を表すものだと考えています。そのようなチームは、ユーザーの幸福度と方向性の揃った、合理的な目標を設定します。そして、目標を達成できるようにするためにテストを実施し、問題があれば計画を変更します。DORA は、チームの評価を特定の SRE 実装と結び付けることなく、適切に「SRE を行っている」チームを表すものとしてこれを使用しています。

結果

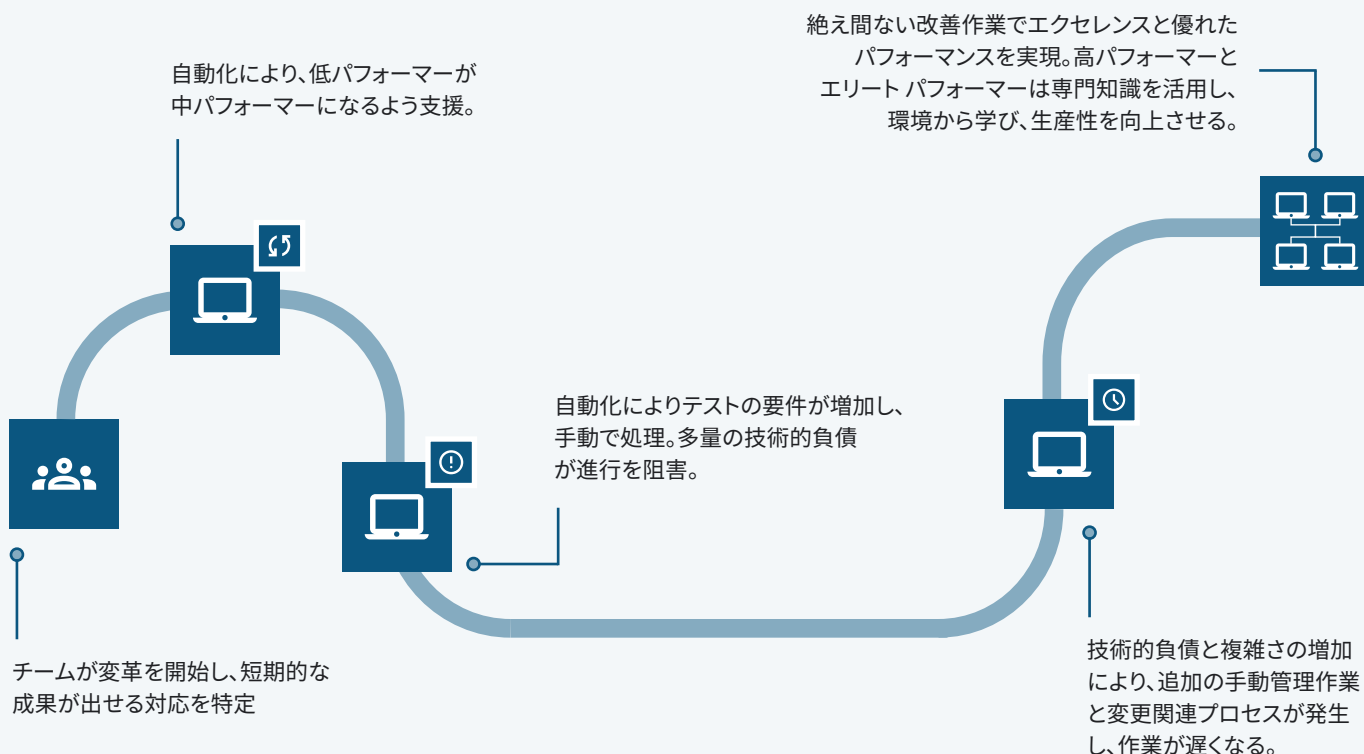
信頼性の手法の J 曲線を確認

DORA は 2018 年以来、運用パフォーマンスと自動化のような手法の間には非線形の関係 (図 1) があるという理論を展

開してきました。信頼性の手法についての探求を深めるにつれ、調査データにこのパターンの裏付けが見られるようになりました。

2022 年にはこれを直接測定しました。チームを調査し、信頼性の手法と信頼性の成果の関係が確かにこのような非線形曲線 (図 2) となることを確認しました。これは、チームが信頼性の手法を多数採用して初めて信頼性の大幅な向上が見られたことを示唆しています¹。しかし、このようにしてデータを見るだけでは、全体像を見ているようには感じられませんでした。2022 年の曲線は、SRE が専門家だけのものであるような、あるいは投資するほどの価値がないものであるような印象を与えるものでしたが、これは多くの SRE チームの経験と矛盾します。もっと多くのデータが必要でした。

図 1: 2018 年の仮説の J 曲線



¹ 2022 Accelerate State of DevOps Report.27-28. <https://dora.dev/research/2022/dora-report/2022-dora-accelerate-state-of-devops-report.pdf#page=27>

図 2: 2022 年の曲線

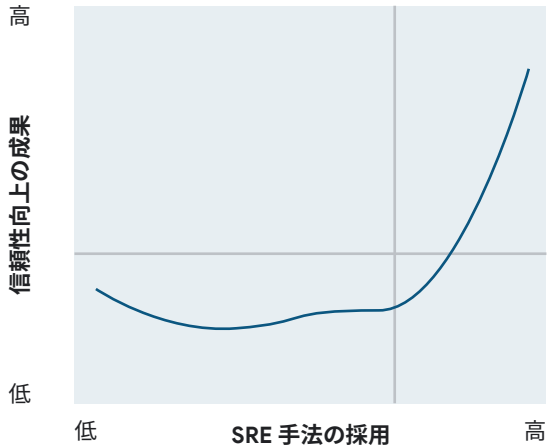
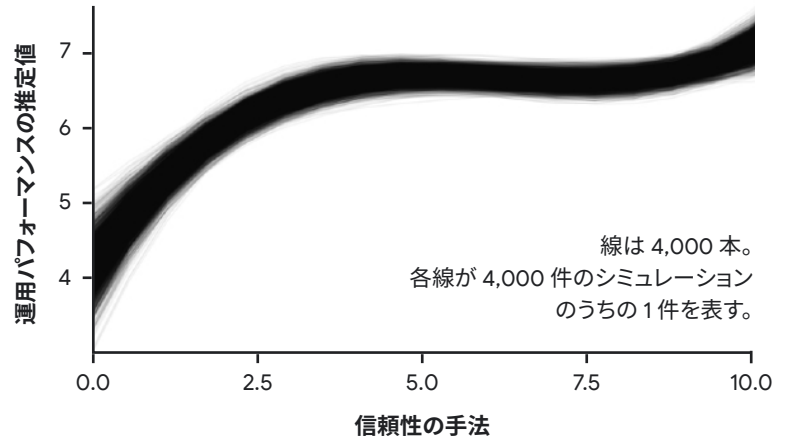


図 3: 2023 年の曲線



2023 年は質問項目を増やすことで、実体験に一層近い曲線を明らかにすることができました。新しい曲線は 2018 年のレポートに記載した変革の仮説の J 曲線に近くなっています (分析方法について詳しくは [方法論](#) セクションをご覧ください)。これは、信頼性の手法の採用ですぐに成果が出て、その後、複雑さが新たな課題をもたらすにつれて小康状態になり、最終的には運用パフォーマンスがもう一度上がることを示唆しています。今回の結果は多くのチームで見られたことを裏付けるものです。

この曲線が重要な理由はいくつかあります。

- 企業が極度に高い信頼性を求めている場合や、SRE への多大な投資を想定していない場合であっても、初期の SRE 採用を正当化し、資金を投じることができます。信頼性の手法を少し採用するだけでも運用パフォーマンスの向上につながり、それによってチームパフォーマンスと組織パフォーマンスにも有益な影響が現れます。
- 信頼性に多額の投資を行おうとしている企業が小康状態を耐え抜くための準備になります。SRE への長期的な投資に比例して望ましい結果が得られることを期待したくなるかもしれませんが、データによると、そうはなりません。チームがこの曲線の非線形性を前もって知っていれば、投資を行うかどうかを判断でき、十分な利益を得る前に投資を中断することのないよう、事前に計画を立てることができます。
- このような変化には組織文化の変革が必要かもしれません²。ボトムアップとトップダウンの組み合わせによって成功がもたらされることがわかりました。チームが信頼性の手法を採用してすぐに成果を獲得すれば、その成果を他のチームに示すことができます。リーダーシップは、これをサポートしインセンティブを与えることができます。このようなインセンティブと体系化されたプログラムは、J 曲線を念頭に置いて設計できます。

²「変革の方法」 <https://dora.dev/devops-capabilities/cultural/devops-culture-transform/>

信頼性の手法とウェルビーイング

従来型の運用手法は非常に事後対応的であり、多くの場合、ユーザーの幸福度よりも技術システムの健全性を重視します。ユーザー エクスペリエンスに影響しないことに関するオンコール アラート、反復的な手作業、ミスをすることへの恐れといった要素が、チームの人員の燃え尽き症候群やウェルビーイングの低下につながります。

信頼性の手法を活用しているチームでは反対の結果が得られました。信頼性の手法を利用していないチームに比べて、生産性と仕事の満足度が高く、燃え尽き症候群のレベルが低いことが報告されています。このようなウェルビーイングの向上は、公開されている SRE の手法によってもたらされていると考えられます。

- トイルの削減³
- 誰も責めない事後分析⁴
- チームの自律性⁵
- チームの劣線形スケーリング⁶

³ Beyer, Betsy, et al. Site Reliability Engineering: How Google Runs Production Systems (O'Reilly, 2016), 49–54. <https://sre.google/sre-book/eliminating-toil/>

⁴ Ibid, <https://sre.google/sre-book/postmortem-culture/>

⁵ Beyer, Betsy, et al. The Site Reliability Workbook (O'Reilly, 2018), <https://sre.google/workbook/team-lifecycles/>

⁶ Brookbank, James, and McGhee, Steve. Enterprise Roadmap to SRE (O'Reilly, 2022), 11. <https://sre.google/resources/practices-and-processes/enterprise-roadmap-to-sre/>

運用パフォーマンス

回答者に自社サービスの運用パフォーマンスについても尋ねました。まず、サービスの信頼性に対する不満をユーザーから直接聞かされる頻度を尋ねました。次に、サービスが利用できない状態や遅い状態、あるいは正しく動作していない状態になる頻度を尋ねました。

➤ 信頼性の手法は運用パフォーマンスを通じて チームと組織のパフォーマンスを高める

信頼性の手法を採用することでチームの運用パフォーマンスが向上します。組織が本番環境のフリートを効果的に運用できれば、他の成果も増幅することがわかりました。成果が高い場合、信頼性の手法でさらに高まります。成果が低い場合、信頼性の手法は役に立たず、成果は低いままになります。

それでも信頼性の高いシステムは、顧客に合ったソフトウェア機能を効果的に提供する必要があります。SRE は単独での運用をまったく意図していないため、これは理にかなっていません。信頼性目標を達成することは、SRE チームにとって成功の重要な指標です。これは運用パフォーマンスに反映されます。信頼性の手法を利用することの利点は他にもありそうですが、データは運用パフォーマンスへの影響が最も重要であることを示しています。さらに、運用パフォーマンスの向上にはサービスの健全性以外の利点もあります。実際、信頼性の手法を利用すると実務者のウェルビーイングが高まることが確認されています。

➤ 運用パフォーマンスがウェルビーイングに影響 する

信頼性の高いサービスは、オンコール対応や時間外の緊急メンテナンスなどでサービス運用者のウェルビーイングに悪影響を与える、というのが業界の共通認識です。しかし、実際は運用パフォーマンスが高ければ燃え尽き症候群が減り、生産性が上がり、仕事の満足度が高まることがわかりました。これは、トイルの削減の SRE 原則に合致します⁷。手作

業部分を自動化すると個人の満足度が上がり、チームにとっても継続的な負担が減ることになります。

➤ 運用パフォーマンスがチームパフォーマンスと組織パフォーマンスを増幅する

運用パフォーマンスは、チームパフォーマンスと組織パフォーマンスの両方に大きな好影響を与えることがわかりました。これは DevOps ムーブメントの支持者にとっては当然のことです。マシンを効果的に運用できれば、チームが多くのことを達成できるようになり、組織は成功します。

➤ 運用パフォーマンスがソフトウェアデリバリーパフォーマンスを高める

ソフトウェアデリバリーパフォーマンスはチームパフォーマンスと組織パフォーマンスの両方を向上させることができますが、どちらも運用パフォーマンスで大幅に強化されます。さらに、ソフトウェアデリバリーパフォーマンスの高いチームは、同時に高い運用パフォーマンスも達成しなければ、高いチームパフォーマンスと組織パフォーマンスを達成することはできません。どちらも必要です。実際、相応の運用パフォーマンスを達成せずにソフトウェアデリバリーパフォーマンスを向上させたチームは、結果的に組織の成果が悪化しました。つまり、魅力的なソフトウェアを素早く作成できても、ユーザーの期待に応える運用が本番環境でできなければ、市場からの見返りが得られないということです。

取り上げられなかったものと今後について

このような相互作用を理解するために役立つ尺度は他にもあるはずです。たとえば今年、コスト管理が能力と成果にどのように関係するのか、ということをよく尋ねられました。コスト意識の高い組織もあれば、そうでない組織もあります。これは組織の計画や意思決定のあり方に影響します。同様に、信頼性の手法については、明確に求められたり計画されたりしなかった場合であっても、非常に協力的な組織文化があれば自然に生まれるのではないかと DORA は推測しています⁸。今後も、チームが既存の IT 運用手法をどのように進化させるのかについて、また、そのような進化がシステムの信頼性、チームパフォーマンス、ウェルビーイングにどのような影響を及ぼすのかについて、理解を深めていきたいと考えています。

DORA は何より、皆様の声を聞きたいと考えています。DORA の実務者のコミュニティ [DORA.community](https://dora.community) にぜひご参加ください⁹。SRE はまだ新しい分野です。その影響は、信頼性の手法を採用した組織ごとに、またこれまでずっと SRE を行ってきたことに気づいた組織でも、それぞれ異なります。こうした変化は緩やかなものです。DORA は、経時的な進歩を示す一貫した測定を行おうと考えています。コミュニティとして、何が役立つのかを共有し、その過程で互いに高め合うことができます。

⁷ Beyer, Betsy, et al. Site Reliability Engineering: How Google Runs Production Systems (O'Reilly, 2016), 49–54. <https://sre.google/sre-book/eliminating-toil/>

⁸ Brookbank, James, and McGhee, Steve. Enterprise Roadmap to SRE (O'Reilly, 2022), 5. <https://sre.google/resources/practices-and-processes/enterprise-roadmap-to-sre/>

⁹ DORA コミュニティ、<https://dora.community/>

Google の SRE の方法

Google 内での SRE は、20 年にわたり、資金の十分な成長組織の中で進化してきました。SRE は、大きな費用をかけずに Google 検索と Google 広告を急成長させる必要性から生まれました。初めのころ、これらのプロダクトではそのリアルタイムな性質から高い信頼性が求められました。検索や広告では一過性のエラーで即座にお客様を失うことになり、やり直しのチャンスはありません。ダイナミック広告はミリ秒単位で計算しなければならず、検索が遅いと Google ブランド全体が評判を落とします。

同時に、SRE は新しいタイプの企業、つまり購入より構築を選ぶボトムアップのエンジニアリング主導型企業の内部で大切に開発されていきました。Google は、分散システムやコンパイラ設計のような学術的コンピュータサイエンスの分野に強い、サイト信頼性エンジニア (SRE エンジニア) のスタッフを活用しました。この組織文化の DNA により、SRE が生まれて成功する豊かな環境が整いました。SRE チームは、本番環境への鍵を託されただけでなく、システムをスケールアップする斬新な方法を見つける任務も与えられました。

しかし、SRE はどのようにして徐々にスケールアップしたのでしょうか。一言で言うなら「劣線形」です。つまり、Google の規模が 2 倍になるたびに雇用する SRE エンジニアの数を 2 倍にすることはできませんでした。初期の Google プロダクトが世界的な需要に合わせてスケールアップしていた (そして Gmail、Google マップ、Android、YouTube、Google Cloud といった新しいプロダクトを導入していた) ペースを考えると、新しくまだ珍しかった SRE エンジニアを顧客ベースの成長と同じペースで増やすことは不可能でした。

そこで、このような言葉が生まれました。

- SRE は、ユーザー数に比例してスケールアップしてはならない。
- SRE は、サーバー数に比例してスケールアップしてはならない。
- SRE は、クラスタ数に比例してスケールアップしてはならない。
- SRE は、サービス数に比例してスケールアップしてはならない。

この制約された成長モデルを維持できるような管理体制が築かれました。SRE エンジニアたちは、連携するプロダクト開発チームと密接な関係を維持しつつ、互いに足並みを揃えて協力しました。SRE エンジニアは、Ben Treynor-Sloss¹⁰をトップとする独自の管理系統に属していました。これらのチームは、プロダクト開発 (「開発」) のプロダクト分野 (PA) と完全に対応した PA に分かれました。SRE エンジニアチームは対応する開発チームと連携し、SRE エンジニアの最適な活用方法を自由に決定しました。開発チームは新しい SRE エンジニアに資金を直接投じることができました。すべての開発チーム

¹⁰ エンジニアリング担当バイス プレジデント Benjamin Treynor Sloss. Site Reliability Engineering: How Google Runs Production Systems (O'Reilly, 2016). <https://sre.google/sre-book/part-1-introduction/>、<https://sre.google/sre-book/introduction/#id-2opuzSjFr>

がユーザー向けのプロダクトを作成していたわけではありません。チームの多くは、Bigtable¹¹（構造化データストレージ）、Borg¹²（コンピューティングスケジュール管理）、Colossus¹³（分散ストレージ）などの、共有インフラストラクチャチームでした。そしてこのような共有インフラストラクチャサービスにより、お客様対応チームは専属のSREチームなしでスケールアップできるようになりました。

このような各種チームを組織内に抱えることで、それぞれのチームは一貫した雇用と昇進プロセスを維持できました。SREチームは開発チームに比べて10分の1以下の規模となる傾向があるため、SREが自律性を持ち、SRE原則に反する方向に引っ張られないようにすることが重要でした。

SREエンジニアはチーム内にプロダクトマネージャーを配置して独自のプロダクトを開発しました。SREプロダクトのユーザーは、適切な方法で本番稼働を行いたい他のチームでした。SREチームは、インシデント対応、モニタリングとオプザバビリティ、リリース管理、キャパシティプランニング、トラブルシューティングに関するプロダクトを開発しました。

SREチームが成長する過程では、常にチームの健全性とサステナビリティが考慮されていました。たとえば、新しいオンコールSREチームの最小規模は12人（2か所に6人ずつ配置）でした。これによりタイムゾーンをまたいだ対応が実現しました。また、チーム内に十分な数の人員を配置することで、燃え尽き症候群になる人を出さず、良好なワークライフバランスを保つことができました。

SREチームは今も状況への適応を続けています。すべてのチームが前述のモデルどおりというわけではなく、別の道を選んだチームもあります。Google内のSREチームのように大規模な組織でも、他の大規模なグループと同様に市況の影響を受けるため、柔軟性が重要です。しかし何よりもSREチームは、リスクの受容、サービスレベルの測定、トイルの排除、自動化の導入、シンプルさの追求といった原則を守っています。



¹¹ Chang, Fay, et al. Bigtable: A Distributed Storage System for Structured Data, 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI), {USENIX} (2006), pp. 205-218, <https://research.google/pubs/pub27898/>

¹² Verma, Abhishek, et al. Large-scale cluster management at Google with Borg, Proceedings of the European Conference on Computer Systems (EuroSys), ACM, Bordeaux, France (2015), <https://research.google/pubs/pub43438/>

¹³ Hildebrand, Dean, et al. Colossus の仕組み: Google のスケーラブルなストレージシステムの舞台裏 2021年4月19日 - <https://cloud.google.com/blog/ja/products/storage-data-transfer/a-peek-behind-colossus-googles-file-system>

6章

柔軟なインフラストラクチャが成功の鍵

要点


柔軟なインフラストラクチャは、チームパフォーマンス、組織パフォーマンス、運用パフォーマンス、ソフトウェアデリバリーパフォーマンスの兆候となる要素です。クラウドコンピューティングは柔軟なインフラストラクチャを実現する核となりますが、自動的に恩恵を受けられるわけではありません。クラウドを使用する方法が重要だということをデータが示しています。

はじめに

DORA の調査の多くにおいて、米国国立標準技術研究所 (NIST) が定義するクラウドコンピューティングの基本特性に注目し、インフラストラクチャについて実務者に尋ねました¹。

- オンデマンドセルフサービス
- 広範なネットワークアクセス
- リソースプール
- 迅速な弾力性
- 測定可能なサービス

DORA は、組織パフォーマンスとソフトウェアデリバリーパフォーマンスの向上がこの5つの特性から予測されることを一貫して確認してきました。今年も、クラウドコンピューティングを使用することが、より柔軟なインフラストラクチャにつながるのかどうかを確認することにしました。



柔軟なインフラストラクチャがある場合、そうでない場合と比べて

30%

組織パフォーマンスが高くなる
ことが予測される

¹NIST 特別刊行物 800-145: "The NIST Definition of Cloud Computing."

	割合
マルチクラウド	19.6%
パブリッククラウド	51.5%
ハイブリッドクラウド	33.6%
オンプレミス	19.8%
小規模なオンプレミス	3.2%
その他	2.5%

* 複数選択式で実施。

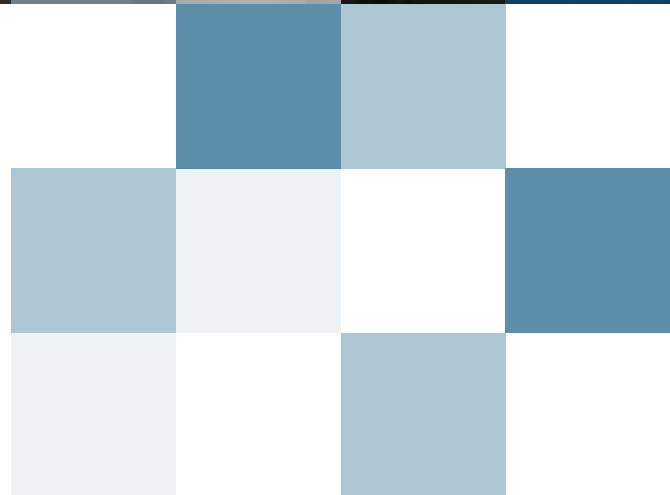
結果

再び以前と同じ調査結果が確認されました。すなわち、チームがクラウドを使用する方法の方が、単にクラウドを使用しているということよりも、パフォーマンスを予測するための要素として重要です。クラウドを使用することは強力な成功要因になり得ますが、自動的に恩恵を受けられるわけではありません。実際、パブリッククラウドではチームが柔軟なインフラストラクチャを活用しない限り、ソフトウェアパフォーマンスと運用パフォーマンスが低下するという示す指標が確認されています。この結果は、単なる「リフト&シフト」(ワークロードをデータセンターからクラウドへ移行する行為)は有益ではなく有害になり得る、という考えをさらに裏付けます。

クラウドコンピューティングの利用は、燃え尽き症候群の大幅な減少や、仕事の満足度と生産性の大幅な向上と関連しています。

コンピューティング環境

上の表は、主なアプリケーションまたはサービスを運用していると回答者が答えた場所を示しています。



クラウドのタイプ	組織パフォーマンス	チームパフォーマンス	ソフトウェアデリバリーパフォーマンス	運用パフォーマンス
プライベート	⊖ 影響の兆候なし	⬆️ クラウドコンピューティングの利用に伴い大幅に向上	⊖ 影響の兆候なし	⬆️ クラウドコンピューティングの利用に伴い大幅に向上
パブリック	⬆️ クラウドコンピューティングの利用に伴い極めて大幅に向上		⬆️ クラウドコンピューティングの利用に伴い大幅に低下	⬆️ クラウドコンピューティングの利用に伴い大幅に低下
ハイブリッド			⬆️ クラウドコンピューティングの利用に伴い大幅に低下	⊖ 影響の兆候なし
マルチ			⊖ 影響の兆候なし	⬆️ クラウドコンピューティングの利用に伴い大幅に低下

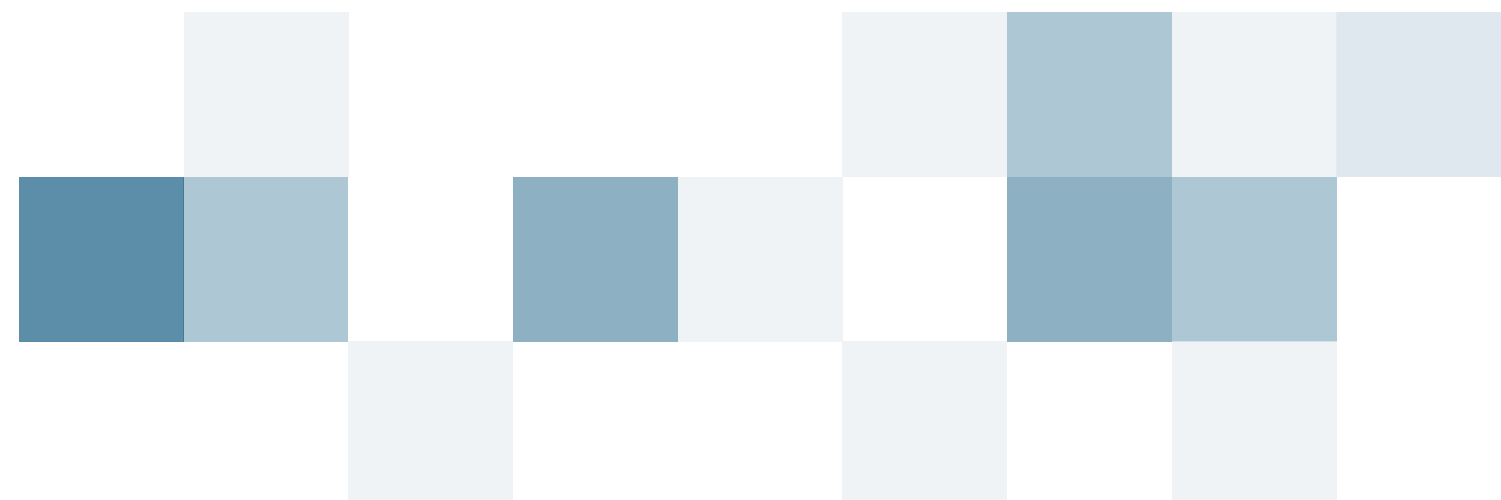
単なる「クラウド利用」では結果はまちまち

結果の表に示すように、単なる「クラウド利用」の場合、ソフトウェアデリバリーパフォーマンスと運用パフォーマンスに対する影響はないか、悪い方向となっています。影響がないか悪影響を与えるということは、クラウド移行の第一歩を踏み出した実務者が、新しい環境での作業、新しいツールでの作業、これまでとは異なるやり方に直面した結果だと考えられます。多くの場合、企業は自社のデータセンターのときと同じようにクラウドを使用しており、新しい環境の複雑さと認知負荷が増えただけになっています。新しい環境への

適応に失敗すると、ソフトウェアデリバリーパフォーマンスや運用パフォーマンスが向上しないどころか、むしろ害が及びます。

ただし、プライベートクラウドでの運用パフォーマンスは例外的な結果となっています。

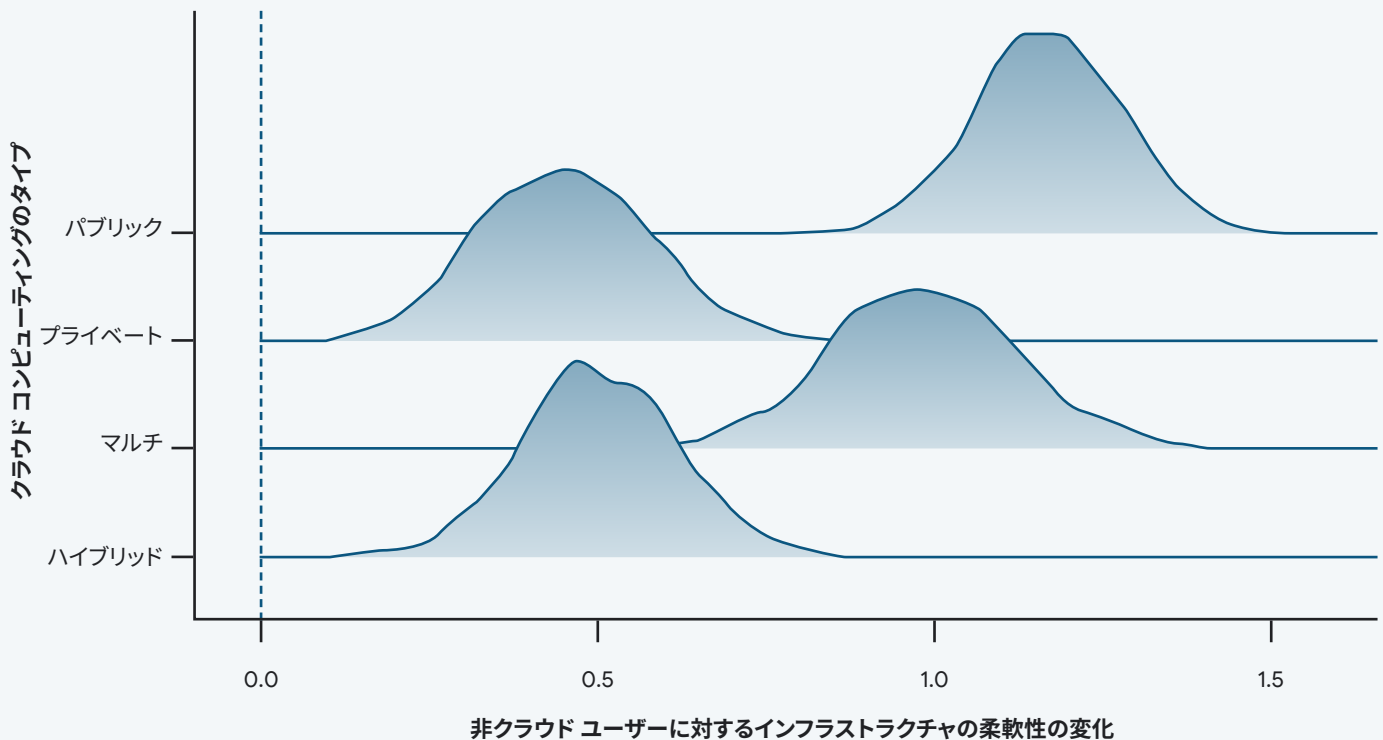
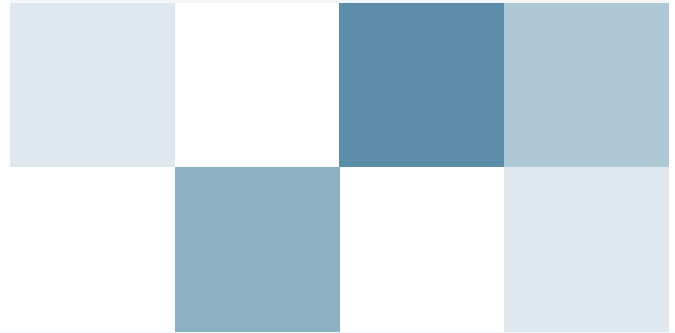
ソフトウェアデリバリーパフォーマンスと運用パフォーマンスを向上させるのは、次に説明する**柔軟なインフラストラクチャ**です。



クラウド インフラストラクチャは柔軟さを生む

パブリック クラウドを使用した場合、クラウドを使用しなかった場合と比べてインフラストラクチャの柔軟性が22%高くなります。複数のクラウドを使用した場合も、単一のパブリック クラウドよりは低いですが、柔軟性は向上します。当然持ち上がる問いは、それはなぜかということです。データによると、クラウド コンピューティングによって実現することの多い柔軟なインフラストラクチャには、クラウド プラットフォームを単に使用することよりも大きな影響力があります。多くの人にとってクラウドは新しい方法であり、習熟するまでに時間がかかります。クラウド プラットフォームはそれぞれ異なるため、プラットフォームが増えると各プラットフォームをうまく運用するために必要な認知負荷が増えます。

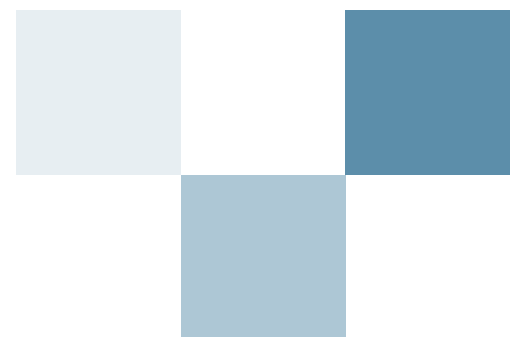
パブリック クラウドを使用した場合、クラウドを使用しなかった場合と比べてインフラストラクチャの柔軟性が高くなります。



柔軟なインフラストラクチャは主な成果のパフォーマンスの向上につながる

能力	組織パフォーマンス	チームパフォーマンス	ソフトウェアデリバリーパフォーマンス	運用パフォーマンス
柔軟なインフラストラクチャ	↑↑ より柔軟なインフラストラクチャに伴い大幅に向上	↑↑ より柔軟なインフラストラクチャに伴い大幅に向上	↑↑ より柔軟なインフラストラクチャに伴い大幅に向上	↑ より柔軟なインフラストラクチャに伴い向上

柔軟なインフラストラクチャは、組織パフォーマンス、チームパフォーマンス、ソフトウェアデリバリーパフォーマンス、運用パフォーマンスを向上させる、ということをご認識しておく必要があります。多くの組織がインフラストラクチャをクラウドにリフト&シフトすることを選択します。これは素晴らしい第一歩になり得ますが、ほんの始まりに過ぎません。ワークロードの一部をリフト&シフトすることを決めたら、次のステップは、柔軟なインフラストラクチャを活用するためのリファクタリングによって、ワークロードをモダナイズすることです。



クラウドコンピューティングは柔軟なインフラストラクチャを通じて主な成果に好影響を与える

柔軟なインフラストラクチャを備えたクラウドのタイプ	組織パフォーマンス	チームパフォーマンス	ソフトウェアデリバリーパフォーマンス	運用パフォーマンス
プライベート	↑ 柔軟なインフラストラクチャによる部分的なメディエーション	↑ 柔軟なインフラストラクチャによる部分的なメディエーション	↑ 柔軟なインフラストラクチャによる部分的なメディエーション	↑ 柔軟なインフラストラクチャによる部分的なメディエーション
パブリック	↑↑ 柔軟なインフラストラクチャによる完全なメディエーション	↑↑ 柔軟なインフラストラクチャによる完全なメディエーション	↑ 柔軟なインフラストラクチャによる部分的なメディエーション	↑ 柔軟なインフラストラクチャによる部分的なメディエーション
ハイブリッド	↑ 柔軟なインフラストラクチャによる部分的なメディエーション	↑↑ 柔軟なインフラストラクチャによる完全なメディエーション	↑ 柔軟なインフラストラクチャによる部分的なメディエーション	↑↑ 柔軟なインフラストラクチャによる完全なメディエーション
マルチ	↑ 柔軟なインフラストラクチャによる部分的なメディエーション	↑ 柔軟なインフラストラクチャによる部分的なメディエーション	↑ 柔軟なインフラストラクチャによる部分的なメディエーション	↑ 柔軟なインフラストラクチャによる部分的なメディエーション



クラウド コンピューティング プラットフォームは、柔軟なインフラストラクチャの特性を最大限に活用するような方法で使用した場合、ソフトウェア デリバリー パフォーマンスと運用パフォーマンスに好影響を与えることが予測されます。この影響の違いからわかるのは、ほとんどの実務者やリーダーがすでに知っていること、つまり、単にデータセンターからクラウドにワークロードを移行するだけでは成功しない、ということです。クラウドで実現する柔軟なインフラストラクチャを活用することが重要です。

インフラストラクチャのタイプ	成果
クラウドと柔軟なインフラストラクチャの組み合わせ	
柔軟性のないクラウド	

可能性を最大限に引き出すには、アプリケーションのビルド、テスト、デプロイ、モニタリングの方法を見直す必要があります。この見直しの大部分は、クラウド コンピューティングの5つの特性(オンデマンドセルフサービス、広範なネットワークアクセス、リソースプール、迅速な弾力性、測定可能なサービス)の活用を中心に展開します。



クラウド コンピューティングでウェルビーイングが向上する

クラウドのタイプ	燃え尽き症候群*	仕事の満足度	生産性
プライベート	⊖ 影響の兆候なし		
パブリック	⬇️ クラウド コンピューティングの利用に伴い極めて大幅に減少	⬆️ クラウド コンピューティングの利用に伴い大幅に向上	⬆️ クラウド コンピューティングの利用に伴い大幅に向上
ハイブリッド	⊖ 影響の兆候なし		
マルチ	⊖ 影響の兆候なし		

* 燃え尽き症候群はカラーパターンが反対になっています。燃え尽き症候群は減少する方が良いからです。

データは、クラウド コンピューティングが従業員のウェルビーイングに大きく貢献していることを示しています。仕事の満足度と生産性の両方が大幅に向上し、燃え尽き症候群には影響がないか好影響が見られます。言い換えると、クラウド コンピューティングでは認知負荷が増し、新しいツールや業務の進め方への習熟が必要になるものの、ウェルビーイングへの悪影響はないということです。

実務者として、この理由について仮説を立てると次のようになります。エンジニアは学習や問題解決を好み、柔軟なコンピューティング特性を持つ環境で働くことを楽しみます。新しい技術を学ぶことは楽しいだけでなく、キャリア形成の面でも価値があります。エンジニアは所属組織が成功している方が幸せです。



7章

組織文化への投資なしには何事もうまくいかない

要点

組織文化は、従業員のウェルビーイングと組織パフォーマンスの重要な原動力です。組織文化が健全であれば、燃え尽き症候群を減らし、生産性を高め、仕事への満足度を高めることができます。また、組織パフォーマンス、ソフトウェア デリバリー パフォーマンス、運用パフォーマンス、チーム パフォーマンスが有意に向上します。成果の向上に関連する技術的能力を活かすうえでも、健全な組織文化がチームの成功を後押しします。



はじめに

組織文化を定義するのは難しいことです。DORA は、職場の人員がどのような体験をしているのかを示す指標に注目しています。そのため、パフォーマンスを予測するうえで重要な指標であり続けてきた Westrum の組織文化の類型¹を使用しています。今年は、Westrum が定義する「創造的文化」を持つ企業が引き続き高いパフォーマンスを示していることが確認されました。

チームや組織の文化に寄与すると思われる要素を下表に示します。

要素	定義
Westrum の組織文化	組織が問題や機会にどのように対応する傾向があるか。組織文化には、創造的、官僚的、不健全の3種類がある。
組織の安定性	従業員にとって職場環境がどの程度安定しているか、または不安定か。
雇用保障	従業員が雇用保障を心配する頻度。
柔軟性	いつ、どこで、どのように働くか。
知識の共有	アイデアや情報が組織全体に広がる仕組み。質問に答えたチームメンバーは他のメンバーにも情報を共有するため、答えを待つ必要がない。
ユーザー中心	ソフトウェアを開発する際にエンドユーザーを重視し、ユーザーのニーズや目標について理解を深める。プロダクトやサービスの改善にユーザー シグナルが使用される。
仕事の配分	負荷の大きいタスクをチームメンバーで公平に分担できるようにする正式なプロセス。

組織文化の要素、プロセスに関する能力、技術的能力の境界線は、必ずしも明確ではありません。DORA は、組織文化は実践から生まれ、実践は組織文化から生まれるものだと考えています。これについては後ほど調査結果について述べる際に触れます。

¹<http://bmi.co/1BRGh5q>

判明した内容とその意味

健全な組織文化は主な成果を向上させる

全体として、健全な組織文化は主な成果すべてに好影響を与えます。今回、創造的文化が組織パフォーマンス、ソフトウェア デリバリー パフォーマンス、運用パフォーマンスを向上させるというこれまでの調査結果が再現しました。また、創造的文化は今年の新しいパフォーマンス指標であるチーム パフォーマンスも向上させます。

ソフトウェア開発のユーザー中心アプローチでパフォーマンスが有意に改善することがわかりました。これは注目に値します。ユーザーを第一に考えることで、組織はさまざまな恩恵を受けられます。ユーザー フィードバックは、チームがプロジェクトを優先順位付けするためにも、ユーザーのニーズを満たすプロダクトやサービスを生み出すためにも役立ちます。これは、ユーザー エクスペリエンスの向上、ユーザー満足度の向上、収益の増加につながります。

チーム間での仕事の配分を測定することで、組織文化の健全性も評価しました。仕事を公平に配分すると、チームパフォーマンスと組織パフォーマンスに好影響を及ぼしますが、同時に、ソフトウェア デリバリー パフォーマンスが低下することがわかりました。仕事の配分に関する正式なプロセスが、ソフトウェア デリバリー パイプラインの一部である負荷の大きいタスクの遂行を遅らせているのだと思われる。また、あるタスクをチーム内の誰が引き受けるべきかに、正式なプロセスが影響を与えている可能性があります。

もう一つの矛盾しているような調査結果として、組織の安定性に関し、少しではあるものの無視できないソフトウェア デリバリー パフォーマンスの低下が見られました。この説明としては、地位が確立された（おそらく大規模な）組

創造的文化のあるチームは
そうでないチームより

30%

組織パフォーマンスが高い

織は、まだ地位が確立されていない新しい（小規模な）組織ほどには、速く動かなければならないというプレッシャーを感じていないのだと考えられます。確立された組織は、定評のあるプロダクトがすでにある場合が多く、そうであればソフトウェア デリバリーのスピードに関して柔軟に対応できます。

情報の流れがスムーズであれば物事はうまくいきます。ソフトウェア デリバリー パフォーマンスと運用パフォーマンスの向上には情報共有レベルの高さが関係していることがわかりました。情報にすぐアクセスできる場合や知識のサイロ化がほとんどない場合、人員はタスクの実施に必要な情報を探すことにはなく、重要なタスクそのものに時間をかけることができます。

最後に、いつどこでどのように働くかを従業員が決定できる柔軟な勤務形態は、すべてのパフォーマンス指標に好影響を与えます。ソフトウェア デリバリー パフォーマンスでは特にそうです。組織がリモートワークのポリシーを強化するにしても、従業員がある程度の柔軟性を維持できるようにすることは有益でしょう。

組織文化の要素	チーム パフォーマンス への影響	組織パフォーマンス への影響	ソフトウェア デリバリー パフォーマンスへの影響	運用パフォーマンス への影響
Westrum の組織文化	↑↑ 大幅に向上	↑↑ 大幅に向上	↑↑ 大幅に向上	↑↑ 大幅に向上
組織の安定性	↑ わずかに向上	↑↑ 大幅に向上	↓ わずかに低下	⊖ 影響なし
雇用保障	↑ わずかに向上	⊖ 影響なし	↑ わずかに向上	↑ わずかに向上
柔軟性	↑ わずかに向上	↑ わずかに向上	↑↑ 大幅に向上	↑ わずかに向上
知識の共有	↑ わずかに向上	↓ わずかに低下	↑↑ 大幅に向上	↑↑ 大幅に向上
ユーザー中心	↑↑ 大幅に向上	↑↑ 大幅に向上	↑ わずかに向上	↑↑ 大幅に向上
仕事の配分	↑↑ 大幅に向上	↑↑ 大幅に向上	↓↓ 大幅に低下	⊖ 影響なし



健全な組織文化で技術的能力が向上する

調査結果によると、優れた組織文化は技術的能力の実際の活用を促進します。DORA は、組織文化と技術的能力の間には「組織文化は実践から生まれ、実践は組織文化から生まれる」という相互関係があると考えています。

組織文化は広範であり、定義するのは難しいですが、技術的能力は通常は範囲が設定され明確に定義されています。これは、組織内の個人がどのように変革に貢献できるかという点に影響を与えます。

たとえばリーダーは、創造的文化を促進するようなインセンティブ構造を作ることができます。リーダーも個々の関係者も、ソフトウェア開発のユーザー中心アプローチに注力できます。個々の関係者は、トランクベース開発、継続的インテグレーション、信頼性の手法、疎結合アーキテクチャ

など、パフォーマンスを向上させる技術的能力の適用を促進できます。こうした技術的能力を実際に活用することは簡単ではありません。成功させるには、皆が協力し、先入観を持たず、互いに支え合い、学び合う必要があります。これらすべてが健全な組織文化の構成要素です。このようなチームは組織内の他チームの模範になり得ます。利用できるあらゆるツールを活用して変革を進めていく意欲を起こさせることができます。

組織文化の長期にわたる有意義な変化は、トップダウンとボトムアップの取り組みを同時に行って変化を起こすことで実現します。

組織文化の要素	トランクベース開発への影響	信頼性の手法への影響	継続的インテグレーションへの影響	継続的デリバリーへの影響	疎結合アーキテクチャへの影響
Westrum の組織文化	↑↑ 大幅に向上	↑↑ 大幅に向上	↑↑ 大幅に向上	↑↑ 大幅に向上	↑↑ 大幅に向上
組織の安定性	↑ わずかに向上	↑↑ 大幅に向上	⊖ 影響なし	⊖ 影響なし	⊖ 影響なし
雇用保障	↓ わずかに低下	↓ わずかに低下	⊖ 影響なし	⊖ 影響なし	⊖ 影響なし
柔軟性	⊖ 影響なし	↓ わずかに低下	↑↑ 大幅に向上	↑ わずかに向上	↑↑ 大幅に向上
知識の共有	⊖ 影響なし	⊖ 影響なし	⊖ 影響なし	↑ わずかに向上	↑ わずかに向上
ユーザー中心	↑↑ 大幅に向上	↑↑ 大幅に向上	↑↑ 大幅に向上	↑↑ 大幅に向上	↑↑ 大幅に向上
仕事の配分	↑↑ 大幅に向上	↑↑ 大幅に向上	↑↑ 大幅に向上	↑↑ 大幅に向上	↑↑ 大幅に向上

健全な組織文化で従業員のウェルビーイングが向上する
 組織文化が健全であれば、燃え尽き症候群が減り、仕事の満足度が高まり、生産性が上がることで、従業員のウェルビーイングが向上します。従業員のウェルビーイングは、あれば嬉しいプラスアルファ要素ではなく、組織全体の健全性と成功の基礎となるものです。

ニカルになり、生産性が下がります。心と体の健康にも悪影響が及びます^{2,3}。燃え尽き症候群は持続的なものであるため、少し休めば治るわけではありません。燃え尽き症候群では離職率も高くなります。従業員が健全な職場環境を求めて退職するのです⁴。そのため燃え尽き症候群を軽減するには、原因に対処する組織変革が必要です。

組織文化に投資しない組織はどうなるでしょうか。燃え尽き症候群が増え、仕事の満足度が低くなります。従業員はシ

組織文化の要素	燃え尽き症候群への影響*	仕事の満足度への影響	生産性への影響
Westrum の組織文化	 大幅に減少	 大幅に向上	 大幅に向上
組織の安定性	 大幅に減少	 大幅に向上	 わずかに向上
雇用保障	 大幅に減少	 わずかに向上	 わずかに向上
柔軟性	 わずかに減少	 わずかに向上	 わずかに向上
知識の共有	 大幅に減少	 わずかに向上	 わずかに向上
ユーザー中心	 わずかに減少	 大幅に向上	 大幅に向上
仕事の配分	 影響なし	 わずかに向上	 わずかに向上

* 燃え尽き症候群はカラーパターンが反対になっています。燃え尽き症候群は減少する方が良いからです。

² Adam Bayes, Gabriela Tavella & Gordon Parker (2021) "The biology of burnout: Causes and consequences," The World Journal of Biological Psychiatry, 22:9, 686–698. DOI: 10.1080/15622975.2021.1907713. <https://doi.org/10.1080/15622975.2021.1907713>

³ Maslach C, Leiter MP. "Understanding the burnout experience: recent research and its implications for psychiatry." World Psychiatry, June 2016, 15(2), 103–11. DOI: 10.1002/wps.20311. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4911781/>. PMID: 27265691; PMCID: PMC4911781.

⁴ L.A. Kelly, et al. "Impact of nurse burnout on organizational and position turnover." Nursing Outlook, January 2021, 96–102, January–February 2021, 96–102. DOI: doi.org/10.1016/j.outlook.2020.06.008

個人の属性が重要になる状況と理由

要点

個人の属性は重要です。回答者の特定のグループで、燃え尽き症候群の増加や生産性の低下など、他の回答者とは異なる結果が出るようになりました。このような悪影響を軽減するために実施できる具体的な手法も明らかになりました。

はじめに

2022 年の分析では、「特定の業務の進め方が一部の人のウェルビーイングに資する可能性はあるが、全員ではない」という一般的な現象が広く見られました。

2022 年の調査では、過小評価グループを自認する人ほど燃え尽き症候群を多く報告していることがわかりました¹。本章では、この調査結果の再現性を確認し、過小評価グループの方が燃え尽き症候群を経験しやすい理由と、それを防ぐために役立つ要素を探ります。

また、多くの業界を襲っている不安定性は、新規採用者に関する問題につながっています。組織は、新しい従業員が生産的になるまでに時間がかかることを懸念しています。新しい従業員を短期間で一人前にさせられるような方法を探しているのです。この点についても掘り下げていきます。

判明した内容とその意味

燃え尽き症候群になりやすい人がいる

去年の調査では、女性を自認する回答者や、ジェンダーを自己決定する回答者、そしてなんらかの形で過小評価状態を自認する回答者の方が、男性を自認する回答者や過小評価状態を自認しない回答者よりも燃え尽き症候群の報告が多いことがわかりました。このような調査結果は、過小評価グループの人員はそうでない人員と比べて燃え尽き症候群²と仕事関連のストレス³が多いことを示唆する多くの先行研究と一致します。

そのため、DORA は今年も燃え尽き症候群に格差が見られるかどうかを調べることにし、結果はそのとおりでした。女性を自認する回答者またはジェンダーを自己決定する回答者が燃え尽き症候群を報告するレベルは、男性を自認する回答者よりも 6% 高くなりました。なんらかの過小評価状態を自認する回答者が燃え尽き症候群を報告するレベルは、過小評価状態を自認しない回答者よりも 24% 高くなりました。

¹ 2022 Accelerate State of DevOps Report. <https://dora.dev/research/2022/dora-report/>

² Sigalit Ronen and Ayala Malach Pines, "Gender Differences in Engineers' Burnout," Equal Opportunities International, November 7, 2008, <https://www.emerald.com/insight/content/doi/10.1108/02610150810916749/full/html>

³ Dalessandro C.; Lovell A.; Tanner OC., "Race, Marginalization, and Perceptions of Stress Among Workers Worldwide Post-2020." Sociological Inquiry, August 3, 2023, <https://onlinelibrary.wiley.com/doi/pdf/10.1111/soin.12505>

燃え尽き症候群にやりやすい業務もある

質の高いドキュメントや安定したチームといった、影響がないか好影響があると思われる職場の要素が、すべての人の燃え尽き症候群を減らすわけではないということがわかりました(4章 - ドキュメントが基礎となる参照)。これは、組織にとっては有益でも人によっては燃え尽き症候群の原因となるタスクがあるためではないかと考えられます⁴。

回答者の燃え尽き症候群の経験を知るために、回答者の業務について尋ね、次の2つに着目して測定しました。

- 具体的なタスク(コーディング、ミーティング、チームメイトのサポートなど)
- 業務の特徴(予定外の業務やその可視性、トイルの量など)

同じタスクでも経験は人や状況によって異なることがあるため、業務の特徴は重要です。たとえばコードレビューには、予定外のトイルになるものもあれば、チーム内ではっきりとしてリーダーシップや技術的な専門知識が発揮されるものもあります⁵。

過小評価グループを自認する回答者では、反復的な業務(トイル)を行っているとの報告が過小評価グループを自認しない回答者よりも24%多くなりました。女性を自認する回答者またはジェンダーを自己決定する回答者では、反復的な業務(トイル)を行っているとの報告が男性を自認する回答者よりも40%多くなりました。この2つの回答者グループでは、予定外の業務、同僚に認識されにくい業務、専門スキルに直接関係しない業務を行っているとの報告も多くなりました。これらのグループが報告した燃え尽き症候群の一部は、この調査結果で説明がつかず。

⁴ Linda Babcock, Brenda Peyser, Lise Vesterlund, and Laurie Weingart. *The No Club* (New York: Simon and Schuster, 2022), 17.

⁵ Murphy-Hill, E. et al. "Systemic Gender Inequities in Who Reviews Code," *Computer Supported Cooperative Work* (2023) (採録決定), <https://research.google/pubs/pub52204>

⁶ Babcock et al., *The No Club*, 17.

昇進につながらないタスク

Babcockらは著書『The No Club』の中で、特定の種類の業務について述べています。「昇進につながらないタスクは、組織にとっては重要だが、キャリアアップには役立たない⁶。」

ここでいう「キャリアアップ」は、報酬の増額や他の仕事での採用可能性など、広い意味で使われています。

女性の方がこの種の業務を多く行っているということが確認されています。Babcockらは、このような仕事の配分の不平等性について理由を述べています。たとえば、女性の方がこの種の業務を頼まれやすく、頼まれた場合に断ると悪影響(社会的コスト)が伴うため、断らない可能性が高くなります。

このような仕事の配分の不平等性がもたらす結果についても述べています。たとえば、一部の女性は次のような結果を被ります。

- キャリアや収入に悪影響がある
- キャリアに見合った適切な業務量にするために、業務時間が長くなる

仕事の配分の正式なプロセスが一部の回答者の 燃え尽き症候群を減らす

仕事を均等に配分するための正式なプロセスがあるかどうかを回答者に尋ねました。これを**仕事の配分**と呼び、これで一部の回答者が経験した燃え尽き症候群が減ると予想しました。

結果として、仕事の配分により、男性を自認する回答者と、女性を自認する回答者またはジェンダーを自己決定する回答者において、燃え尽き症候群が減ったことがわかりました。仕事の配分のレベルが高くなると、ジェンダーによる燃え尽き症候群の差はなくなりました。

仕事の配分が、過小評価グループを自認する回答者の経験する燃え尽き症候群のレベルに影響を与えないことがわかったのは驚きでした。この調査結果からは、さらに疑問が生じます。仕事を均等に配分するための正式なプロセスがあっても、仕事が平等に配分されないのでしょうか。仕事の平等性では、中断や可視性などのタスクの特徴が考慮されているのでしょうか。また、このグループにとってより重要な可能性がある、業務以外で燃え尽き症候群につながる他の要因を、どのように軽減すればよいのでしょうか。



帰属意識の醸成

著者: Jeffrey Winer 博士

(ハーバード大学医学大学院ボストン小児病院助教授、所属心理士)

重要な調査結果と背景

本レポートの調査結果で重要なのは、過小評価グループを自認する人の燃え尽き症候群は同僚よりはるかに多いということです。ここまで、考えられる理由を見てきました。本セクションでは、この調査結果を、帰属意識に関する幅広い研究や関連する組織の方策と結び付けて考えてみます。

グループ内で過小評価グループを自認することは、広く知られた心理学的現象である「帰属の曖昧性」⁷に対する脆弱性を示しています (Geoffrey Cohen 博士による「Understanding and Overcoming Belonging Uncertainty」⁸参照)。この曖昧性(「私はここに属しているのか?」、「私のような者はここで成功できるのか?」など)は、継続的な経験とその経験の解釈を通じて、強化されるか再定義されます。このような帰属の曖昧性に関する広く知られたプロセスは、「燃え尽き症候群を報告するレベルは過小評価グループを自認する人の方がはるかに高い」という本レポートの調査結果を説明する助けとなるかもしれません。

組織にできること

忘れてはならないのは、多様性、インクルージョン、公正性、帰属性はそれぞれ異なり、達成するためには相互に関連した種々の持続的な戦略が必要になる、ということです。帰属意識の確立には、本当の持続的な取り組みが必要です。

組織内で苦闘している人がいた場合、まず問うべきことは「この人の何がいけないのか」ではありません。まず問うべきことは、「このように感じているのはなぜなのか、そして組織のどのような構造的要素がこのように感じさせているのか(どの要素がこのような感情を維持または悪化させているのか、など)」です。

問題が特定されたら、組織レベルでの変革だけでなく、個人レベルでの支援も行う必要があります(網羅的アプローチ)。組織の運営システムに影響を与えられるように個人を支援すると、変革がシステムに組み込まれ、その個人の代を越えて長続きするようになります。このようなシステムとサステナビリティの考え方を取り入れると、変革を制度そのものに組み込むことができ、やはりその個人の代を越えて長続きするようになります。この生成的な性質があつてこそ、組織は帰属性を目指して取り組むことができます。「懸命に取り組む」ということが重要です。帰属意識は持続的な経験と行動によって築かれるものであり、終わりはありません。だからこそ、職場の健全性と生産性の基礎をなしているのです。

この取り組みに関して組織を支援するツールは数多く存在します。たとえば 2023 年の米国公衆衛生局長官の孤独に関するレポートでは、燃え尽き症候群に対する重要な対抗手段は社会的なつながりと帰属意識であるとされています⁹。

⁷ Walton GM;Cohen GL., "A Brief Social-Belonging Intervention Improves Academic and Health Outcomes of Minority Students," Science (New York, N.Y.), 2023 年 9 月 20 日アクセス, <https://pubmed.ncbi.nlm.nih.gov/21415354/>

⁸ <https://behavioralscientist.org/understanding-and-overcoming-belonging-uncertainty/>

⁹ Office of the Assistant Secretary for Health (OASH), "New Surgeon General Advisory Raises Alarm about the Devastating Impact of the Epidemic of Loneliness and Isolation in the United States," HHS.gov, May 3, 2023, <https://www.hhs.gov/about/news/2023/05/03/new-surgeon-general-advisory-raises-alarm-about-devastating-impact-epidemic-loneliness-isolation-united-states.html>

新規採用者は生産性の面で苦勞する

新規採用者(チームでの経験年数1年未満)の生産性のスコアは、経験豊富なチームメイト(経験年数1年以上)よりも8%低くなっています。これは予想どおりかもしれませんが、新しいチームでスタートするのは難しいことであり、たとえその役割の経験があっても、仕事を軌道に乗せるために必要となるチーム特有の知識は気が遠くなるほどの量になります。さらに、チームへの所属は、単にスキルや知識だけの問題ではありません。事例を見ると、生産性を高めるには社会的な要素も欠かせないようです。帰属意識や、貢献するメンバーとしての自覚、心理的安全性などを築くには時間がかかります。

新規採用者の成長を助けるには

DORAは、組織が新規採用者を支援できる方法は次の3つであるという仮説を立てました。

- 質の高いドキュメントを用意する
- AIをワークフローに組み込む(経験の浅い従業員の場合は経験豊富な従業員の場合よりもこれが役に立つということが他の研究で示されています)
- 対面での共同作業を行う(新人研修の段階では特に有益であることが指摘されています)



今年の調査結果は、質の高いドキュメントが生産性に大きな影響を与えること(4章 - [ドキュメントが基礎となる参照](#))と、AIには個々の生産性に一定の効果があること(3章 - [技術的能力からパフォーマンスを予測する参照](#))を示唆しています。これらの影響が新規採用者に及ばないと考える理由はありません。データからは、これらの手法は新規採用者の助けになるものの、程度はそれ以外の人と変わらないということがわかりました。つまり、新規採用者が特別な恩恵を受けるわけではありません。

新規採用者も他の従業員も支援する場合は、生産性に対する影響が大きく明確であることから、まず質の高いドキュメントを用意することをおすすめします。ドキュメントが適切である(平均より標準偏差が1個分大きい)チームの方が、ドキュメントが適切でない(平均より標準偏差が1個分小さい)チームより新規採用者の生産性が30%高いことは注目に値します。

火に油を注がないように、オフィス回帰に関する議論は手短かに済ませます。柔軟性が重要であることについては、[7章 - 組織文化への投資なしには何事もうまくいかない](#)で述べました。さらに、本調査のデータは実験的なものではありません。結果に偏りが生じるような要因をコントロールするよう努めてはいますが、勤務形態の利点は複雑で社会的に深い問題であるため、はっきりとした結論を出すことは困難です(この件に関する研究や考察を読む際に留意する価値はあります)。本調査のデータで明らかなのは、柔軟性が生産性に好影響を与えるということです。働く場所が生産性に好影響を与えるかどうかは明らかではありません。

新規採用者についても同じことが言えます。対面での共同作業が新規採用者に特に有益だという裏付けは確認されませんでした。生産性の最適化を試みる場合、新規採用者がいつでもどのように働くのかについて柔軟性を持たせる方が、出勤を強制するよりも確実なようです。もちろ

ん、組織は生産性のためだけに最適化を行っているわけではありませんし、そうすべきでもないでしょう。また、DORAは、単なるアウトプット(書かれたコードの行)ではなく価値につながる仕事、そして燃え尽き症候群もトイレも生じない仕事という観点から生産性を考えています。



あとがき

最後に

今年の調査にご協力いただき、また本レポートをご覧いただき、ありがとうございました。DORA は、チームの働き方とチームが達成できる成果の関係を探る最適な方法を常に模索しています。

長年にわたる研究プログラムでわかった最も重要なことは、継続的改善の考え方と手法を取り入れているチームが最高の成果を達成できるということです。

DORA が調査してきた能力は、成果を高めるダイヤルとして使用できます。こうしたダイヤルの中には、個人の手の届く範囲にあるものもあれば、組織全体の協調的な取り組みでしか利用できないものもあります。組織でどのダイヤルを調整する必要があるのかを特定し、その調整に投資しましょう。

改善の取り組みに終わりはありませんが、個人、チーム、組織に長期的な成功をもたらすことができます。リーダーと実務者は、この改善の取り組みを促進する役割を共に担っています。

調査結果を実践に移すには

今回の調査結果を、所属組織やチームの観点から、またユーザーに提供しているサービスの観点から精査してみてください。

DORA コミュニティ (<https://dora.community>) に参加して、経験を共有し、他者から学び、継続的改善に取り組んでいる仲間からヒントを得ましょう。

謝辞

本レポートは、世界中にいるたくさんの情熱的な協力者の支えがあって毎年公開されています。アンケートの質問の考案、ローカライズ、分析、執筆、編集、組版など、制作のすべての段階において、この大規模なプロジェクトを実現するために協力してくれました。著者一同、これらすべての方々のご意見、助言、友情に感謝いたします。

協力者

コアチーム

James Brookbank
Kim Castillo
Derek DeBellis
Nathen Harvey
Michelle Irvine
Amanda Lewis
Eric Maxwell
Steve McGhee
Dave Stanke
Kevin Storer
Daniella Villalba
Brenna Washington

編集者

Mandy Grover
Jay Hauser
Stan McKenzie
Anna Eames Mikkawi
Mike Pope
Tabitha Smith
Olinda Turner

アンケートの ローカライズ

Daniel Amadei
Kuma Arakawa
William Bartlett
Antonio Guzmán
Shogo Hamada

Yuki Iwanari
Vincent Jobard
Gustavo Lapa
Mauricio Meléndez
Jeremie Patonnier
Miguel Reyes
Pedro Sousa
Laurent Tardif
Kimmy Wu
Vinicius Xavier
Yoshi Yamaguchi

各分野のアドバイザー、 専門家

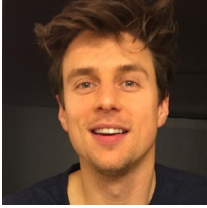
Jared Bhatti
Lisa Crispin

Rob Edwards
Dave Farley
Steve Fenton
Nicole Forsgren 博士
Aaron Gillies
Denali Lumma
Emerson Murphy-Hill
Harini Sampath
Robin Savinar
Dustin Smith
Jess Tsimeris
Laurie Weingart 博士
Betsalel (Saul)
Williamson
Jeffrey Winer 博士

スポンサー



著者



Derek DeBellis

Google の定量的ユーザー エクスペリエンス リサーチャーである Derek は、DORA の主任調査員です。Derek は、アンケート調査やログ分析のほか、プロダクトや機能がユーザーに独自の大きな価値を提供していることを示すコンセプトの測定方法の考案に注力しています。また、人と AI の相互作用、COVID-19 (新型コロナウイルス感染症) が禁煙に及ぼす影響、NLP エラーを想定した設計、プライバシーの考察における UX の役割、チーム文化、従業員のウェルビーイングや生産性と AI の関係に関する記事を公開しています。現在は、信条と力の伝播のシミュレーション方法を探る課外研究に取り組んでいます。



Amanda Lewis

DORA コミュニティの開発リードである Amanda Lewis は、Google Cloud の DORA アドボカシー チームのデベロッパーリレーションズ エンジニアです。Amanda はそのキャリアを通じて、開発者、運用者、プロダクト マネージャー、プロジェクト管理者、リーダーの関係構築に携わってきました。また、e コマースプラットフォーム、コンテンツ マネジメント システム、オブザーバビリティ ツールの開発チームで働き、開発者をサポートしてきました。このような関係と対話は、顧客満足度や業績の向上につながります。Amanda は経験と共感力を生かし、チームがソフトウェア デリバリーと信頼性の手法を理解し実践するための支援を行っています。



Daniella Villalba

Daniella Villalba は Google のユーザー エクスペリエンス リサーチャーです。Daniella は開発者が幸福になり生産的になる要素を知るためにアンケート調査を行っています。Google に入社する前は、瞑想トレーニングの利点や、大学生の体験に影響を及ぼす心理社会的な要素について研究していました。Daniella はフロリダ国際大学から、実験心理学で博士号を授与されています。



Dave Farley

Continuous Delivery Ltd の創設者でマネージング ディレクターの Dave Farley は、『Modern Software Engineering』の著者であり、ベストセラー『Continuous Delivery』の共著者です。『Reactive Manifesto』の共著者でもあり、オープンソースの LMAX Disruptor プロジェクトで Duke Award を受賞しています。Dave は継続的デリバリーのパイオニアであり、ソートリーダーであり、CD、DevOps、テスト駆動開発 (TDD)、ソフトウェア デザインの熟練した実務者です。長年にわたり、高パフォーマンスのチーム作り、組織の成功のサポート、優れたソフトウェアの作成に携わっています。自身の経験や技術を世界中のソフトウェア 開発者と共有し、ソフトウェアの設計、品質、信頼性の向上を支援しています。Dave はコンサルティング¹、YouTube チャンネル²、トレーニング コースを通じて専門知識を共有しています。



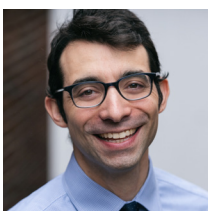
Eric Maxwell

Eric Maxwell は Google の DevOps トランスフォーメーションの取り組みでリーダーを務めており、価値の迅速な提供による事業の改善について、世界のトップ企業に助言を行っています。Eric は、キャリアの前半を現場エンジニアとして過ごす中でさまざまな自動化の作業に関わり、他の実務者の共感を得てきました。Eric は、Google の Cloud アプリケーション モダナイゼーション プログラム (CAMP) の共同創設者で、DORA チームのメンバーです。Google に入社する前は、Chef Software のエキサイティングな環境で同僚とともに優れた成果を生み出していました。



James Brookbank

James Brookbank は Google のクラウド ソリューション アーキテクトです。ソリューション アーキテクトとして、複雑な技術的問題を解決し、アーキテクチャに関する専門的なアドバイスをすることで、Google Cloud のお客様を支援しています。Google に入社する前は、IT インフラストラクチャと金融サービスを中心に、さまざまな大企業に勤めていました。



Jeffrey Winer 博士

Jeffrey P. Winer 博士は、ボストン小児病院トラウマおよびコミュニティレジリエンス センター (TCRC) の所属心理士、行動医療システム コンサルタント、心理社会的治療法開発者であり、ハーバード大学医学大学院の助教授でもあります。TCRC の同僚とともに、難民や移民のバックグラウンドを持つ若者や家族に対する、文化に対応したトラウマに配慮した心理社会的介入の構築、テスト、普及、実施に注力しています。『Mental Health Practice with Immigrant and Refugee Youth: A Socioecological Framework』³ の共著者でもあります。米国とカナダにおけるプログラムのコンサルティングも行っています。開発や改良を支援した心理社会的予防および介入ツールは、今では世界中で使用されています。詳細については <http://www.drjeffwiner.com> をご覧ください。

¹ <https://continuous-delivery.co.uk/engineering-for-software>

² <https://www.youtube.com/@ContinuousDelivery>

³ <https://www.apa.org/pubs/books/4317536>



Kevin Storer

Kevin M. Storer は Google のユーザー エクスペリエンス リサーチャーです。ソフトウェア開発チームが DevOps ツールをどのように操作し、何をするのかを知るための研究を率いています。Google に入社する前、カリフォルニア大学アーバイン校で情報学の博士号を取得しました。公共部門と民間部門の両方にわたり、人間中心のプログラミング、開発者体験、情報行動、アクセシビリティ、ユビキタス コンピューティングをテーマとした影響力のある論文発表を行っています。



Kim Castillo

Kim Castillo は Google のユーザー エクスペリエンス プログラム マネージャーです。Kim は研究業務の監督から本レポートの公開まで、DORA を支える部門横断的な取り組みを率いています。Google Cloud の Duet AI の UX リサーチにも携わっています。Google に入社する前は、ソフトウェア デリバリー分野でキャリアを積み、技術プログラム管理やアジャイル コーチングに携わりました。Kim のキャリアの根底にあるのは、母国フィリピンにおける超法規的殺人、都市貧困層対象の開発事業、コミュニティのトラウマおよびレジリエンスをテーマとした、心理社会学的研究です。DORA は、ソフトウェア デリバリーの手法、心理的安全性、ティール組織など、Kim が熱中するものの宝庫です。



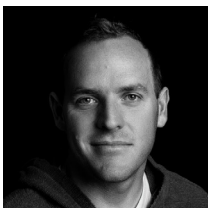
Michelle Irvine

Michelle Irvine は Google のテクニカル ライターで、技術ドキュメントの影響と制作に関する研究を率いています。Google に入社する前は教育系出版社で勤務経験があり、また、物理シミュレーションソフトウェアのテクニカル ライター職に従事していました。Michelle はウォータールー大学から物理学の理学士号と、レトリックおよびコミュニケーション設計の修士号を授与されています。



Nathen Harvey

Nathen Harvey は Google Cloud のデベロッパーリレーションズ エンジニアリング マネージャーとして、DORA アドボカシー チームを率いています。これまでに数々の優れたチームやオープンソース コミュニティとともに仕事をする機会を得て、DevOps と SRE の原則とプラクティスを適用するための支援を行いました。3 年間にわたり、Accelerate State of DevOps レポートの共著者を務めています。Nathen は『97 Things Every Cloud Engineer Should Know』を共同で編集し、寄稿しました。



Steve McGhee

信頼性アドボケイトの Steve McGhee は、信頼性の高い世界水準のサービスを構築し運用する最善の方法をさまざまなチームに伝えています。以前は、Google のサイト信頼性エンジニアとして 10 年以上を過ごし、Google 検索、YouTube、Android、Google Cloud のグローバル システムのスケーリング方法について学びました。また、カリフォルニア州、日本、英国で複数のエンジニアリングチームを管理しました。カリフォルニア州を拠点とする企業のクラウド移行を支援したこともあります。

方法論

本章では、このレポートがどのようにして初期のアイデアから（アイデアの完成度が少し上がって）公開に至っているのかを概説します。本レポートがどのように制作されたのかについて多くの疑問に答えることで、ご自身で調査を始める際の青写真となれば幸いです。

ステップ 1. パフォーマンスの高い技術主導型組織にとって重要だと思われる一連の成果を挙げる

これは非常に重要です。DORA のプログラムは、目標に向けて人員を導く手助けを基本としています。人員、組織、チームの行く先を把握していなければ、良くないスタートを切ることになります。どのようにして把握すればよいでしょうか。DORA は、質的調査（人員、チーム、組織で達成したいことを人員に尋ねること）、アンケート、広範なコミュニティとの交流、多くのワークショップを組み合わせています。その中で、一貫して次のような成果が挙げられています。

- **組織パフォーマンス:** 組織は収益を上げるだけでなく、顧客のために、また、その周囲に広がるコミュニティのために価値を生み出す必要があります。
- **チームパフォーマンス:** アプリチームやサービスチームが価値を生み、イノベーションを起こし、協力する能力が必要です。
- **従業員のウェルビーイング:** 組織やチームで採用する戦略は、従業員にとって有益なものである必要があります。つまり、燃え尽き症候群を減らし、仕事への満足度を高め、価値ある成果を生み出す能力（生産性）を高めることが求められます。

人員が次のような目標を話すのも耳にします。

- **ソフトウェアデリバリーパフォーマンス:** チームがソフトウェアを迅速かつ適切にデプロイできること。
- **運用パフォーマンス:** 公開したソフトウェアが信頼性の高いユーザー エクスペリエンスを提供すること。



ステップ 2. 成果がいつ、なぜ、どのように達成されるのかについて仮説を立てる

ステップ 1 で挙げた成果を踏まえ、チームをそこへ到達させる道筋について仮説を立てる必要があります。これには、成果に確実な影響を与えると思われる要因を探ることが含まれます。「他のすべてが等しい場合、x は y に影響する」というような法則を見つけたいわけです。この情報は、どのような変革を試みるかについて、実務者がデータに基づいた決定を行う際に役立ちます。

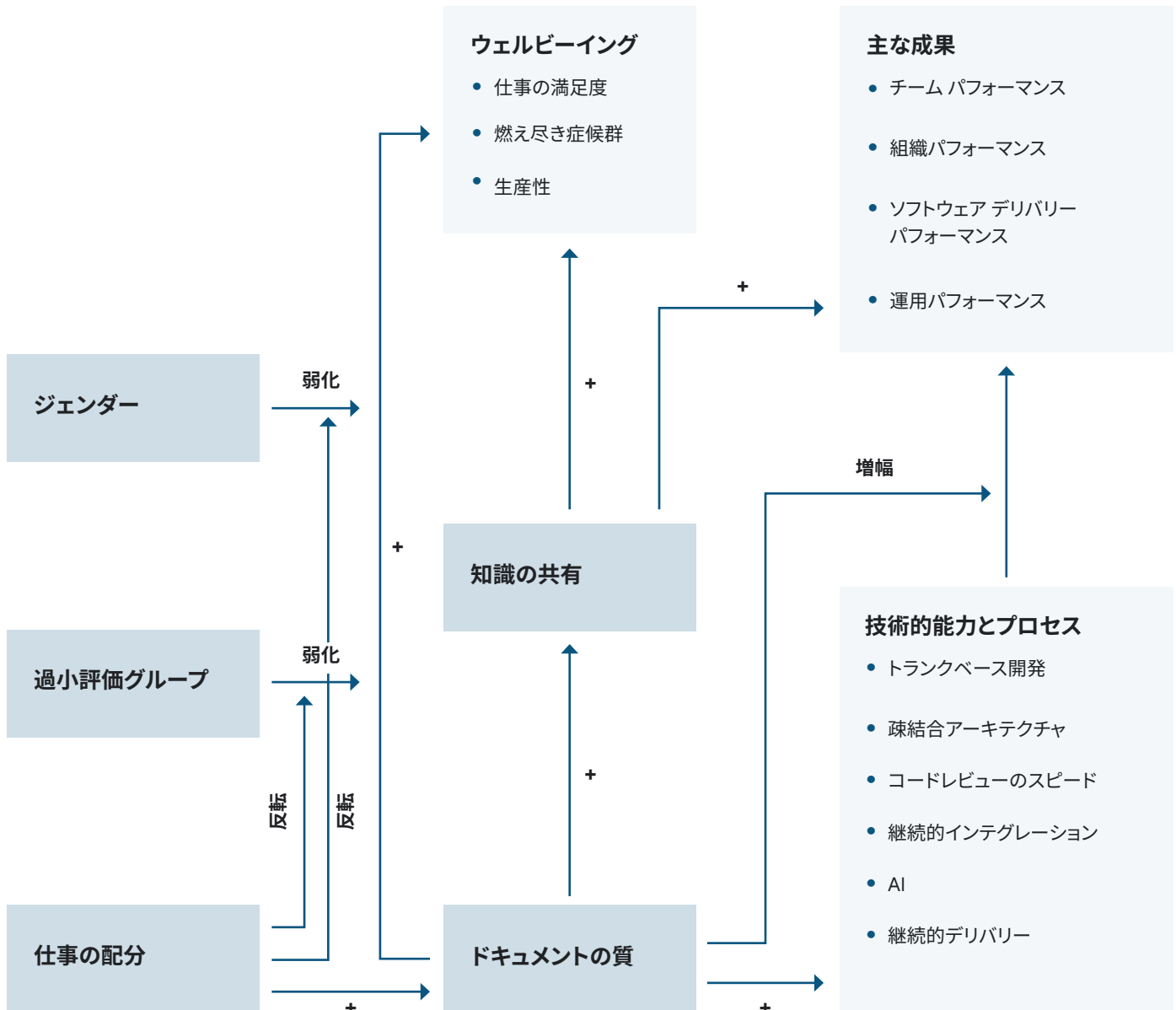
どの道筋が影響を与えるのかを知りたいだけでなく、そうした道筋の影響が増減する条件も探りたいと考えています。これはつまり、「いつ」、「誰のために」という問いになります。たとえばドキュメントの質は、概して、燃え尽き症候群を劇的に減らすことが証明されています。しかし、過小評価グループだと報告した回答者を調べてみると、反対に、ドキュメントの質が燃え尽き症候群を増やしています。どのチームでも、どんな人でも、平均的であることは稀であるため、こうした影響の条件を把握することは極めて重要です。

さらに、理由や方法を説明するメカニズムについて仮説を立てます。今年は、去年の調査結果と広範な関連文献に基づき、過小評価グループを自認する人員ほど燃え尽き症候

群が多くなるという仮説を立てました。最初に出てくる疑問は「なぜこのようなことが起こるのか」です。この疑問に答えようと、DORA はテストするメカニズムの候補について仮説を立てました。たとえば過小評価グループを自認する人員ほど手間にかかる業務を引き受ける（または割り振られる）ため、燃え尽き症候群が多くなるのかもしれない。



アンケートを構築して分析を行うために、こうした仮説はマッピングされます。影響、条件、メカニズムを含む、ドキュメントに関する章の仮説モデルの例を以下に示します。

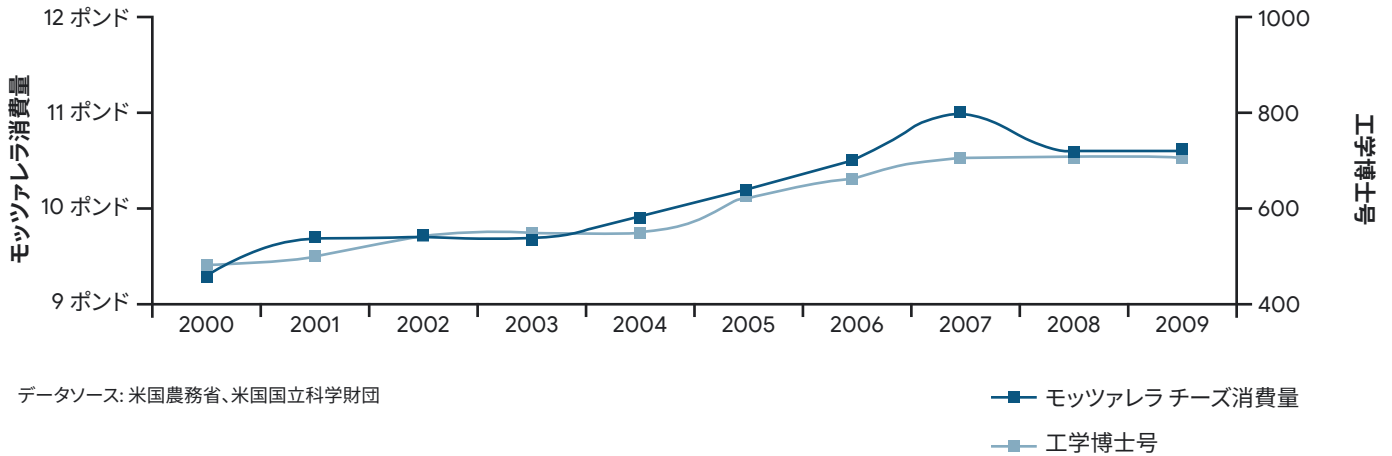


ステップ 3. 起こり得る交絡について仮説を立てる

データについて議論したことがあれば、おそらく疑似相関に遭遇したことがあるでしょう。次の例のように、疑似相関を数多く提示しているウェブサイト¹を見たことがあるかもしれません。

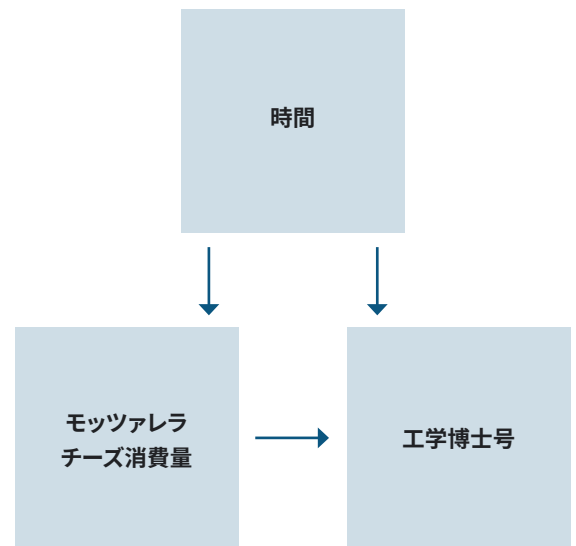
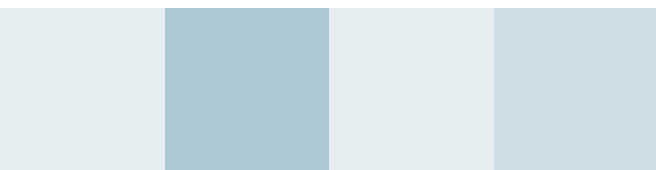
モッツァレラ チーズの 1 人あたりの消費量と土木工学の博士号取得数の相関

相関性: 95.86% (r=0.958648)



工学博士号とモッツァレラ チーズ消費量に因果関係があるとは思えません。やはり、この関係の背後には「時間」という交絡要素 (正しい因果関係の判断を妨げる要素) が潜んでいます。モッツァレラ チーズ消費量と工学博士号が同じ期間に正のトレンドがあれば、正の相関がある可能性が高くなります。

モデルに時間を含めるか、データのトレンド除去を行えば、おそらくこの関係は無効になります。モデルを次のように描くことができます。



¹ <https://www.tylervigen.com/spurious-correlations>

時間(第3の変数)を考慮しなければ、データはモッツァレラチーズ消費量と工学博士号の間の疑似相関を示すかもしれません。

これに関しては、Dagitty (<https://dagitty.net/dags.html>) など、研究者を支援するツールがあります。このツールを使用すると因果モデルを特定できます。XがYに及ぼす影響を適切に推定できるように、モデルの意味、考慮する必要のあること、考慮すべきでないことを教えてください。Dagittyのようなツールを使用すると、ある相関が因果関係を意味しないという結論に至ることもあります。同時に、因果関係についての考え方についても学べます。

50時間に及ぶアンケートを実施するか、全知全能の神でもない限り、研究者の推定にバイアスをもたらす要素をすべてキャッチすることは不可能です。それでも、DORAは皆様が目指す成果に対するさまざまな活動、技術、仕組みの影響を正確に推定できるよう、道筋のバイアスを考慮するために最善を尽くしています。結局のところ、実務者の多くが知りたがっているのは、どのような要因がこれらの主な成果に影響を与えるのかということです。バイアスを考慮できないモデルは、実務者が必要とするガイダンスを提供できません。モッツァレラチーズがソフトウェアデリバリーパフォーマンスを向上させるなどと発表したくはありません。しかしそのような間違いは、思いのほか起こりやすいことなのです。



ステップ 4. アンケートを作成する

アンケートの作成には、運用化、体験、ローカライズという3つの要素があります。

運用化

対象のコンセプトを適切かつ確実に捉える尺度が必要です。抽象的なコンセプトを測定可能なものに変換することが運用化の技術です。このような尺度が、材料としてすべての分析のベースとなります。尺度で明確なシグナルが得られない場合、どのようにすれば残りの分析を信頼できるでしょうか。生産性など、とらえどころのないコンセプトをどのようにして測定すればよいのでしょうか。燃え尽き症候群や運用パフォーマンスについてはどうでしょうか。

まず文献に目を向け、適切な尺度がすでに存在していないか確認します。有効性を検証済みの尺度をアンケートで使用できれば、質問に関して集めたすべての文献とアンケートとの橋渡しができます。DORA は現在 Westrum の組織文化のタイプを使用していますが、これが過去に検証された尺度の再利用の一例です。

しかし DORA の研究分野では、まだ多くのコンセプトが検証されていません。その場合、質的調査を行ってコンセプトがどのように理解されているのかを解き明かし、コンセプトの複雑さについてさらに哲学的な文献に目を通します。

アンケート体験

DORA はアンケートを、わかりやすく、簡単で、必要以上に長くなく、広く利用できるものになりたいと考えています。質問事項の多さや、質問に答えるために必要となる技術的知識、そして一部の手法にはさまざまな専門的名称が存在することを考えると、これは難しい目標です。DORA はリモートでモデレーターなしの評価を行い、これらの点においてアンケートが一定のしきい値を超えていることを確認しています。これには、プロセスを何度か繰り返す必要があります。

ローカライズ

毎年、世界中の人がアンケートに回答してくださっています。今年はアンケートを英語、スペイン語、フランス語、ポルトガル語、日本語にローカライズすることで、より多くの人にアンケートに参加していただけるようにしました。これは、DORA コミュニティのメンバーが率いた草の根活動でした。世界中の Google 社員がこの取り組みに協力しました。また、フランス語へのローカライズでご協力いただいたこの分野のパートナー、Zenika (<https://www.zenika.com>) に感謝いたします。DORA はこのような取り組みをさらに拡大し、真に文化の垣根を越えたアンケートにしたいと考えています。

ステップ 5. アンケートの回答を集める

回答者の募集には複数のチャネルを利用しました。チャネルは2つに分類されます。オーガニックとパネルです。

オーガニックアプローチでは、ソーシャルのあらゆる手段を利用して、アンケートを実施することを伝えます。ブログ投稿や、メール キャンペーンを行います。また、ソーシャル メディアに投稿して、コミュニティのメンバーにも拡散を依頼します(スノーボール サンプリング)。

パネルアプローチは、オーガニック チャネルを補うためにに行います。ここでは、従来から過小評価されているグループの人を広範な技術コミュニティで募集し、また、特定の業界や組織から十分な回答を得られるよう努めます。つまり、オーガニック アプローチでは行えない、募集対象のコントロールを行います。パネル アプローチには、単に十分な数の回答者を確保する役割もあります。オーガニック アプローチでは、実施予定の分析に必要な回答が得られるかどうか分からないためです。

ステップ 6. データを分析する

分析の中心部分は、データ クリーニング、測定の検証、モデルの評価という3つのステップで構成されています。

データ クリーニング

データ クリーニングの目的はノイズに対するシグナルの比を高めることです。さまざまな理由により、回答の一部はノイズになります。回答パターンの中には、回答者が注意散漫だったことや、アンケートを急いでいたこと、誠実に回答していないことを示すものもあります。DORA のデータでは非常に稀ですが、起こり得ることで、実際の体験を正しく表していない兆候が見られる回答はノイズであるため、分析から除外されます。ノイズを除去する際の課題は、誤ってシグナルを除去しないようにすることです。特に、バイアスのかかった形で、あるいは仮説を正しく見せるような形でシグナルを除去するのは避けなければなりません。たとえば、「A の値が高い場合に B の値が低くなることはない」と結論付けた場合に、それに当てはまる (A の値が高く B の値が低い) 回答者を除外しようとしてしまいがちです。そうすれば、データが想定どおりになり、仮説に合った結果になる確率が高まるからです。



測定の検証

本レポートの冒頭で、測定しようとしているコンセプトについて述べました。表現の仕方はさまざまですが、一つの見方として、このコンセプトの尺度を変数と呼びます²。これらの変数はモデルの材料であり、モデルは DORA の研究に含まれる要素です。このような尺度の妥当性を分析する方法は、大まかに内的なものとの外的なものに分けられます。

尺度の内的妥当性を把握するには、あるコンセプトの存在を示すと思われるものを調べます。たとえば質の高いドキュメントであることは、問題解決のためにドキュメントを使用する人がいることによって示される可能性があります。

調査対象の複合概念は多面的であるため、変数の大部分は複数の指標で構成されます。変数の多面的な性質を理解するには、この複合概念を表すのに使用する項目がどの程度機能するかをテストします。うまく機能する場合（つまり、高い水準で同じ分散を共有している場合）、その根底には何かがある、たとえば対象のコンセプトがあると想定します。

例として、幸福について考えてみます。幸福は多面的です。幸福であれば、ある種の感情を抱いており、ある種の行動をとり、ある種の考え方をすると予想されます。特定の感情、思考、行動のパターンの根底に幸福があると想定するので、そのため、幸福が存在していれば、ある種の感情、思考、行動と一緒に現れると予想されます。次に、そのような感情、思考、行動についての質問をします。実際に一緒に現れるかどうかは確認的因子分析でテストします。

今年はこの分析に lavaan R パッケージを使用しました³。lavaan は適合に関するさまざまな統計情報を返します。これは、複合概念が質問への答えを実際に表しているかどうかを理解するために役立ちます。あるコンセプトの指標がう

まく機能しない場合は、そのコンセプトを測定する確実な方法が見つからないことが明らかであるため、そのコンセプトを修正するか破棄する必要が生じることがあります。

複合概念の外的妥当性とは、その構成要素がどの程度現実適合するかを調べることです。たとえば、ある複合概念が他の複合概念と特定の関係を持つことが予想される場合があります。幸せと悲しみのように、2つの複合概念が負の関係を持つことが予想される場合もあります。幸せの尺度が悲しみと正の相関を示すようになれば、尺度が理論に疑問を抱くでしょう。同様に、2つの複合概念が正の関係を持つ強い関係ではない、ということが予想される場合もあります。生産性と仕事の満足度には正の相関がありそうですが、同一であるとは思いません。相関しすぎていると、同じものを測定しているようだと言うかもしれません。つまり、2つのコンセプトの違いを拾えるほどには尺度が調整されていないか、仮定した差異が実際には存在しないということです。

² Moore, Will H., and David A. Siegel. A mathematics course for political and social research. Princeton University Press, 2013.

³ Rosseel, Y. "lavaan: An R Package for Structural Equation Modeling," Journal of Statistical Software, 48(2), 2012.1–36. <https://doi.org/10.18637/jss.v048.i02>

モデルの評価

ステップ 2 と 3 で仮説モデルを構築しました。ステップ 6 でクリーンなデータが得られたら、そのモデルがどの程度データに適合するかを確認します。今年は、帰無仮説（つまり影響のないもの）に対するデータの該当可能性ではなく、データに対するさまざまな仮説の妥当性を理解できるようにするために、ベイズアプローチを採用しました。R で使用する主なツールは blavaan⁴ と rstanarm⁵ です。ある影響が大幅または劇的であって、単なる軽微なものではない確率をテストできます。モデルの評価では節減を追求します。これは、非常に単純化されたモデルから始めて、複雑さが正当化されなくなるまで複雑さを追加するという方法です。たとえば、組織パフォーマンスはソフトウェア デリバリー パフォーマンスと運用パフォーマンスの相互作用の産物であると予測します。

単純化したモデルに相互作用は含まれていません。

組織パフォーマンス ~ ソフトウェア デリバリー
パフォーマンス + 運用パフォーマンス

第 2 のモデルが相互作用を追加します。

組織パフォーマンス ~ ソフトウェア デリバリー パフォー
マンス + 運用パフォーマンス + ソフトウェア デリバリー
パフォーマンス x 運用パフォーマンス

「Regression and other stories」⁶ と 「Statistical Rethinking」⁷ での推奨事項に基づき、追加の複雑さが必要かどうかを判断するために、一つ抜き交差検証 (LOOCV) と渡辺・赤池の広く使える情報量規準⁸ を使用します。

⁴ Merkle, Edgar C., and Yves Rosseel. "blavaan: Bayesian structural equation models via parameter expansion." arXiv preprint, 2015. <https://doi.org/10.48550/arXiv.1511.05604>

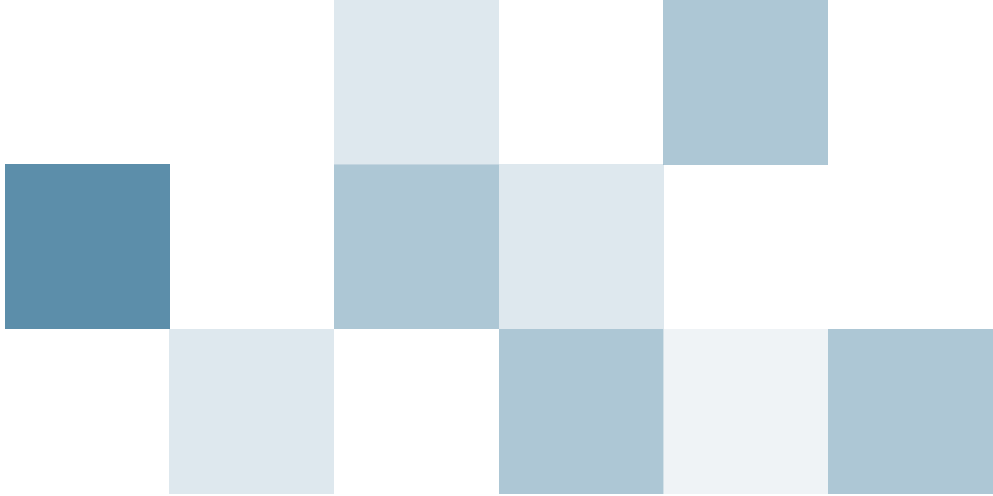
⁵ Goodrich, Ben, Jonah Gabry, Imad Ali, and Sam Brilleman. "rstanarm: Bayesian applied regression modeling via Stan." R package version 2, no. 1 (2020).

⁶ Gelman, Andrew, Jennifer Hill, and Aki Vehtari. Regression and Other Stories (Cambridge University Press, 2020).

⁷ McElreath, Richard. Statistical rethinking: A Bayesian course with examples in R and Stan. Chapman and Hall/CRC, 2018.

⁸ Vehtari, Aki, Andrew Gelman, and Jonah Gabry. "Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC." Statistics and Computing, 27, 2017.1413-1432. <https://doi.org/10.1007/s11222-016-9696-4>





ステップ 7. 調査結果を報告する

次に、結果をチームで確認しました。今年は皆でコロラド州ボルダーに数日間滞在し、データを対象分野の専門家の経験と組み合わせて検証しました。これを本レポートのすべての章について、仮説ごとに行いました。データの解釈には、偏り、憶測、事実に基づかない逸話、飛躍のリスクが常にあります。このようなリスクは、多様なバックグラウンドを持つ人たちに参加してもらい、疑問の提示、異なる意見、独自の観点、好奇心を奨励することで軽減されました⁹。

結果を基に、本レポートの著者たちはそれぞれの場所に戻って執筆に取りかかりました。執筆の過程を通して、編集

者や対象分野の専門家に意見を求めました。こうした観点を得ることは、DORA の考えを伝えるために不可欠でした。このデータの分析の担当者は、私たちの主張がデータの内容から逸脱していないことを確認する役割を担いました。

これらの章は、優れたデザインパートナーである BrightCarbon によってまとまりのあるデザインに仕上げられました¹⁰。

ステップ 8. 調査結果をコミュニティとともに活用する

DORA はこうした調査結果を活用、解釈する方法を考案するにあたり、コミュニティの関与を頼りにしています。推奨事項は細心の注意を払って提示するよう努めていますが、結局のところ、明らかになった結果に基づいてチームが実践できることは無数にあります。たとえば疎結合アーキテクチャは、測定する成果に基づいた有益な手法だと思われます。しかし、疎結合アーキテクチャを確立する方法は 1 つだけではありません。コミュニティとしてアプローチを生み出し、共有することは、継続的改善の唯一の方法です。DORA の想定する世界は、お客様やその所属チーム、所属組織が実際に活動している地域や状況を解釈し、抽象化したものです。

DORA のグローバルな実践コミュニティに参加するには、DORA コミュニティのサイト (<https://dora.community>) をご覧ください。

⁹ Stasser, G., & Titus, W. (1985). "Pooling of unshared information in group decision making: Biased information sampling during discussion." *Journal of Personality and Social Psychology*, 48(6), 1985.1467-1478. <https://doi.org/10.1037/0022-3514.48.6.1467>

¹⁰ <https://www.brightcarbon.com/>

ユーザー属性と企業特性

アンケートの対象者

DORA の研究プログラムは、パフォーマンスの高い技術主導型組織の能力、手法、尺度について 10 年近くにわたって調査を行ってきました。その中で、さまざまな業界のあらゆる規模の組織で働く 36,000 人超のプロフェッショナルの方々にお話を伺いました。皆様のご協力に感謝いたします。今年は、全世界から各種の業界に属する約 3,000 人の現役のプロフェッショナルに経験を共有していただき、パフォーマンスの高い技術主導型組織の原動力となる要素について理解を深めることができました。

今年のユーザー属性と企業特性に関する質問では、Stack Overflow による研究を活用しました。2022 Stack Overflow Developer Survey¹には 70,000 人を超える回答者が参加しました。このアンケートはさまざまな理由からすべての技術者を対象とはしませんでした。開発者界の国勢調査に限りなく近いものになりました。そこから得られる母集団を意識することで、データ内の回答バイアスを見つけ、調査結果をどこまで一般化すべきかを検討できます。このデータと、Stack Overflow Developer Survey で尋ねたユーザー属性と企業特性に関する質問は、内容が十分に練られており、借用する価値があります。DORA のサンプルセットは Stack Overflow Developer Survey

と比較して、女性、障がいのある参加者、大規模な組織で働く参加者の割合が高くなっています。人種や民族に関しては Stack Overflow のサンプルセットと同様です。

今年のアンケートでは、オーガニックの回答者の数が 2022 年と比較して 3.6 倍増加しました。

今年のアンケートでは
2022 年と比較して
3.6 倍
オーガニックの回答者の
数が増加

¹ <https://survey.stackoverflow.co/2022#overview>

ユーザー属性

ジェンダー

2022年と比較して、今年のサンプルは女性回答者の割合が少なくなりました(18% に対し 12%)。

ジェンダー	全回答者に占める割合
回答しない	3%
自由回答	2%
女性	12%
男性	81%

障がい

障がいについては、Washington Group Short Set²の基準に従い、6つの観点で識別しました。障がいについての質問を行うのは今年で5年目です。障がいのある回答者の割合は、2022年の11%から減り、2023年は6%でした。

障がい	全回答者に占める割合
障がいのいずれもあてはまらない	87%
あり	6%
回答しない/回答なし	7%

過小評価グループ

過小評価グループのメンバーであるという認識は、人種、性別、その他の特性に関連している場合があります。過小評価についての質問を行うのは、今年で6年目です。過小評価グループを自認する回答者の割合は2022年の19%から多少減り、2023年は15%でした。

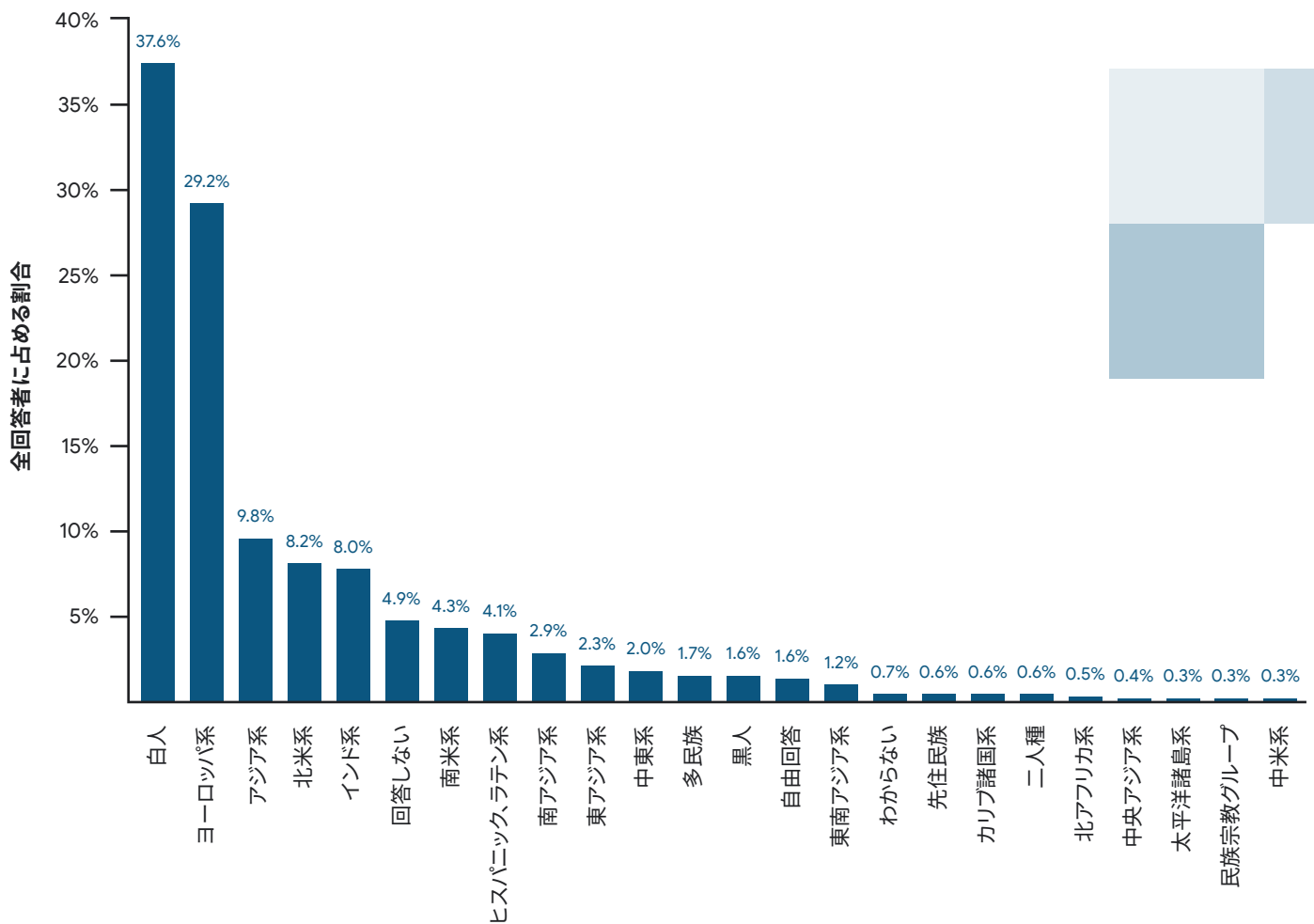
過小評価グループ	全回答者に占める割合
いいえ	77%
はい	15%
回答しない	7%

² <https://www.washingtongroup-disability.com/question-sets/wg-short-set-on-functioning-wg-ss/>



人種や民族

2022 Stack Overflow Developer Survey³の質問を採用しました。先述のとおり DORA のサンプルセットはこれと似ていますが、1つだけ、ヨーロッパ系の人の比率が低いという顕著な相違点があります。

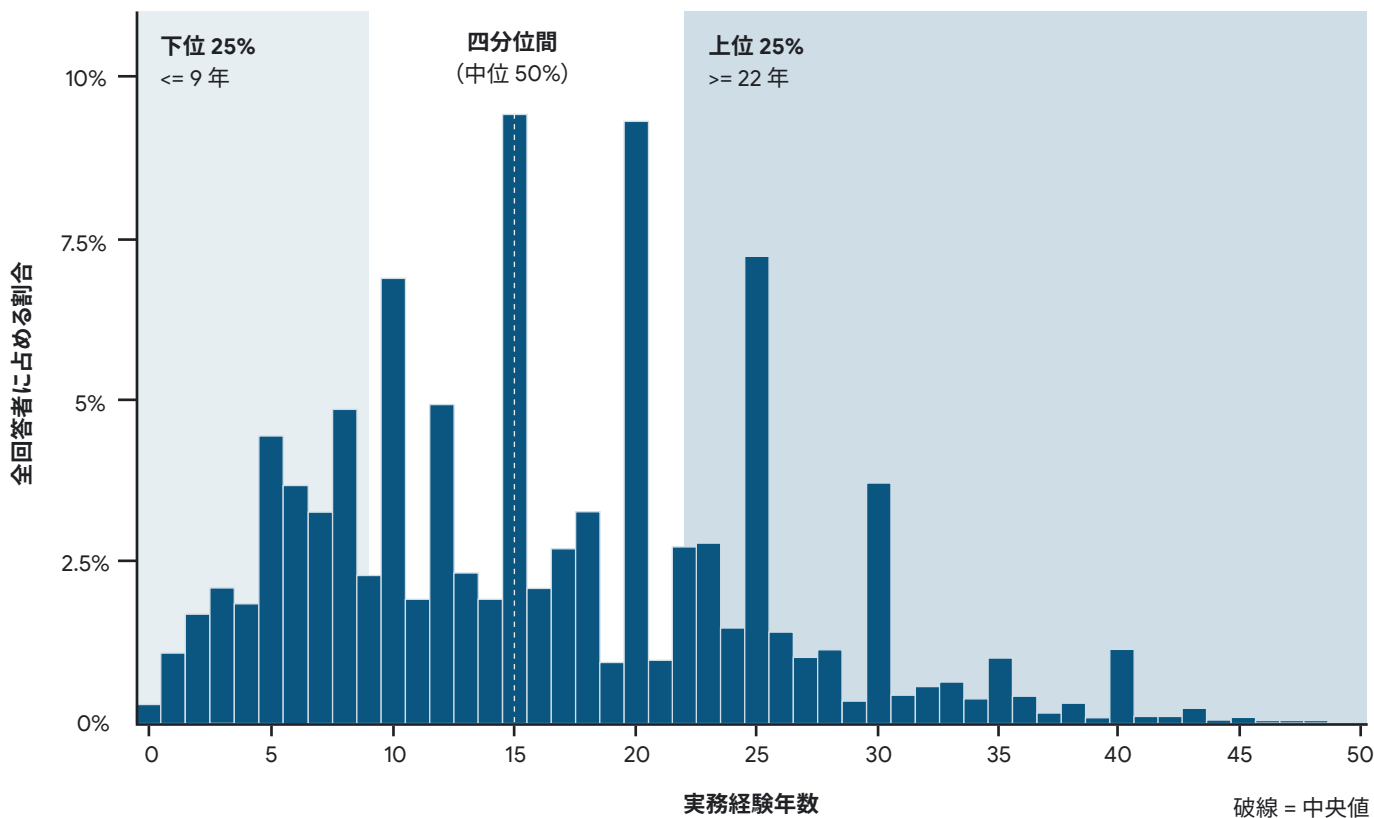
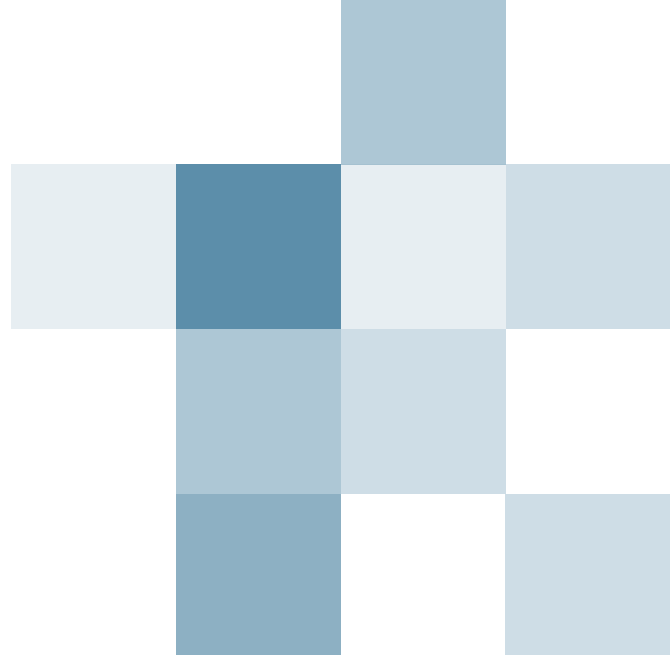


³ <https://survey.stackoverflow.co/2022#overview>

経験年数

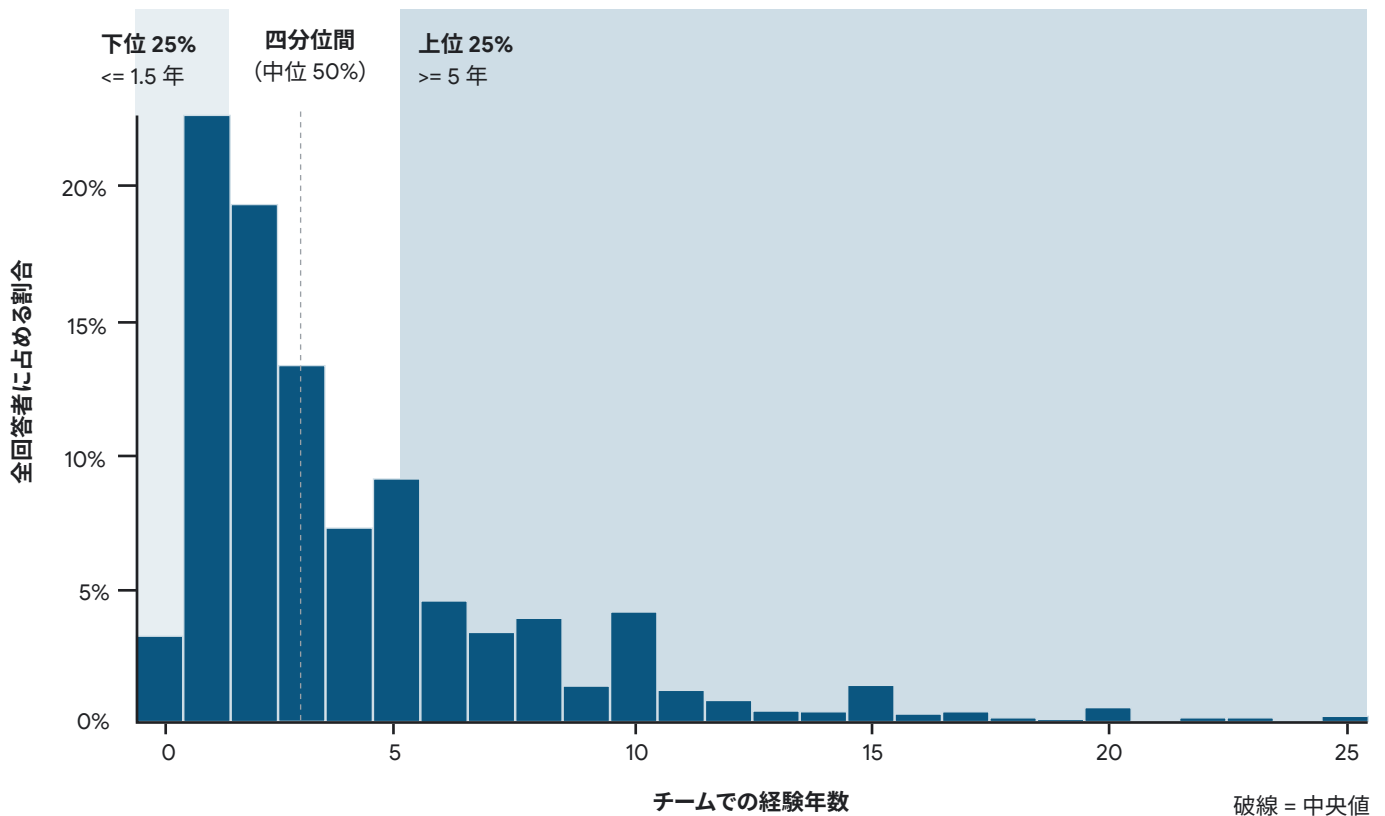
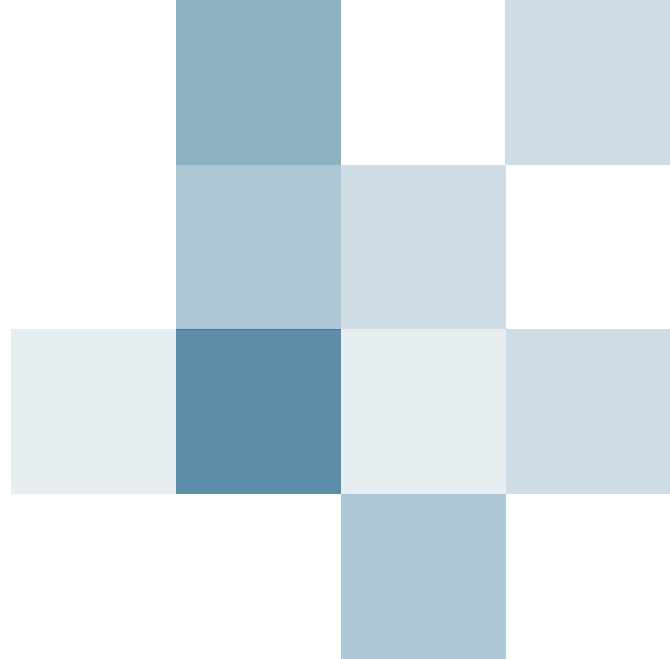
業務の経験

回答者の実務経験年数を把握しようと考え、ここでは「実務経験は何年ですか。」と尋ねました。まとめると、DORAがデータを収集した実務者グループはかなり経験豊富でした。内訳は、回答者の50%が実務経験15年以上、回答者の25%が実務経験22年以上、回答者の25%が実務経験9年以下です。結果論ですが、回答者が何を「実務」としてカウントしているのかは明らかではありません。



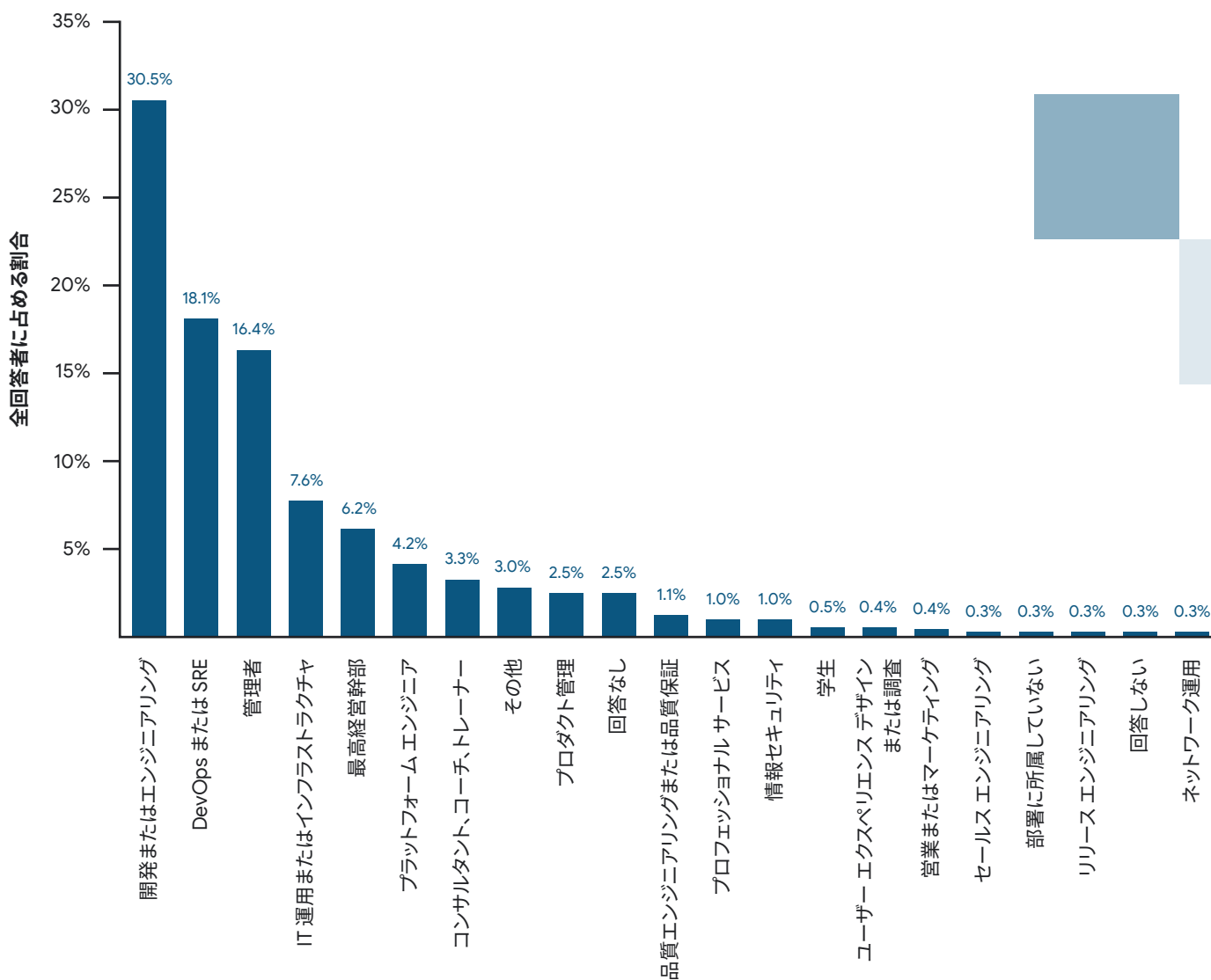
チームでの経験年数

新しいチームで働くと、やり直して少しリセットしたような気分になることがあります。それを調べるために、「現在のチームで何年働いていますか。」と尋ねました。実務経験は長いようでも、回答者の多くはチームの新顔です。回答者の50%は新しいチームに所属して3年未満、回答者の25%はチームに所属して1.5年以下でした。チームに所属して5年以上の回答者は25%だけでした。これは、回答者の継続的改善の考え方を反映しているのでしょうか。経済の変動性と不安定性を反映しているのでしょうか。



役割

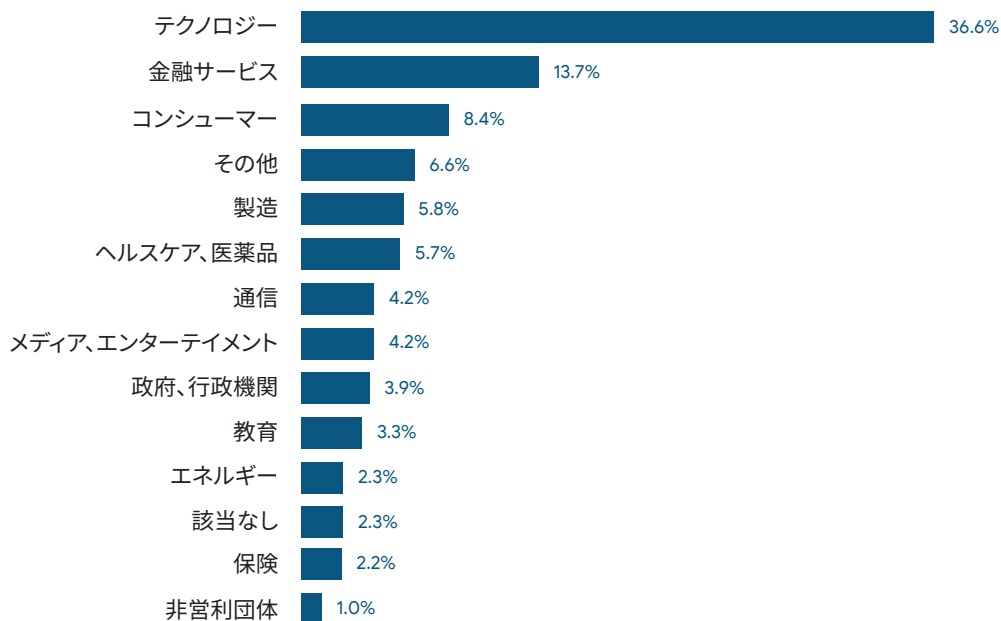
回答者の72%は、開発またはエンジニアリングチーム(30%)、DevOpsまたはSREチーム(18%)、IT運用またはインフラストラクチャチーム(8%)に所属する個人、そして管理者(16%)で構成されています。2022年は、こうした役割の人が回答者の85%を占めていました。回答者に占めるこれら4つの役割の割合が減ったということは、より多くの種類の役割にリーチできたということです。IT運用またはインフラストラクチャチームの割合(8%)は、2022年(19%)のピークを境に2021年(9%)の水準に戻りました。



企業特性

業種

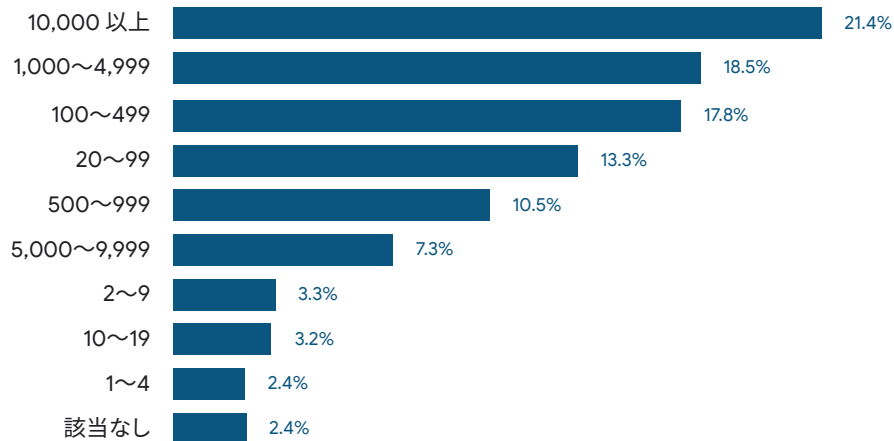
全回答者に占める割合



従業員数

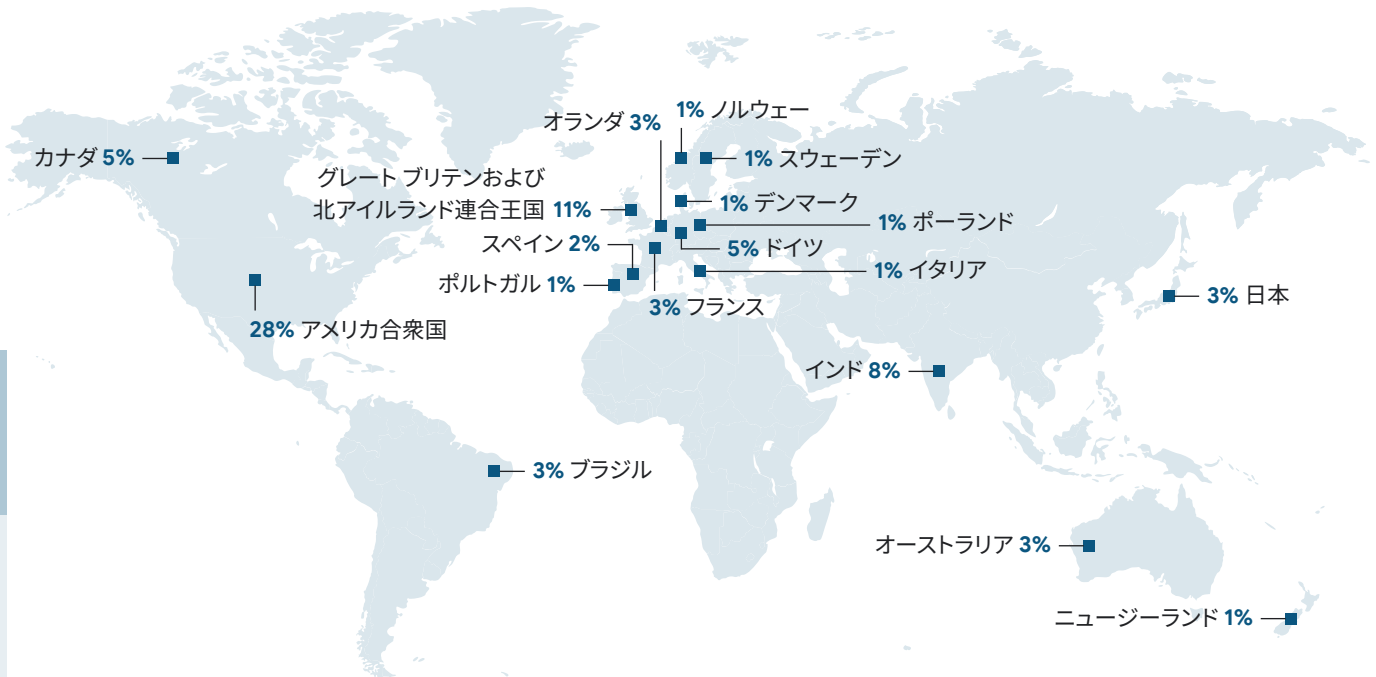
貴社の従業員は何名ですか。

全回答者に占める割合



国

喜ばしいことに、いつも世界中の方々からアンケートにご協力いただいています。皆様、ありがとうございます。



国	国	国	国	国	国
米国	デンマーク	リトアニア	チュニジア	バングラデシュ	グアテマラ
英国	スイス	タイ	ウルグアイ	ドミニカ共和国	ホンジュラス
インド	オーストリア	ハンガリー	アフガニスタン	ガーナ	ラトビア
カナダ	ケニア	イスラエル	アルジェリア	香港(特別行政区)	レバノン
ドイツ	南アフリカ	ベトナム	エジプト	カザフスタン	ルクセンブルク
オーストラリア	アルゼンチン	アラブ首長国連邦	エストニア	ミャンマー	モルディブ
ブラジル	チェコ共和国	ブルガリア	アイスランド	サウジアラビア	マルタ
該当なし	ベルギー	クロアチア	イラン	ソマリア	モーリシャス
オランダ	コロンビア	エクアドル	ナイジェリア	スーダン	モンゴル
日本	フィンランド	インドネシア	ペルー	ウガンダ	モロッコ
フランス	アイルランド	フィリピン	スロバキア	アルバニア	ネパール
スペイン	中国	アルメニア	スロベニア	バハマ	カタール
スウェーデン	ルーマニア	ジョージア	韓国	ベラルーシ	マケドニア旧ユーゴスラビア共和国
イタリア	シンガポール	ギリシャ	スリランカ	ボリビア	
ニュージーランド	メキシコ	マレーシア	アンドラ	カンボジア	トリニダードトバゴ
ポーランド	トルコ	パキスタン	アンゴラ	コスタリカ	
ノルウェー	ウクライナ	ロシア連邦	アンティグア バーブーダ	ジブチ	タンザニア連合共和国
ポルトガル	チリ	セルビア	バーレーン	エルサルバドル	ジンバブエ

勤務形態

雇用形態

回答者の 88% がフルタイム正社員、回答者の 10% が契約社員です。一部の契約社員は、フルタイム正社員とは大きく異なる体験を報告しています。

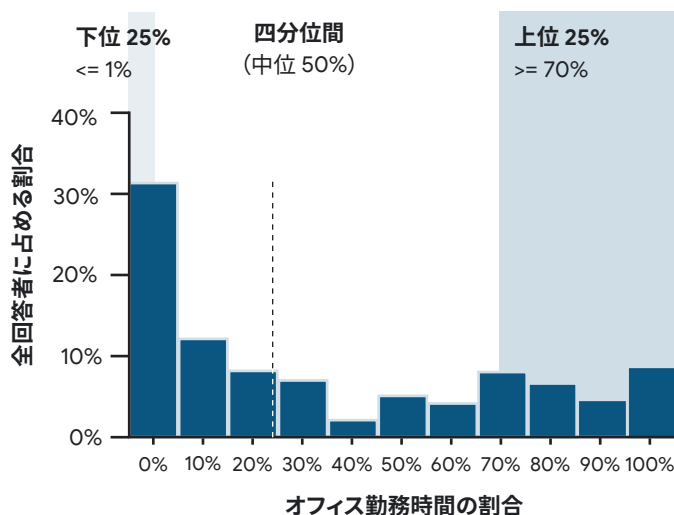
契約	全回答者に占める割合
フルタイム正社員	88%
フルタイム契約社員	8%
パートタイム正社員	2%
パートタイム契約社員	2%

* 担当している主なアプリケーションまたはサービスについて、そのアプリケーションまたはサービスを所有する組織におけるあなたの雇用形態を最もよく表しているものはどれですか。

体験の違いは、チームにおける位置付けに起因する可能性があります。一部の契約社員は、共に働く一員としてチームに組み込まれていると報告しています。つまり、毎日チームメンバーと緊密に連携しており、フルタイム正社員との違いはごくわずかだと考えています。契約社員の回答者の 70% が、自分は一員としてチームに組み込まれているという文に強く同意または同意しています。

勤務場所

今年の回答パターンからは、オフィス回帰の動きがあるにもかかわらず、多くの従業員にとっては依然として在宅勤務が現実であるということがわかります。回答者の約 33% はほぼ在宅勤務（オフィス勤務時間が 5% 未満）で、回答者の 63% は在宅勤務の方がオフィス勤務より多くなっています。残りの回答者については、ハイブリッド勤務が最も一般的な勤務形態なのかもしれません。これは、オフィス勤務時間が 70% 未満の回答者が 75% を占めることからわかります。オフィスに強い愛着を持つ人はあまり多くありません。オフィス勤務時間が 95% を超える回答者はわずか 9% でした。



モデル

はじめに

これまで、DORA は1つの巨大なモデルを作成していました。今年は以下の理由から、複数のモデルに分割することにしました。

- 巨大なモデルはすぐに扱いにくくなります。変数を追加するたびにモデルの機能が変化します。そのため推定が不正確になり、変化の理由を突き止めることが困難になります。
- 今年はセクションごとに仮説を立てました。そのため、セクションごとにモデルを作成する方が理にかなっています。
- Y に対する X の影響を推定するにあたり、巨大なモデルの利点は明らかではありません。Y に対する X の影響を把握するために、有向非巡回グラフを使用して、モデルに含めるべき共変量と含めるべきでない共変量を確認しました。
- 今年扱った仮説の数では、読者が巨大なモデルの意味を理解することが非常に難しくなります。以下の可視化のための図をすべて1つの図にまとめるようなものです。



図の読み方

読み方を知れば、これらの図が多くの情報を効率的に伝えられるツールであることがわかるでしょう。



変数

変数とは、測定しようとしたコンセプトのことです(ドキュメントの質など)。

変数カテゴリ

- 変数
- 変数
- 変数
- 変数

変数カテゴリは、単にカテゴリとして考えていることを示すものであって、分析とは関係ありません。高次の複合概念であるかどうかを統計的に評価したということです。

+



正の影響。単に増えるという意味であって、良いという意味ではありません。

-



負の影響。単に減るという意味であって、悪いという意味ではありません。



データで立証されなかった仮説上の影響。



明示的に分析したメディエーション経路の一部。

モデルは一般的な総括であることに注意する

読みやすくするために、いくつかの変数を一緒に分類しています。この分類戦略では、変数カテゴリに向かう矢印、変数カテゴリから出る矢印、またはその両方が一般的な結果パターンになりますが、カテゴリ内のすべての変数に当てはまるわけではありません。たとえば、知識の共有は主な成果のほとんどに好影響を与えます。そのため、知識の共有から主な成果の変数カテゴリに向けて、プラス記号(+)付きの矢印を引くこととなります。ただし、知識の共有はソフトウェアデリバリーパフォーマンスには好影響を与えません。詳細については、関連する章をご覧ください。

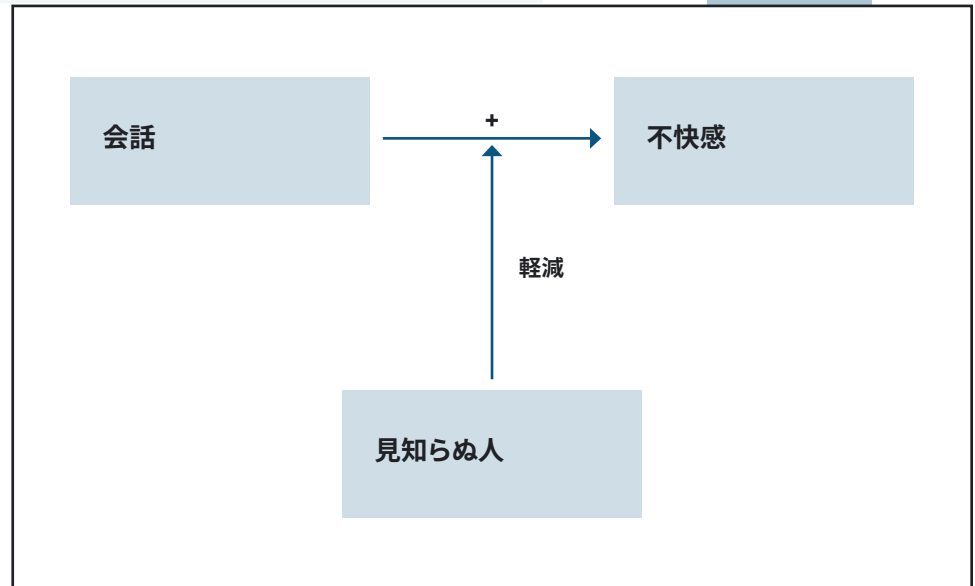
モデレーションの例

モデレーションは理解が難しい統計情報のコンセプトですが、実際の状況においては、「場合による」というような意味になります。本レポートの文脈におけるモデレーションのコンセプトを明らかにするために、簡単な例を紹介します。

『ラリーのミッドライフ☆クライシス』シーズン3で、ラリーデヴィッドが「知り合いとは話したくないが、見知らぬ人となら問題ない」ということを言っています。これを基にして、簡単な図で説明します。

「知り合いとは話したくないが、見知らぬ人となら問題ない。」

ラリー デヴィッド



この図は、ラリーにとって会話が不快感に正の影響を与えていることを示しています。ここでいう「正」とは、単に増えるという意味であって、必ずしも良いこととは限りません。これが「会話」と「不快感」を結ぶ実線と、「不快感」を指す矢印の向きで表されています。この矢印で、「会話」から「不快感」に向かう因果の流れがあると考えていることを示しています。見たところ、会話はラリーに不快感を起こさせる傾向があります。

次に注意すべき点は、「見知らぬ人」(ここでは、見知らぬ人: はい / いいえ、のブール値を表します)は、別の変数を指していないということです。代わりに、影響の矢印を指しています。つまり、「見知らぬ人」が変更するのは変数ではなく影

響だと考えているわけです。そのため、矢印を「見知らぬ人」から別の変数ではなく、別の矢印に向けて引いているのです。ラリーが見知らぬ人と話しているかどうかは、不快感に対する会話の影響を左右する、ということです。言い換えると、ラリーの会話相手が見知らぬ人かどうかによって、不快感に対する会話の影響が変わる、ということです。相手が見知らぬ人の場合、会話の影響はラリーにとって「問題ない」ものです。見知らぬ人はラリーが会話中に感じる不快感を軽減してくれる、と言えるかもしれません。

その他のモデレーションの種類

- **増幅** - 正の影響の正を強め、負の影響の負を強めます。
- **弱化** - 影響を弱めます。
- **軽減** - 正の影響の正を弱め、負の影響の負を弱めます。
- **反転** - 正の影響を負にし、負の影響を正にします。
- **変更** - 影響が変化したものの、そのパターンを一言ではうまく要約できないことがあります。これは、原因として機能するカテゴリ変数でよく発生します。たとえば、ある業界が、さまざまな条件下（景気変動など）で異なる反応を示すことがあります。

メディエーションの例

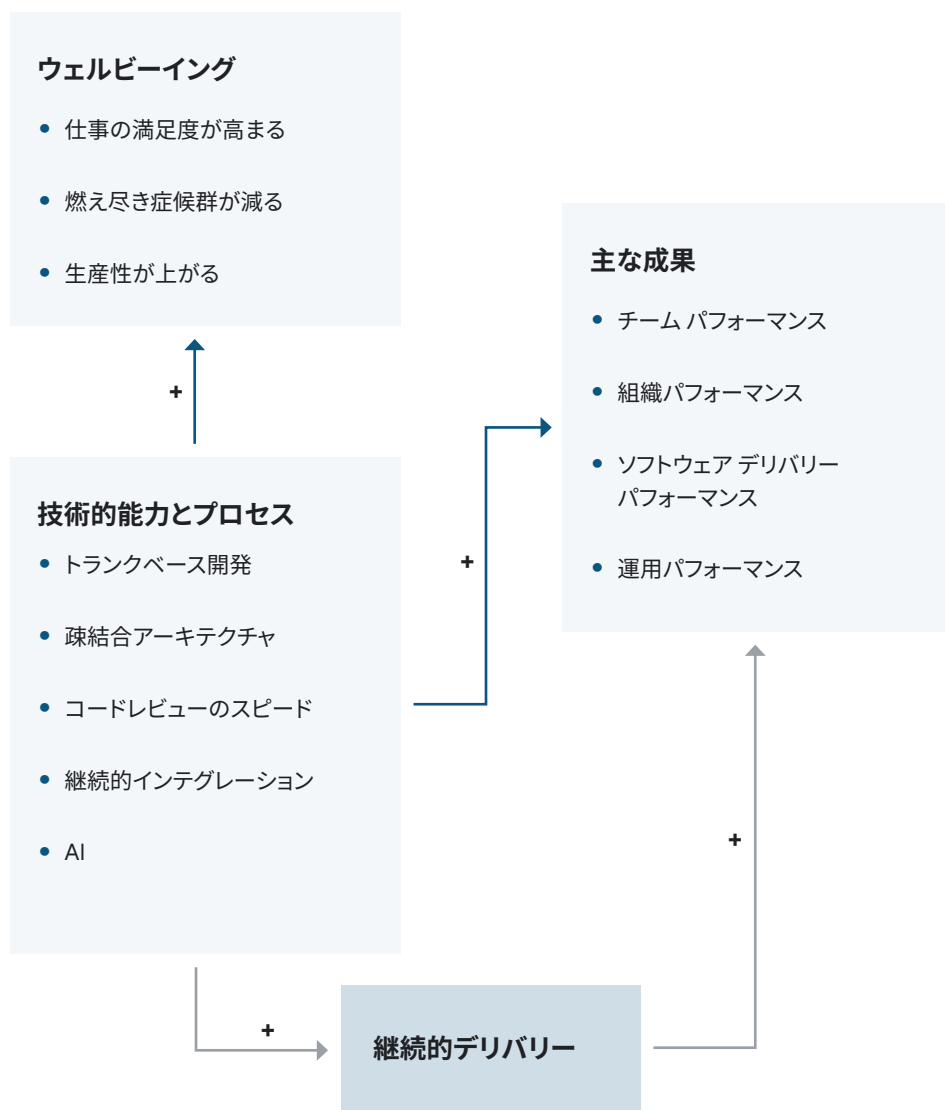
モデレーションと同様に、メディエーションについてもよく考えます。メディエーションの根底にあるのは、「なぜ」、「どのように」です。太陽によって植物の丈が伸びるのはなぜでしょうか。イチゴを食べると甘く感じるのはなぜでしょうか。アクセルを踏むと車が動くのはなぜでしょうか。鎮痛剤はどのようにして不快感を軽減するのでしょうか。統計学ではメディエーションのテストを行えます¹。これは「Y に対する X の影響は、M で完全にまたは部分的に説明がつくようだ」と言えることを指します。たとえば、植物の丈に対する太陽の影響は、光合成で説明がつかます。



¹<https://lavaan.ugent.be/tutorial/mediation.html>

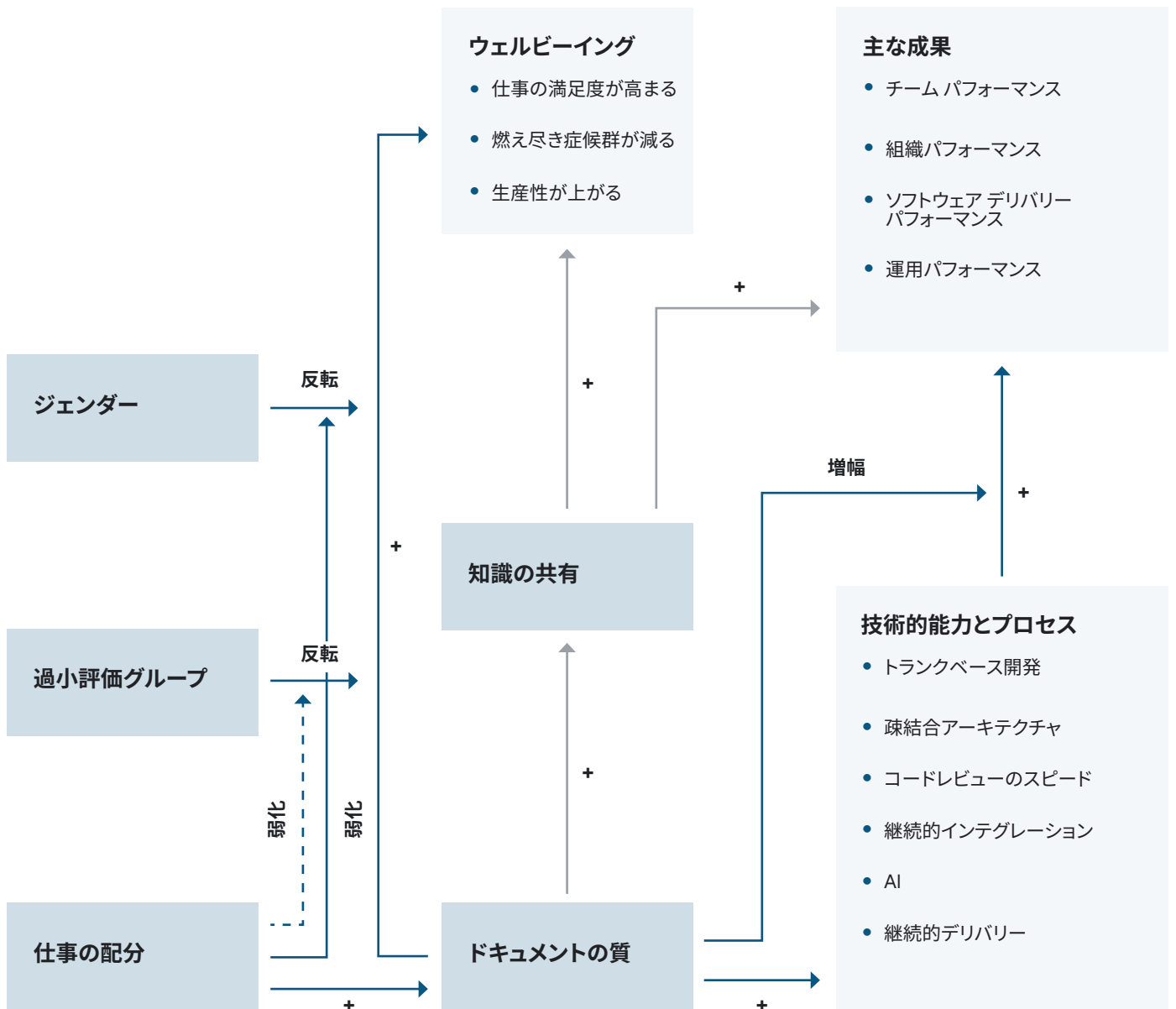
3章「技術的能力からパフォーマンスを予測する」のモデル

技術的能力とプロセスは、ウェルビーイングと主な成果に好影響を与えます。主な成果に対する技術的能力の影響には、継続的デリバリーにより部分的なメディエーションがあります。つまり技術的能力は、主な成果に対してダウンストリームの影響を与える継続的デリバリーの環境を作り出します。



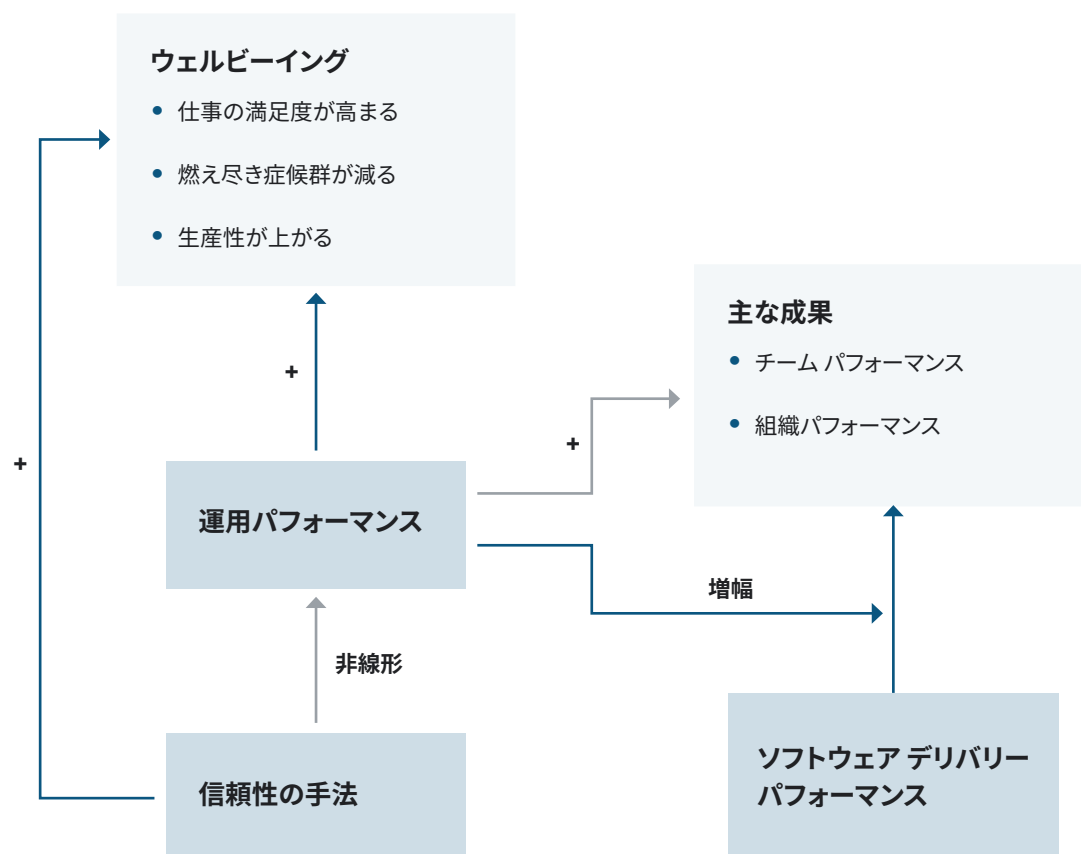
4章「ドキュメントが基礎となる」のモデル

簡潔に言うと、質の高いドキュメントには、ウェルビーイングと主な成果に対する正の影響があります。これは部分的には、知識の共有が可能な環境が生まれるためです。質の高いドキュメントは、チームが技術的能力とプロセスを確立するためにも役立ちます。そのうえ、技術的能力とプロセスが主な成果にさらに強い影響を与えるためにも役立ちます。最後に、ドキュメントの質が全員のウェルビーイングの向上につながるわけではありません。この複雑な調査結果の内訳については、該当するセクションをご覧ください。



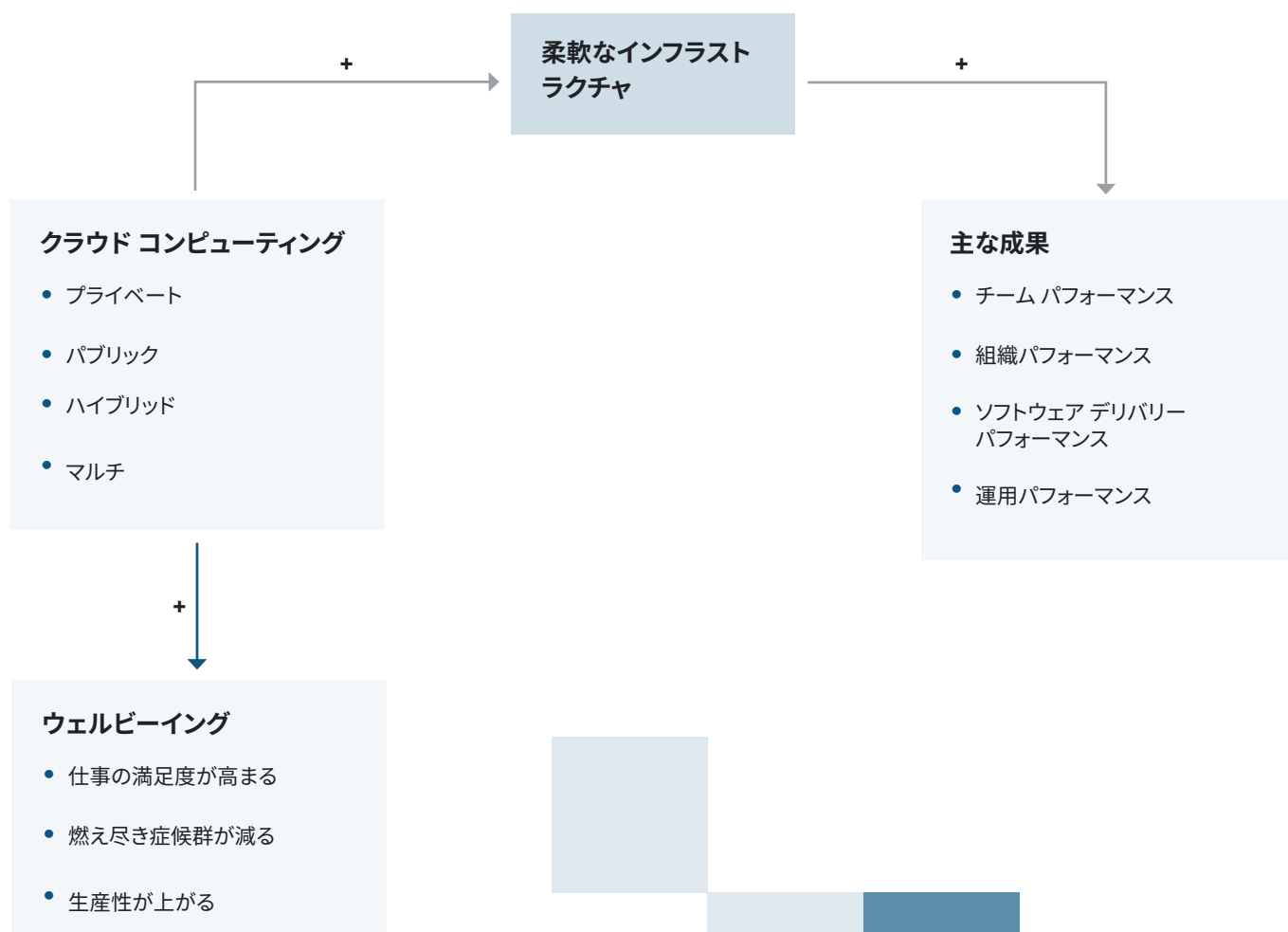
5 章「信頼性がパフォーマンスを高める」のモデル

ここでは、運用パフォーマンスの中心的な役割を探ります。運用パフォーマンスは、ウェルビーイングと主な成果を向上させるだけでなく、ソフトウェア デリバリー パフォーマンスの影響も増幅します。信頼性の手法は運用パフォーマンスと非線形の関係にあります。詳細については 5 章をご覧ください。



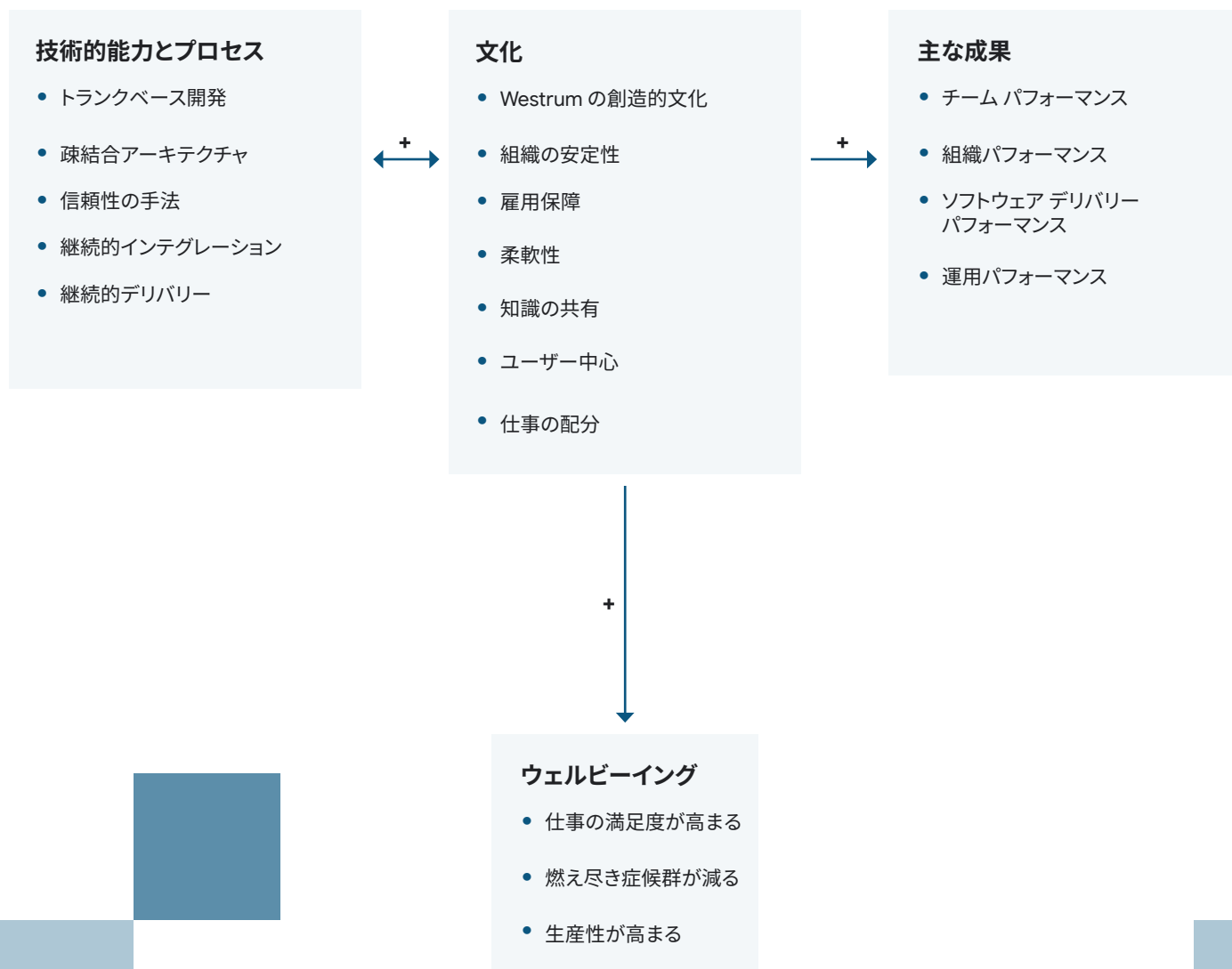
6章「柔軟なインフラストラクチャが成功の鍵」のモデル

クラウドコンピューティングは、インフラストラクチャの柔軟性を高めるため、主な成果に影響を与えます。また、クラウドコンピューティングはウェルビーイングの向上にもつながります。



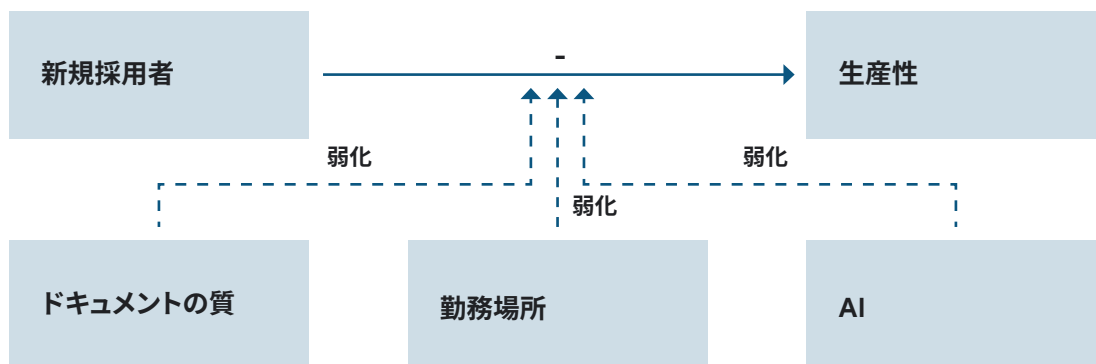
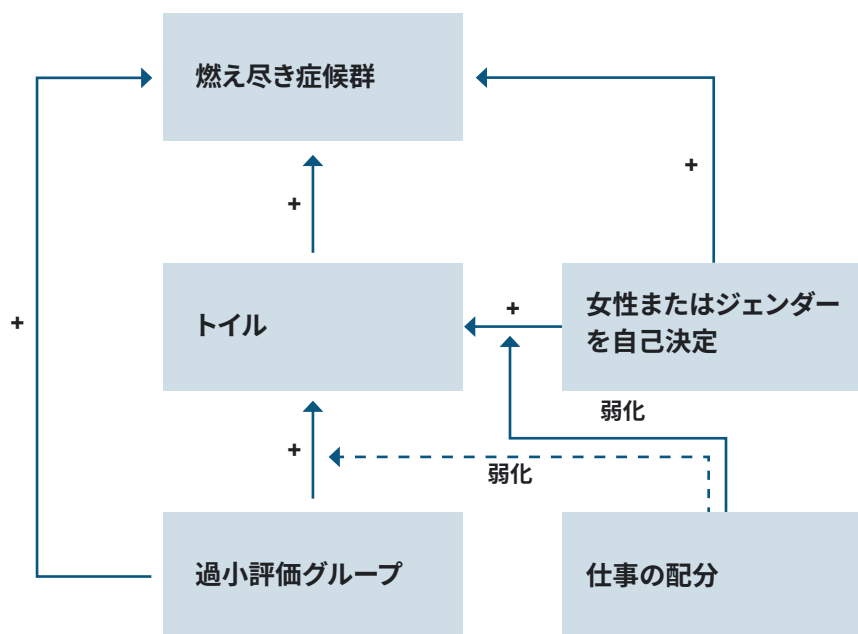
7章「組織文化への投資なしには何事もうまくいかない」のモデル

この図から、組織文化がさまざまなものの中心にあることがわかります。組織文化は技術的能力、主な成果、ウェルビーイングと正の関係にあることがわかりました。



8章「個人の属性が重要になる状況と理由」のモデル

このセクションのモデルは2つあります。一方は、過小評価グループを自認する人と男性を自認しない人の燃え尽き症候群が、なぜ、いつ、高水準になるのかを探ります。もう一方は、ドキュメントの質、勤務場所、AIが、新規採用者の生産性を高めるために役立つかどうかを探ります。



関連情報

DORA コミュニティに参加して、ソフトウェア デリバリーと運用パフォーマンスの向上について話し合い、学び、協力する [DORA コミュニティ](#)

DORA DevOps クイックチェックを行う
<https://dora.dev/quickcheck>

ソフトウェア デリバリー パフォーマンスと組織パフォーマンスを改善する技術、プロセス、組織文化の能力を調べる
<https://dora.dev/devops-capabilities/>

SRE のリソースを探す
<https://sre.google>
<https://goo.gle/enterprise-roadmap-sre>

書籍を読む: 『Accelerate: The science behind devops: Building and scaling high performing technology organizations』IT Revolution
<https://itrevolution.com/product/accelerate/>

SPACE フレームワークを使用して、チームに適した指標群を探す “The SPACE of Developer Productivity: There’s more to it than you think”
<https://queue.acm.org/detail.cfm?id=3454124>

最新のコードレビューに関する研究がいくつかあります。レポートをいくつか紹介します。

- “Expectations, Outcomes, and Challenges of Modern Code Review”
<https://dl.acm.org/doi/10.5555/2486788.2486882>

- “Code Reviews - From Bottleneck to Superpower”
<https://learning.acm.org/techtalks/codereviews>
- “Modern Code Review: A Case Study at Google”
<https://dl.acm.org/doi/10.1145/3183519.3183525>
- “The Choice of Code Review Process: A Survey on the State of the Practice” https://link.springer.com/chapter/10.1007/978-3-319-69926-4_9
- “Investigating the effectiveness of peer code review in distributed software development based on objective and subjective data”
<https://jserd.springeropen.com/articles/10.1186/s40411-018-0058-0>

書籍を読む: 『The No Club: Putting a Stop to Women’s Dead-End Work』Simon & Schuster
<https://www.simonandschuster.com/books/The-No-Club/Linda-Babcock/9781982152338>

DORA の研究プログラムの出版物 (過去の Accelerate State of DevOps レポートを含む) <https://dora.dev/publications/>

研究とレポートに関するよくある質問 <http://dora.dev/faq>

正誤表 - 本レポートの変更、訂正、説明の閲覧と送信
<https://dora.dev/publications/errata>

付録

ソフトウェア デリバリー パフォーマンスの測定方法の改良

今年は、変更時の障害と、障害からの復旧に関する評価方法を変更しました。

回答者が**変更時の障害率**を報告する方法を変更しました。これまでは、回答者に6つの選択肢(0~15%、16~30%など)を提示していました。今年は回答者にスライダーを提示し、0%から100%の間で任意の値を選択できるようにしました。

この変更を行った理由は2つあります。

- 変更時の障害率は常に、他に3つあるソフトウェア デリバリー パフォーマンスの尺度とは少し異なる挙動を示していました¹。そこで、バケットのサイズにその一因があるのではないかと考えました。答えの精度を上げることで統計的なパフォーマンスが向上する可能性があります。この理論は検証できました。
- 10年ほど前に研究が始まったときより現在の方が、自らの変更時の障害率についてチームがよく理解しているという仮説を立てました。この仮説を検証する方法はまだ見つかっていません。

障害からの復旧について、今年は2種類の方法で尋ねました。

- これまではこのように質問していました。「主なアプリケーションまたはサービスで、サービスのインシデントや、ユーザーに影響を与える障害(計画外のサービス停止やサービス障害など)が発生した場合、サービスの復旧に通常どれくらいの時間がかかりますか。」
- 今年はこのように修飾語句を加えました(ここでは太字にしていますが、実際のアンケートでは太字にしていません)。「主なアプリケーションまたはサービスで、**本番環境に変更を加えたかユーザーに変更をリリースした結果、サービスの低下(サービス障害、サービスの停止など)が発生して、対策(ホットフィックス、ロールバック、フィックス フォワード、パッチなど)が必要になった後、サービスの復旧に通常どれくらいの時間がかかりますか。**」

¹ Forsgren, N., Humble, J., and Kim, G. Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations (IT Revolution Press, 2018), 37–38.

復旧にかかる時間について尋ねるこれまでの方法では、ソフトウェアの変更によって障害が発生した場合と、地震でデータセンターのサービスが中断するなどして障害が発生した場合を区別できませんでした。そこで、より正確な表現を使用することで同様の種類の障害を相互に比較でき、他に3つあるソフトウェア デリバリー パフォーマンスの尺度に統計学的に沿ったものになるという仮説を立てました。

現在は「デプロイ失敗時の復旧までの時間」という用語を使用していますが、これは過去に使用していた、「MTTR」と略すこともある一般的な「復旧時間」と区別するためです。MTTRはコミュニティに「Mは平均値なのか、中央値なのか」などの混乱を引き起こしました。さらに、レジリエンス エンジニアリング分野の実務者など、障害から多くのことを学ぼうとする実務者は、学びと改善に役立つ確実な尺度としては MTTR を使わなくなっています²。

新しく追加された質問と新しい指標「デプロイ失敗時の復旧までの時間」の方が、ソフトウェア デリバリー パフォーマンスを測定するという趣旨に沿っています。

² “Moving Past Simple Incident Metrics: Courtney Nash on the VOID”
<https://www.infoq.com/articles/incident-metrics-void/>



比較を支える数学

本レポート全体を通じて、「ある変数の測定値が大きいほど、ある要素のスコアがN倍大きくなる」ということが統計データによって示されています。これは何でしょうか。何と比較しているのでしょうか。ここでは、式を立てる流れを紹介します。

1. バイアスが生じ得る経路を考慮した回帰手法を使って評価されたモデルから、式を立てます。

$$\text{happiness} \sim 5.64 + 0.19 \times \text{sunshine} + 0.14 \times \text{temperature}$$

2. 目的の変数、たとえば sunshine (日照量) について、比較するために high (高い) 値と low (低い) 値を求めます。

$$\text{mean} = 6.3, \text{sd} = 1.4$$

$$\text{high sunshine} = \text{平均より 1 sd 大きい} = \text{mean} + \text{sd} = 6.3 + 1.4 = 7.7$$

$$\text{low sunshine} = \text{平均より 1 sd 小さい} = \text{mean} - \text{sd} = 6.3 - 1.4 = 4.9$$

3. 共変量の平均値を計算します。この例では temperature (温度) であり、0.24 と求めることができます (標準化された温度)。
4. ステップ 1 の式を、high sunshine と low sunshine それぞれに完成させます。式の中で異なる数字は 1 つだけだということにお気づきでしょうか。このようにして他のすべてを等しく保ち、目的とする 1 つの違いだけを分離します。

$$\text{high sunshine happiness 予測値} = 5.64 + 0.19 \times 7.7 + 0.14 \times 0.24 = 7.1$$

$$\text{low sunshine happiness 予測値} = 5.64 + 0.19 \times 4.9 + 0.14 \times 0.24 = 6.6$$

5. 比率を計算します。

$$\frac{\text{high sunshine happiness 予測値}}{\text{low sunshine happiness 予測値}} = \frac{7.1}{6.6} = 1.1 \text{ 倍}$$

6. この比率は、日照量が多い場合は少ない場合に比べて幸福 (happiness) の程度が 10% 高くなることを示しています。



シミュレーションとは

データは架空のものではありません。ベイズ統計学を使用して**事後分布**を計算し、「異なるパラメータ値が現れる頻度の予測値」³を捉えます。「シミュレーション」部分は、この事後分布を1,000回超利用し、データを基にしたとき最も信頼できるパラメータ(平均、ベータ重み、シグマ、切片など)の値を調べます。「事後分布はパラメータ値(0.1、0.7、0.5、1などの数字)でいっぱい**の**バケットのようなものです。バケットの中で、各値はその事後確率に比例して存在します。ピーク付近の値はテールの値よりもはるかに一般的です⁴。」

シミュレーションでデータの解釈の候補を調べ、不確実性の程度を把握するという事です。各シミュレーションについては、データといくつかのルールしか知らない小さなAIが、情報に基づいた推測で空白(パラメータ)を埋めようとしているのだと考えることができます。これを4,000回繰り返

すと、あるパラメータに対する小さなAIの推測が4,000件得られます。こうした推測から学べることはたくさんあります。平均的な推測についてや、推測の89%が該当する値区間、あるレベルを超える推測の件数、推測のばらつきの程度などを知ることができます。さまざまなモデルの推測(シミュレーション)を組み合わせてみてもよいでしょう。

線がたくさんあるグラフや、可能性のある値の分布のグラフでは、データに関して最も妥当だと思われることや、不確実性の程度を示そうとしています。

³ McElreath, Richard. *Statistical rethinking: A Bayesian course with examples in R and Stan*. Chapman and Hall/CRC, 2018, pg.50

⁴ McElreath, Richard. *Statistical rethinking: A Bayesian course with examples in R and Stan*. Chapman and Hall/CRC, 2018, pg.52



Google LLC の「Accelerate State of DevOps 2023」は、CC BY-NC-SA 4.0 の下での使用が許可されています。