

DX Criteria ver.201912

「2つのDX」とデジタル時代の経営ガイドライン



1. DX Criteria の目指すところ
2. DX Criteria の構造
3. DX Criteria の使い方
4. 詳細項目一覧
5. 今後の流れ

Appendix



(株)レクター



(一社)日本CTO協会

DX Criteria v201912 は 株式会社レクターのDX組織診断の一部を
日本CTO協会に寄贈し、リバイズしたものです。

2. DX Criteria の目指すところ

VISION AND MESSAGE

デジタル時代の超高速な仮説検証能力 を得るには「2つのDX」が必要不可欠



日本CTO協会では「DX」という言葉を2つの意味で捉えています。

一つは、企業がどれだけ経営に対してデジタル技術を用いたビジネス変革ができているかを表す**企業のデジタル化 (Digital Transformation)**です。

もう一つは先端開発者にとっての働きやすい環境と高速な開発を実現するための文化・組織・システムが実現されているかを意味する**開発者体験 (Developer eXperience)**です。

これらの2つは、経営にとって**ヒト・モノ・カネ**が一体であるように、一体で実現されるものです。デジタル技術は目に見えないため、しばしばわかりやすいものにだけ投資して見えない品質をおろそかにしてしまいます。そのため、開発者体験は悪化し、企業のデジタル化を阻害してしまうことがあるのです。

私たちは、「2つのDX」を一体で捉えた基準をつくり、その普及をしていきたいと考えています。



DX Criteriaの目的 = 超高速な事業仮説検証能力を得ること

Point 1

組織文化と「見えない」投資

高速な開発を行う組織には一度体験しないと価値がわかりにくい投資や習慣があります。この説明コストの高さを軽減し、導入を促します。

Point 2

タスク型ダイバーシティ

事業価値あるサービスが実現するためには様々なデジタル人材と既存事業人材の相互理解と共創関係が必要で、この進展を促します。

Point 3

メリハリのあるIT戦略

標準化・コモディティ化した領域については外部サービスを利用し、競争領域に特化して内製化をすすめるためのメリハリのある投資を促します。

Point 4

組織学習とアンラーニング

新しいツールや潮流に挑戦するための組織学習と、時代が変わってしまった習慣のアンラーニング(学びほぐし)を促します。

Point 5

自己診断と市場比較

関連するレポートと自己診断によって競合状況との差を認識しやすくし、自社の強み弱みを理解して段階的に変化できるように促します。



Point 1

組織文化と「見えない」投資



システム品質への投資

- 自動的テスト
- 安全なリリース自動化
- 疎結合なアーキテクチャ
- システムモニタリング

etc,...



組織風土・チーム文化

- 心理的安全性
- システムと事業への帰属意識
- リソース効率からフロー効率
- 経験主義とふりかえり習慣

etc,...



実際に体験してメリットを享受しないとわかりにくいいため、重視されない。



Point 1

組織文化と「見えない」投資

cont'd

システム品質への投資

組織風土・チーム文化

改善速度

品質投資の十分なシステム

×

良い開発者文化のあるチーム

=

高速な開発サイクル

品質投資の十分なシステム

×

良い開発者文化のないチーム

=

事業スピードに合わない

品質投資が不十分なシステム

×

良い開発者文化のあるチーム

=

事業スピードに合わない

品質投資が不十分なシステム

×

良い開発者文化のないチーム

=

修正がほとんどできない

事業競争力

技術的負債

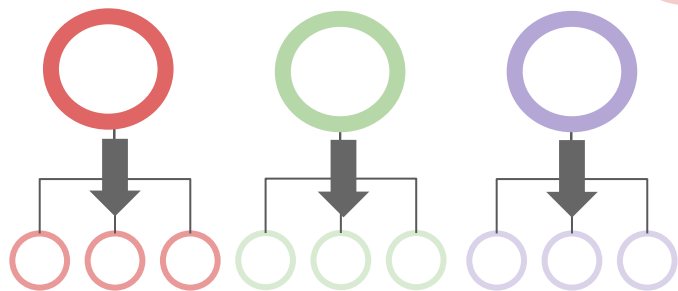


Point 2

タスク型ダイバーシティ*

タスク型ダイバーシティ

低

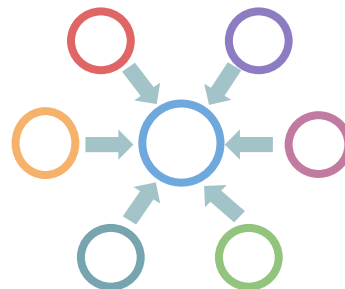


同じ職種のメンバーだけで構成されたチーム

- 組織規律が重要。
- 同質性が高いため、マネジメントが比較的容易。
- 専門性を高めることがしやすい。
- 計画を立てやすく、意思決定サイクルが長い。

タスク型ダイバーシティ

高



複数の専門職で1つことを行うチーム

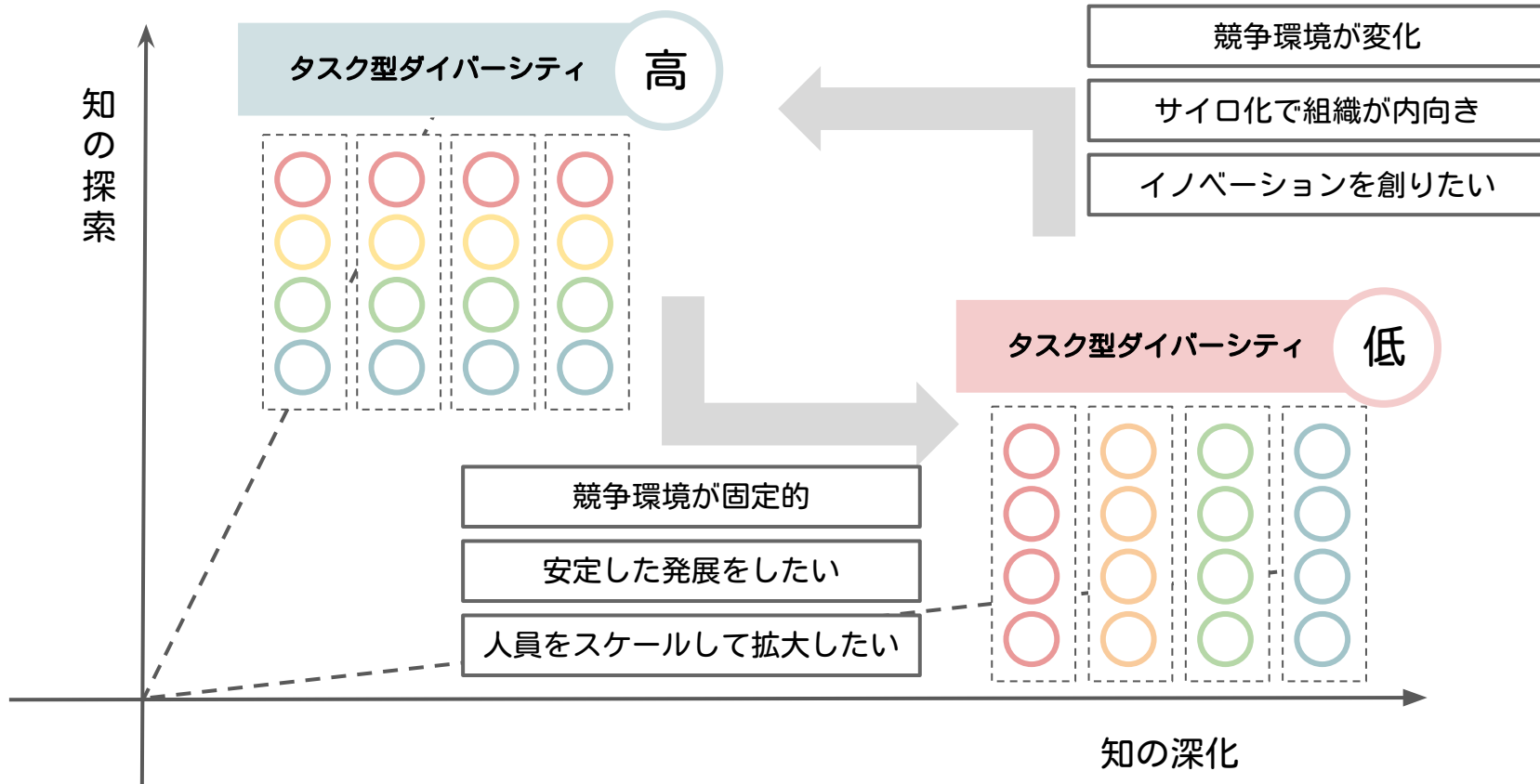
- ビジョンマネジメントが重要。
- 相互理解とコミュニケーションを重視する。
- イノベーションが起こりやすい。
- 権限の委譲を明確にして意思決定を早める。



Point 2

タスク型ダイバーシティ

cont'd





Point 3

メリハリのあるIT投資

守りのIT投資

コモディティ領域

コスト削減や管理を目的としたもの

社会課題としてすでに解決策があるもの

XaaS利用および外部ベンダーとの協調

攻めのIT投資

競争領域

売上や利益の増大を生み出すもの

自社事業の競争優位につながるもの

内製化またはラボ型開発の推進





Point 3

メリハリのあるIT投資

cont'd

自社の強みとして、
改善する価値がある

攻めのIT投資

競争領域



他社/社会と同じ課題を
解決するもの

守りのIT投資

コモディティ

内製化チーム

特定のKPIや事業目標に向かって、内部チームで継続的に開発をすすめるスタイル。事業理解とシステムノウハウが蓄積する。

内部プロジェクト

内製エンジニアや一部フリーランサーなどと協力しながら、一時的なプロジェクトとしてシステム開発を行う。

ラボ型開発

テックリードとプロダクトデザイナーを外部の協力を仰ぎながら、開発と同時にノウハウも伝承できるようにした開発スタイル。

外注型開発・共同開発

XaaSと業務のつなぎ込み部分を交換しやすい形で外注したり、業界コンソーシアムで共通化して、開発を行って効率化を図る。

XaaS/パッケージ利用

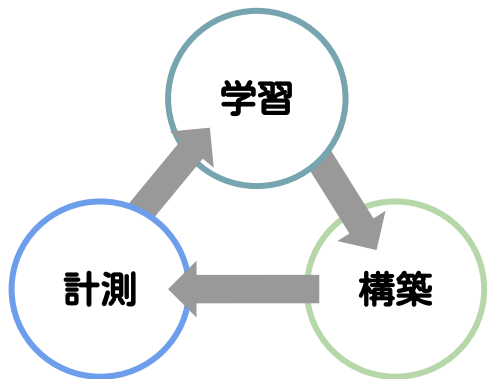
最新の動向をチェックした上でうまく選定していく。メンテナンスや追加開発をしつづける費用よりも専門SaaS事業者のほうがリーズナブルなことが多い。



Point 4

組織学習とアンラーニング

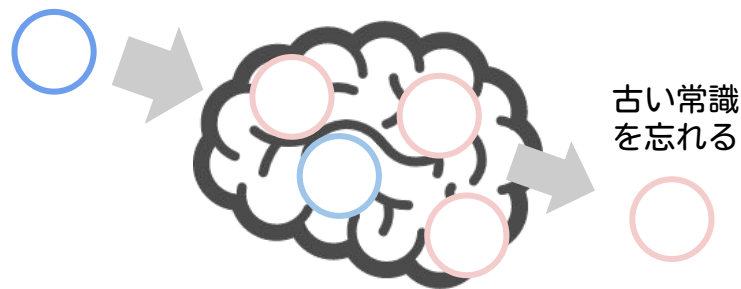
組織学習（計測とふりかえり）



組織として評価のために数字を追うのではなく、事実に基づいた計測から洞察を得て学び、そして改善していくというサイクルが重要です。この点を確認していきます。

アンラーニング（学びほぐし）

新しい当たり前



アンラーニング(学びほぐし)とは、成功体験のある古い常識を忘れて、新しい当たり前を取り入れやすくすることです。時代の変化に適應できる組織にはアンラーニングの習慣が宿っています。



Point 5

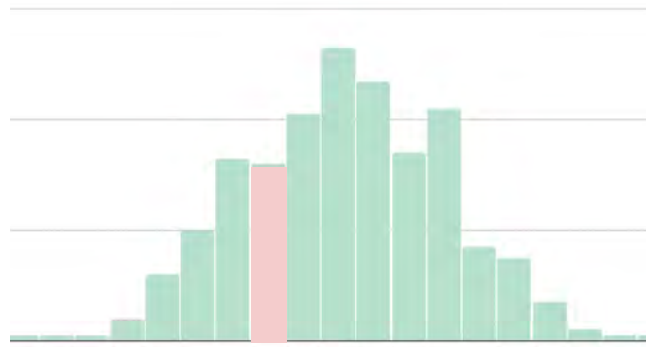
自己診断と市場比較

自己診断(アセスメント)

	1	2	3	4	5	6	7	8	9	10	11
バージョン管理	0.4	0.3	0.8	0.8	0.2	0.4	0.9	0.8	0.5	0.2	0.5
ソースコードの明確さ	0.5	0.1	0.0	0.4	0.5	0.5	0.1	0.9	0.4	0.6	0.2
継続的インテグレーション	0.3	0.7	0.9	0.9	0.2	0.3	0.2	0.1	0.2	0.6	0.7
継続的デプロイ	0.8	0.5	0.9	0.1	0.0	0.9	0.9	0.5	0.8	0.1	0.8
API駆動開発	0.9	0.0	0.6	0.4	0.9	0.3	0.6	0.7	0.6	0.3	0.7
疎結合アーキテクチャ	0.8	0.4	0.6	0.4	0.1	0.2	0.5	0.4	0.4	0.3	0.8
システムモニタリング	0.5	0.9	0.2	1.0	0.4	0.6	0.9	0.8	0.8	0.1	0.5
セキュリティシフトレフト	0.1	0.8	0.2	0.5	0.2	0.8	0.2	0.2	0.9	0.9	1.0

さまざまな項目について、自己診断を行えるような個別具体的なプラクティスを列挙しています。そのことで、あえて取り入れていないのか、検討が進んでいないのかなどより現場感のある議論を促進することが目的です。

市場比較



今後、計測ツールなどを公開していき、自社がどのような状況にあるのかを全体と比較できるようにしていく方針です。ご協力いただける各社の統計データと比較して、今自分たちがどのレベルにいるのかを知れるようにしていく予定です。



Point 5

自己診断と市場比較

なぜ、変える必要があるのか？

DXの実現のためには、今までの常識であったことから「新しい当たり前」を取り入れていく必要があります。

しかし、その外部の基準が存在しないことによって、「なぜそれを取り入れるのか/やってみるのか」の説明をゼロから求められ、結果的に現場は学習性無気力状態になってしまい変革が遅れてしまいます。

なぜ、変えていかないのか？

本基準で、リストアップされている項目は必ずしもすべてを実現する必要はありません。

しかし、それをやらない場合には、自分たちはなぜやらないのかを説明できる必要があるでしょう。

このように基準を使いながら、「変える理由」から「変えない理由」を確認するような文化形成を育むきっかけになればと考えています。

基準を通じて、「説明責任の向き」を反転させていきたい。

2. DX Criteriaの構造

5つのテーマと高速仮説検証ループ



チーム

システムに関わるチームがどれだけ生産的に高速な仮説検証や開発を行うことができる状態にあるかをチェックする。



システム

システム自体がレガシー化されずにどれだけ安全かつ高速に改善できる状態にあるかをチェックする。



データ駆動

社内外のデータがどれだけ活用しやすい状態にあるか、また経営や意思決定に活用されているかをチェックする。



デザイン思考

デザインとUXから事業価値を生み出すために必要な仮説設定能力や習慣、効率的に行うための組織についてチェックする。



コーポレート

経営やミドルオフィス・バックオフィス機能がどれだけデジタル戦略を意識した活動ができているかをチェックする。



DX Criteriaの構造と観点

テーマ (5個)



カテゴリー (各8個)

1	チーム構成と権限委譲
2	チームビルディング
3	心理的安全性
4	タスクマネジメント
5	透明性ある目標管理
6	経験主義的な見積りと計画
7	ふりかえり習慣
8	バリューストリーム最適化



チェックリスト (各8項目)

メトリクスの計測	チームの人数は、5人以上12人以下か。(ヒザ2枚ルール)	はい / いいえ
学習と改善	チームの仕事にある特定の人物に属人化した仕事を洗い出し、減らしていく仕組み・習慣があるか。	はい / いいえ
プラクティスと習慣	チームの権限について、RACI図やデリゲーションボーカールなどによって、可視化され共有されているか。	はい / いいえ
アンチパターン	チームは価値提供をするに必要な全職能のメンバーで構成されているか。(フィーチャーチーム)	はい / いいえ
	チームおよびチームリーダーは、チームのミッションのために必要な外部のリソースを調達するための予算や権限を持っているか。	はい / いいえ
アンチパターン	チームは存在するが、それぞれのやっている仕事の内容をよく知らないし、代わりにやることもできない。	はい / いいえ
	チームリーダーが複数のチームやプロジェクトを兼務しており、自チームのためにすべての時間を使うことができない。	はい / いいえ
アンチパターン	チームの仕事にある特定の人物に属人化した仕事を洗い出し、減らしていく仕組み・習慣があるか。	はい / いいえ

合計**320**個の観点から、企業のDXの進捗度を自己診断。強みと弱みを分析し、次の一手に。



評価項目の構造

メトリクスの計測

DXの進んだ組織においては、従来型のパフォーマンス指標とは異なる指標が計測・管理されることがある。それらを計測/管理しているかを問う。

学習と改善

失敗も含め、組織学習を健全に回すためのサイクルが回っていると新しい挑戦や改善が生まれやすい。そのため、そのサイクルの存在を確認する。

プラクティスと習慣 (各3)

デジタル化の進んだ企業群では当たり前に行われる習慣や実践手法が行われているかを確認する。経営数値に見えにくい文化レベルの成熟をとらえるためにチェックする。

アンチパターン (各3)

デジタル化が進む過程で減っていく慣習的行動をチェックする。逆指標として用いる。古い常識によって生まれていることであり、組織的なアンラーニングが行われているかを確認する。

5テーマ x 各8カテゴリ x 8項目

チーム	システム	データ駆動	デザイン思考	コーポレート
チーム構成と権限委譲	バージョン管理	顧客接点のデジタル化	ペルソナの設定	スパン・オブ・コントロール
チームビルディング	ソースコードの明確さ	事業活動データの収集	顧客体験	開発者環境投資
心理的安全性	継続的インテグレーション	データ蓄積・分析基盤	ユーザーインタビュー	コミュニケーションツール
タスクマネジメント	継続的デプロイ	データ処理パイプライン	デザインシステムの管理	人事制度・育成戦略
透明性ある目標管理	API駆動開発	データ可視化とリテラシー	デザイン組織	デジタル人材採用戦略
経験主義的な見積りと計画	疎結合アーキテクチャ	機械学習プロジェクト管理	プロトタイピング	モダンなITサービスの活用
ふりかえり習慣	システムモニタリング	マーケティング自動化	ユーザビリティテスト	経営のデジタルファースト
バリューストリーム最適化	セキュリティシフトレフト	自動的な意思決定	プロダクトマネジメント	攻めのセキュリティ

3. DX Criteriaの使い方

各項目のアセスメント方法

各項目は合計**8点満点**で評価します。

メトリクスの計測	チームの人数は、5人以上12人以下か。(ピザ2枚ルール)	はい / いいえ
学習と改善	チームの仕事にある特定の人物に属人化した仕事を洗い出し、減らしていく仕組み・習慣があるか。	はい / いいえ
プラクティスと習慣	チームの権限について、RACI図やデリゲーションポーカーなどによって、可視化され共有されているか。	はい / いいえ
	チームは価値提供をするのに必要な全職能のメンバーで構成されているか。(フィーチャーチーム)	はい / いいえ
	チームおよびチームリーダーは、チームのミッションのために必要な外部のリソースを調達するための予算や権限を持っているか。	はい / いいえ
アンチパターン	チームは存在するが、それぞれのやっている仕事の内容をよく知らないし、代わりにやることもできない。	はい / いいえ
	チームリーダーが複数のチームやプロジェクトを兼務しており、自チームのためにすべての時間を使うことができない。	はい / いいえ
	チームの仕事にある特定の人物に属人化した仕事を洗い出し、減らしていく仕組み・習慣があるか。	はい / いいえ

「メトリクスの計測」、「学習と改善」、
「プラクティスと習慣」のポジティブな項目は、
回答が「はい」であれば1点、「いいえ」なら0点。

「アンチパターン」のポジティブな項目は、
回答が「いいえ」であれば1点、「はい」なら0点のように
評価が逆転します。

各項目のアセスメント方法

「はい、でも・・・」「いいえ、でも・・・」といった状態は0.5点換算

入力規則	ポイント	備考
Yes	1	実施できている。当たり前になっている。
Yes, but	0.5	実施したが、不完全な状態。あるいは辞める予定になっている。
No, but	0.5	実施していないが、過去に実施してやめた。あるいは、これから実施予定。
No	0	実施できていない。

実際にアセスメントに利用しようとする、ゼロかイチかで評価するのが難しいこともあります。

たとえば、あるプラクティスを来月から実施することになっているとか実施してみたら、色々問題があって今は停止中とかそういった状態がありえます。

このような状態に関しては半分の評点として扱うようにしています。一度だけの失敗ですべてやめてしまうのではなく、当たり前の習慣になるように何度も挑戦していくことが重要です。



アセスメントシートの利用

テーマ	No	カテゴリ	メトリクスの計測	学習と改善	プラクティスと習慣	アンチパターン
team	1	チーム構成と権限委譲	Yes	Yes	Yes	No
team	2	チームビルディング	Yes	Yes	No	Yes
team	3	心理的安全性	No	Yes	Yes	No
team	4	タスクマネジメント	Yes	No, but	No, but	Yes
team	5	透明性ある目標管理	Yes	Yes	Yes	Yes, but
team	6	経験主義的な見積りと計画	No, but	No	Yes	No
team	7	振り返り習慣	No	Yes, but	Yes, but	No
team	8	バリエーションとチーム最適化	Yes	Yes	Yes	Yes
system	1	バージョン管理	No	No	No	No
system	2	ソースコードの可読性	Yes	Yes	No, but	No
system	3	継続的インテグレーション	Yes	No, but	Yes, but	Yes
system	4	継続的デプロイ	Yes, but	No, but	No	Yes
system	5	API駆動開発	Yes	No, but	Yes	Yes
system	6	疎結合アーキテクチャ	No	Yes	Yes, but	No
system	7	システムモニタリング	No	Yes	Yes	No, but
system	8	セキュリティシフトレフト	Yes	Yes	Yes	Yes, but
data	1	顧客接点のデジタル化	No, but	Yes	Yes, but	Yes
data	2	非構造化データの収集	No	No	No, but	No
data	3	データ整理・分析基盤	No	No	Yes	Yes, but
data	4	データ処理パイプライン	No	Yes	No	Yes
data	5	データ可視化とリテラシー	Yes, but	No, but	Yes	Yes, but
data	6	機械学習プロジェクト管理	No, but	Yes, but	Yes	No, but
data	7	マーケティング自動化	Yes	No	Yes, but	No
data	8	自動的な意思決定	No	No	Yes	Yes
design	1	ペルソナの設定	No, but	Yes, but	Yes	No, but
design	2	顧客体験	Yes	Yes, but	Yes	No
design	3	ユーザーインタビュ	Yes	Yes	Yes	Yes
design	4	デザインシステムの管理	Yes	No	Yes	No, but
design	5	デザイン規範	Yes, but	No, but	Yes	Yes
design	6	プロトタイプング	Yes	No	No	Yes, but
design	7	ユーザビリティテスト	Yes	No	Yes	No, but
design	8	プロダクトマネジメント	Yes	Yes	Yes	No
corporate	1	スパン・オブ・コントロール	Yes	No, but	Yes	No
corporate	2	開発者環境投資	Yes	Yes	No	Yes, but
corporate	3	コミュニケーションツール	Yes	Yes, but	Yes	No
corporate	4	人事制度・育成戦略	Yes	No	Yes	Yes
corporate	5	デジタル人材採用戦略	Yes, but	Yes	No	No
corporate	6	モダンなITサービスの活用	No	Yes	No	No, but
corporate	7	経営のデジタルファースト	Yes, but	Yes	No	Yes, but

カテゴリ	メトリクスの計測	学習と改善
チーム構成と権限委譲	Yes	Yes
チームビルディング	Yes	Yes
心理的安全性	Yes, but	No
タスクマネジメント	No, but	Yes
透明性ある目標管理	No	Yes
経験主義的な見積りと計画	No	No
振り返り習慣	No	Yes, but

今後、ツールなどで評価できるようにしていく予定です。
当初はgoogle spreadsheetベースのアセスメントシートがご利用いただけます。



アセスメントシートの利用：評点

テーマ	No	カテゴリー	合計								カテゴリー	
team	1	チーム構成と権限委譲	3.0	1.0	1.0	3.0	3.0	3.0	3.0	3.0	8.0	47.0
team	2	チームビルディング	3.0	3.0	3.0	3.0	3.0	3.0	3.0	5.0		
team	3	心理的安全性	3.0	3.0	1.0	3.0	3.0	3.0	3.0	5.0		
team	4	タスクマネジメント	3.0	3.0	3.0	3.0	3.0	3.0	3.0	5.0		
team	5	透明性ある目標管理	3.0	3.0	3.0	3.0	3.0	3.0	3.0	7.5		
team	6	経験主義的な見積りと計画	0.5	3.0	0.5	3.0	3.0	3.0	3.0	4.0		
team	7	振り返り習慣	0.0	0.5	0.5	3.0	3.0	3.0	3.0	5.5		
team	8	バリューストリーム最適化	3.0	3.0	3.0	3.0	3.0	3.0	3.0	7.0		

テーマ	No	カテゴリー	合計								カテゴリー	
team	1	チーム構成と権限委譲	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	8.0	47.0
team	2	チームビルディング	1.0	1.0	1.0	0.0	1.0	1.0	0.0	0.0	5.0	
team	3	心理的安全性	1.0	0.0	1.0	1.0	1.0	1.0	0.0	0.0	5.0	
team	4	タスクマネジメント	1.0	1.0	0.5	0.5	1.0	1.0	0.0	0.0	5.0	
team	5	透明性ある目標管理	1.0	1.0	1.0	1.0	1.0	0.5	1.0	1.0	7.5	
team	6	経験主義的な見積りと計画	0.5	0.0	0.0	1.0	0.0	1.0	1.0	0.5	4.0	
team	7	振り返り習慣	0.0	0.5	0.5	1.0	1.0	1.0	1.0	0.5	5.5	
team	8	バリューストリーム最適化	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	7.0	

アセスメントシートにデータを記入していただくと、自動的に評点が集計されます。各項目の総合点や、テーマごとの点数などを見て改善点を見つけ出すための議論の土台にしましょう。

system	1	バージョン管理	0.5	3.0	0.5	1.0	1.0	1.0	0.0	2.0	33.5	
system	2	ソースコードの明確さ	0.5	1.0	0.5	3.0	3.0	3.0	3.0	6.0		
system	3	継続的インテグレーション	1.0	1.0	0.5	0.5	0.5	1.0	1.0	4.0		
system	4	継続的デプロイ	0.5	0.5	0.5	0.5	0.0	0.0	3.0	1.0		
system	5	API自動開発	1.0	0.5	1.0	1.0	1.0	0.0	1.0	6.5		
system	6	積極的なアーキテクチャ	0.0	1.0	0.5	0.0	0.5	0.0	1.0	3.0		
system	7	システムモニタリング	0.5	1.0	1.0	0.0	0.5	0.0	3.0	3.5		
system	8	セキュリティシフトLEFT	3.0	3.0	1.0	0.5	1.0	0.0	1.0	5.5		
data	1	顧客視点のデジタル化	0.5	3.0	0.5	1.0	3.0	1.0	0.0	5.0	34.5	
data	2	事業活動データの収集	0.0	0.5	0.5	0.0	1.0	0.0	1.0	2.5		
data	3	データ集積・分析基盤	0.5	3.0	1.0	0.5	3.0	3.0	1.0	3.0		4.0
data	4	データ処理パイプライン	0.0	1.0	3.0	3.0	0.5	0.5	3.0	4.0		
data	5	データ可視化とリテラシー	0.5	3.0	1.0	0.5	1.0	0.5	1.0	3.0		4.5
data	6	機械学習プロジェクト管理	0.5	3.0	1.0	0.5	3.0	1.0	1.0	1.0		6.0
data	7	マーケティング自動化	1.0	3.0	1.0	0.5	3.0	1.0	0.5	3.0		4.5
data	8	自動的な意思決定	0.0	0.0	1.0	3.0	3.0	0.5	1.0	4.0		
design	1	ペルソナの設定	0.5	3.0	1.0	0.5	3.0	0.5	0.5	3.0	33.5	
design	2	顧客体験	1.0	3.0	1.0	0.0	3.0	0.5	3.0	3.5		
design	3	ユーザーインタビュー	1.0	1.0	1.0	1.0	1.0	0.5	1.0	6.5		
design	4	デザインシステムの管理	1.0	3.0	1.0	0.5	0.0	0.0	0.0	3.0		
design	5	デザイン組織	0.5	0.5	1.0	0.5	0.0	0.0	0.0	4.0		
design	6	プロトタイピング	3.0	3.0	0.5	0.5	1.0	1.0	1.0	4.5		
design	7	ユーザビリティテスト	1.0	0.0	0.0	1.0	0.5	1.0	0.0	4.5		
design	8	プロダクトマネジメント	1.0	1.0	1.0	0.5	1.0	0.0	0.5	4.5		
corporate	1	スパン・オブ・コントロール	1.0	0.5	1.0	0.0	3.0	3.0	1.0	4.5	36.5	
corporate	2	開発者環境改善	1.0	1.0	0.5	0.5	3.0	1.0	3.0	5.0		
corporate	3	コミュニケーションツール	1.0	3.0	1.0	0.0	1.0	1.0	1.0	6.5		
corporate	4	人事制度・育成戦略	0.0	1.0	0.0	3.0	0.0	1.0	1.0	4.5		
corporate	5	デジタル人材採用戦略	0.5	1.0	0.5	0.5	0.5	1.0	1.0	4.5		
corporate	6	モダンなITサービスの活用	0.0	1.0	1.0	3.0	3.0	0.0	0.0	3.5		
corporate	7	経営のデジタルフォースト	0.5	1.0	0.5	0.5	1.0	1.0	0.0	4.0		
corporate	8	次世代のセキュリティ	0.5	1.0	0.0	0.5	3.0	1.0	0.0	4.0		
合計			25	25	68	66	185					

アセスメントシートの使い方

ファイルにアクセス

コピーの作成

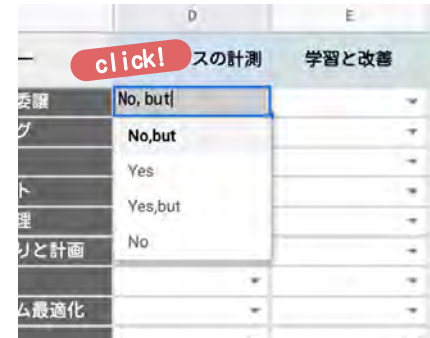
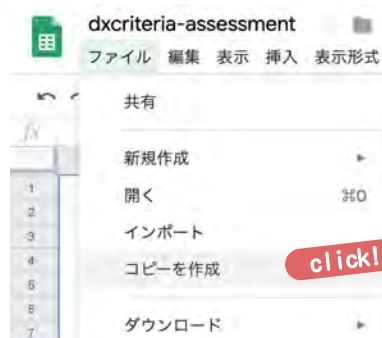
情報入力



or

click!

公開URL



上記URLかQRコードから、公開中のDX Criteriaのアセスメントシートファイルにアクセスしてください。

アクセス先は「閲覧のみ」のファイルになっていますので、ご自身のGoogle Driveにコピーを作成してください。

コピーしたファイルの「assessment」のシートに記入例を参考に入力してみてください。

DX Criteriaの 3つの使い方

その1

自社のDX進捗度の簡易的なアセスメント

その2

チームとシステムごとの詳細なアセスメント

その3

外部パートナーとのコミュニケーション手段



ご利用上の注意点



理解をせずに導入する/導入しない

DX Criteriaの目的は、不確実な時代に必要な事業活動の競争力を得ることです。

一つ一つ実践しながら、体感的な理解を積み重ねていくことが重要です。そのうえで、自社にあった適切な形を模索していくためのきっかけとしてご利用ください。



過度に数字を気にしすぎる

DX Criteriaでは、適切なメトリクスを測ることでより議論が明確化し、活発な改善と対話を促進していきたいと考えています。

しかし、これらを経営が一方的に数値目標としてしまうと、本来の価値を喪失してしまいます。



内容より結果に注目する

DX Criteriaは、高速な仮説検証をする組織が持つ習慣や文化・ケイパビリティに注目するものです。

そのため、すべての項目を満たせばよいというのではなく、自社の事業速度において、どこがボトルネックになっているかを判断した上でお使いください。



誰かを攻撃するのに使う

DX Criteriaは、基準を満たさない誰かを攻撃するためにつくられたものではありません。

これらの基準を通じて、ソフトウェア開発の見えない性質に対する理解が促進され、より発展した議論に導くためのものです。

4. 詳細項目一覧



4-1. チームのチェック項目

4-2. システムのチェック項目

4-3. データ駆動のチェック項目

4-4. デザイン思考のチェック項目

4-5. コーポレートのチェック項目



チームのチェック項目

チーム

システムの開発チームが素早く仮説検証するためには、十分な権限委譲がされた小さなチームであることが重要です。

また、経験主義的・仮説検証的に不確実な世界と向き合っていく文化を持つことがとても重要になります。

問題や課題を相互に言いやすく、解決していけるという確信が持てる人間関係の構築と事実に基づいた観察とふりかえりが強い開発チームを作り出します。

1	チーム構成と権限委譲
2	チームビルディング
3	心理的安全性
4	タスクマネジメント
5	透明性ある目標管理
6	経験主義的な見積りと計画
7	ふりかえり習慣
8	バリューストリーム最適化

チーム評価項目 01

チーム構成と権限委譲

メンバーが多すぎるチームも少なすぎるチームも、管理を複雑にするばかりでなく、生産性を悪化させてしまうことが知られています。

また、高速に仮説検証を行うためには予算・意思決定・実行能力を1つのチームで保つ必要があります。

メトリクスの計測	システムを開発するチームの人数は、5人以上12人以下か。 (ピザ2枚ルール)	はい / いいえ
学習と改善	ある特定の人物に属人化した仕事を洗い出し、減らしていく仕組みや習慣がチームにあるか。	はい / いいえ
プラクティスと習慣	チームとチームメンバーの権限について、RACI図やデリゲーションポーカーなどによって、可視化され共有されているか。	はい / いいえ
	チームは価値提供をするのに必要な全職能のメンバーで構成されているか。(フィーチャーチーム)	はい / いいえ
	チームおよびチームリーダーは、チームのミッションのために必要な外部のリソースを調達するための予算や権限をもっているか。	はい / いいえ
アンチパターン	チームは存在するが、それぞれのやっている仕事の内容をよく知らないし、代わりにやることもできない。	はい / いいえ
	チームリーダーが複数のチームやプロジェクトを兼務しており、自チームのためにすべての時間を使うことができない。	はい / いいえ
	チームリーダーがメンバーに権限委譲できておらず、ボトルネックになっている。	はい / いいえ

チーム評価項目 02

チームビルディング

システムを改善するチームは、組成してからパフォーマンスするまでに時間がかかります。

多くの場合、継続的に同じメンバーで構成されることで、学習が進み高速な開発が実現できるのです。一方で、あまりに長い期間同じメンバーでも属人化や改善のサイクルが止まってしまう。

メトリクスの計測	チームは少なくとも半年以上継続して存在しているか。	はい / いいえ
学習と改善	チームは月に一度以上の頻度で仕事のふりかえりをおこなっており、その際にプロジェクト憲章またはインセプションデッキの認識を揃えているか。	はい / いいえ
プラクティスと習慣	インセプションデッキまたはプロジェクト憲章を作成し、チームの存在理由についてチーム全員が把握しているか。	はい / いいえ
	新しくチームに参画するメンバー用のオンボーディング・デッキ（チームの一員として働き始めるための、価値観・実務・スキル・相互理解のための明文化されたドキュメント集）が存在するか。	はい / いいえ
	チームメンバー全員で定期的にかジュアルにコミュニケーションをとる場がある（ランチ、ディナー、レクリエーション等）	はい / いいえ
アンチパターン	オンボーディングプログラムが、文章を読むだけのものになっており、ハンズオンやミッション理解の伴わない形骸化したものになっている。	はい / いいえ
	1年以上チームのやる事が変わっておらず、チームメンバーも固定されている。	はい / いいえ
	チーム内でチームミッションの改善に関する議論がどんな理由であれ発生していない。	はい / いいえ

チーム評価項目 03

心理的安全性

チームメンバーが相互に自分の意見を言ったとしても、不利益を受けることがないという環境がソフトウェア開発の生産性につながるということが知られています。

このような状態を心理的安全といえます。心理的安全性について、継続的に投資することはチームの生産性に投資することにつながります。

メトリクスの計測	チームメンバーの心理的安全性を測る指標があり、定期的に計測しているか。	はい / いいえ
学習と改善	1on1や、落ち着いた場面でのオフサイトミーティングなどの直接業務に関わらないキャリアやタスクの壁打ちを、月に1度程度は実施しているか。	はい / いいえ
プラクティスと習慣	チームメンバーの行動（成果ではなく）/発言を明示的に承認行動をとる習慣があるか。（行動したこと自体に対して、拍手する・感謝を述べるなど）	はい / いいえ
	チームメンバー間で挨拶をしたり、雑談をする習慣はあるか。	はい / いいえ
	チームの不安や不満などを可視化し、吸い上げるための仕組みを持っているか。	はい / いいえ
アンチパターン	ミッションや共通のゴール設定をしないまま意見を集め、課題のためというよりも個人のための意見しか出てこない状況になっている。	はい / いいえ
	心理的安全性を仲の良さで捉えて、事業のための意見ではなく、仲良くすることが目的化しているため、意見を封殺してしまう。	はい / いいえ
	事業目標や納期目標に対して、威圧的なマネジメントや権威的な命令を繰り返したことで、意見が出てこない状況になっている。	はい / いいえ

チーム評価項目 04

タスクマネジメント

チーム全体で、タスクが明文化されないまま口頭などでのやり取りが多かったり、曖昧な要求が多いとそれだけ手戻りの手間が生じてしまいます。

明文化しきれないことも多々ありますが、継続的に基準をアップデートすることで、ムラのない仕事のスピードを得ることができます。

メトリクスの計測	アイデアレベルの要望や構想から、要件に落ちるまでのリードタイムは計測されているか。	はい / いいえ
学習と改善	仕事を始めるための定義と、完了するための定義はチームやステークホルダーで定期的に見直されているか。	はい / いいえ
プラクティスと習慣	上長やステークホルダーを含め、共通のタスク管理ツールを利用しており、プロジェクトの状況を可視化したダッシュボードに常にアクセス可能か。	はい / いいえ
	チームが仕事を始めるために必要な課題やタスクの粒度について、明文化されたフォーマットが存在するか。	はい / いいえ
	チームのタスクに関して、「完了の定義(Definition of DONE)」が存在するか。	はい / いいえ
アンチパターン	他部署からの依頼が明確なタスクツールではなく、担当者へのダイレクトメール/メッセージ/口頭など透明性のない形で行われている。	はい / いいえ
	どのチームタスクであるかが曖昧なとき、ボールが落ちないようにするための仕組みや文化が存在しない。	はい / いいえ
	どのチームタスクであるか曖昧な仕事が発生したあとに、事後検証（ポストモータム）が行われず都度話し合いで解決している。	はい / いいえ

チーム評価項目 05

透明性ある目標管理

強いチームを支えるのは明確な目標管理です。たとえば、OKRはそれを支えるツールになりえます。

目標はノルマではなく創造性を生み出すためのツールです。そのため、天下りの数字のブレイクダウンではなく、仮説とフォーカスをはっきりさせる必要があります。

メトリクスの計測	1年後などのチームとプロダクトの目指す姿が、言語化され、いくつかの計測可能な指標により明晰化されているか。	はい / いいえ
学習と改善	仮説的な目標に対して、うまくいかなかった際にチームとして「学んだこと」を言語化・他チームへ公開・学習しているか。	はい / いいえ
プラクティスと習慣	四半期にフォーカスすべき目標が言語化され、いくつかの計測可能な指標によって明晰化されているか。	はい / いいえ
	四半期にフォーカスすべき目標に対して、どのようにアプローチするのかの計画をチームで共有しているか。	はい / いいえ
	ビジネス上、重要なマイルストーンとそのスケジュールをチームで常に共有し、その進捗を確認しているか。	はい / いいえ
アンチパターン	目標管理が強く評価制度に結びついているため、ストレッチしたゴール設定をすることが難しくなっている。	はい / いいえ
	目標項目の一部に、その達成手段が健全に行われているかをチェックするための目標を立てていない。	はい / いいえ
	目標が定量的でなく、第三者からみて達成度合いが不明確なものになっている。	はい / いいえ

チーム評価項目 06

経験主義的な見積りと計画

アジャイルプロセスにおいては、計画や見積りを行わないという誤解を持っていることがあります。

本来は実験を繰り返すことで精度を高めながら、プロジェクト予測の正確性をたかめていくための計画管理を行うことが多いです。ミッションにおうじて適切な手法を選ぶ能力があるかを調査します。

メトリクスの計測	チームのベロシティを把握しており、その分散値の変化を計測しているか。	はい / いいえ
学習と改善	見積りと実績の履歴を元に、見積りの精度を向上させるための方法について定期的な学習/ふりかえりを行なっているか。	はい / いいえ
プラクティスと習慣	見積りは、実際にその仕事を行う本人を含むチームで行われているか。	はい / いいえ
	スケジュールがクリティカルなプロジェクトでは精密に、仮説検証や価値がクリティカルなプロジェクトでは粒度を荒くするなどして、状況に応じて見積りや計画の方法を変えているか。	はい / いいえ
	見積りと実績の履歴を元に、見積りの精度を向上させるための方法について定期的な学習/ふりかえりを行なっているか。	はい / いいえ
アンチパターン	スケジュールのバッファ（緩衝期間）を全体計画に対して25%以上確保していない。	はい / いいえ
	機能要件のバッファ（緩衝機能）を全体計画に対して設けており、「必須な機能」と「あったらよい機能」を分類しているか。	はい / いいえ
	相対見積りによって得られたベロシティの数値自体を、生産性の指標にしており、数値的なコミットメントや改善が要求されている。	はい / いいえ

チーム評価項目 07

ふりかえり習慣

チーム活動において、ふりかえりは最も基礎的で最も重要な習慣です。

一方で当たり前すぎるために議論が発散したり、マンネリ化を招いてしまうことがあります。

テーマを決めて事実に基づいたふりかえりを行う習慣があるかを検査します。

メトリクスの計測	ふりかえりのテーマごとに数字を計測し、ファクトベースで議論できるようにしている。	はい / いいえ
学習と改善	チームのメンバーは、チームで合意した1カ月以内のサイクルでふりかえりを実施しているか。	はい / いいえ
プラクティスと習慣	ふりかえりの場合はチーム全員が参加しているか。	はい / いいえ
	ふりかえりはテーマを定め、議論を行い次回のふりかえりまでに実行可能なタスクが切り出されているか	はい / いいえ
	ふりかえりにおいて前回のふりかえりで改善するために起案したタスクが実行されたか検証しているか	はい / いいえ
アンチパターン	ふりかえりをしていたが、しばしば意見が出なかったため、ふりかえり自体をやめた。	はい / いいえ
	ふりかえるべきテーマに関しての、起きた出来事を時系列に事実関係を整理するなどの準備をせずにふりかえりをすすめている。	はい / いいえ
	ふりかえりに適切なファシリテーターがない。	はい / いいえ

チーム評価項目 08

バリューストリーム最適化

高速な仮説検証サイクルを行うチームにおけるマネジメントは、一時的なプロジェクトとしてのタスク管理やスケジュールの管理ではなく、継続的にすばやく機能を仮説検証することができるのかという効率性を重視します。

そのため、各工程でのサイクルタイムやリードタイムを計測し、それらを短くするように活動します。

メトリクスの計測	チームのリードタイム、フロー効率性および各工程のサイクルタイムを継続的に計測しているか。(またはエンジニアリングインテリジェンスのサービスを利用している)	はい / いいえ
学習と改善	チームのバリューストリームマッピングを作成し、繰り返しボトルネックを把握しながら自動化と学習を繰り返しているか。	はい / いいえ
プラクティスと習慣	バリューストリーム改善のために、開発リソースの10%以上を継続的に確保しているか。	はい / いいえ
	設定ファイルや一部のソースコードに対して、エンジニアでなくても必要に応じて修正のためのPull Request (Merge Request) を投げることはあるか。	はい / いいえ
	ペアプログラミング/モブプログラミングを実施しているか。	はい / いいえ
アンチパターン	属人的なタスクがある状態を効率的だと解釈して改善をしない。	はい / いいえ
	フロー効率性などの数値指標に囚われすぎて、全体のリソース効率が悪化したり、価値提供に結びつかない状態にならない。	はい / いいえ
	数値改善のために簡単に予測可能な仕事ばかりを引き受け、難しめのタスクを引き受けない。	はい / いいえ



システムのチェック項目

システム

ソフトウェアをシンプルに保ち、ビジネス上の様々な要求に応えやすくするためには、開発者の活動を支援する様々な自動化やアーキテクチャ設計への投資が必要不可欠です。

ですが、目に見える機能と異なり、ソフトウェアの中身に宿る品質への投資は経営者やエンジニアでないものには理解されにくいものです。

目に見えない投資について、「見える化」をしながら改善がなされていることが重要な視点です。

1	バージョン管理
2	ソースコードの明確さ
3	継続的インテグレーション
4	継続的デプロイ
5	API駆動開発
6	疎結合アーキテクチャ
7	システムモニタリング
8	セキュリティシフトレフト

システム評価項目 01

バージョン管理

ソースコードのバージョン管理は、最も基本的な開発者の習慣です。

このような習慣がない場合、共同でソフトウェアを開発すること自体が困難になり、属人化したノウハウや暗黙的なルールの温床になりえます。また、GitHubなどのサービスはさまざまな開発者向けツールの起点となる重要な基盤でもあります。

メトリクスの計測	バージョン管理システムの履歴情報(Code Churn)の分析をもとにバグ予測や品質上の問題を指摘するツールを導入し、継続的に改善しているか。	はい / いいえ
学習と改善	明文化されたランチ戦略が存在するか。そして、それは守られているか。	はい / いいえ
プラクティスと習慣	すべてのアプリケーションコードをGit/GitHubなどのバージョン管理システムで自社管理しているか。(権利を有する全てのソースコードについて、自社が管理するバージョン管理システムで扱っているか)	はい / いいえ
	インフラ構成とシステム要素のプロビジョニングをソースコードとして実行可能な形式にした上で、バージョン管理システムで管理しているか。(Infrastructure as Code)	はい / いいえ
	統合テスト/デプロイメントの自動化に関わるソースコードをアプリケーションコードと同一のバージョン管理システムで管理しているか。	はい / いいえ
アンチパターン	ソースコード自体のセキュリティレベルを高く設定しており、開発支援系SaaSの利用を禁止している。	はい / いいえ
	バージョン管理システムが複数存在していたり、1つのツールからすべての履歴を閲覧することができないなど、中途半端な状態のままになっていないか。	はい / いいえ
	システムのソースコードの閲覧を関連するエンジニアのみに限定している。(別チームのエンジニアや他のステークホルダーが閲覧できない。)	はい / いいえ

システム評価項目 02

ソースコードの明確さ

複雑なソースコードは、それだけで自動的にテストを難しくし、バグや障害を生み出しやすいだけでなく、開発者のモチベーションや生産性に悪影響を与えます。

ソフトウェアが経営から不可視である要素から、内部的な品質が軽視される傾向があるため、十分な注意をはらい続ける必要があります。

メトリクスの計測	アプリケーションコードの循環的複雑度などのメトリクスを、ツール/サービスを用いて継続的に計測しているか。	はい / いいえ
学習と改善	デッドコードを四半期以上のサイクルで定期的に棚卸しし、削除や分解をしているか。	はい / いいえ
プラクティスと習慣	コードレビューをする習慣や規則があり、masterブランチへのマージはコードレビューを必須としているか。	はい / いいえ
	明文化されているコードレビューガイドラインが存在するか。	はい / いいえ
	コードレビューガイドラインを満たさないコードを自動的に検出・補正する各種Linterやフォーマッタなどのツール群を整備しており、ソースコードを変更する誰もが使うことができるか。	はい / いいえ
アンチパターン	コードレビューを行うことができる人物がチームの中におらず、レビュー待ちに1、2営業日がかかる。	はい / いいえ
	コードレビューガイドラインは1年以上メンテナンスされておらず、形骸化している。	はい / いいえ
	コードレビューガイドラインの多くの項目が、自動的なフォーマッタなどで統一・解決可能な些末な事柄である。	はい / いいえ

システム評価項目 03

継続的インテグレーション

継続的インテグレーションとは、自動的に定期的に実施される結合テスト環境のことです。

この環境が簡単でかつ信頼できるほど、開発者は、誰かの手作業によるテストを待つ必要がなくなり、自分の手元でソースコードの改善を繰り返しやすくなります。これは生産性と品質向上に寄与します。

メトリクスの計測	すべてのインテグレーションテストにかかる時間が計測されており、それは30分以内に完了するか。	はい / いいえ
学習と改善	テストカバレッジ基準や自動テストガイドラインを用意し、これらを継続的に改善するための工数がチームで割かれているか。	はい / いいえ
プラクティスと習慣	プロダクトの半分以上のモジュール/クラスファイルに対して、ユニットテストが存在しているか。	はい / いいえ
	テスト用データやスタブ/モックなどを整備し、テストを書きやすくするための環境整備をしているか。	はい / いいえ
アンチパターン	継続的インテグレーション環境が存在し、開発者は開発ブランチの全テストをリソース調整することなく、自由に行うことができるか。	はい / いいえ
	一部の人だけがテストを書き、一部の人にはテストを書かないといったように自動テスト習慣を個々人の努力目標などになっている。	はい / いいえ
	テスト自体が複雑になって、長期間メンテナンスされていない。	はい / いいえ
	自動テストが失敗したまま、そのコードが本番デプロイされることを許容している。	はい / いいえ

システム評価項目 04

継続的デプロイ

継続的デプロイとは、完成したソースコードを自動的かつ簡単・安全にサービスインするための仕組みです。

この環境への投資が整っていない場合、開発者は本番環境のリリースのたびに様々な作業負荷がかかり、仮説検証や継続的な品質改善に対しての足止めになってしまいます。

メトリクスの計測	デプロイ頻度とデプロイ成功率を継続的に測定しており、これらを改善することを目標管理しているか。	はい / いいえ
学習と改善	デプロイ時に社内のユーザーや開発者のみを対象もしくは、一部のサーバのみにサービスをリリースしてエラーがないかを確認めるカナリアリリースができるか。	はい / いいえ
プラクティスと習慣	デプロイ完了時、および構成変更時にインフラ構成に関する自動的なテスト(e2eのスモークテストおよびServer specなどのインフラ環境テスト)を実行しているか。	はい / いいえ
	ブルーグリーンデプロイメントができるか。(稼働中のサーバを切り替えるのではなく、別環境にデプロイ作業をしてから本番の向き先を切り替えるデプロイ手法。)	はい / いいえ
	デプロイ作業を伴わず、一部の機能を安全にオフにしたり、オンにしたりすることができるか。(Feature Toggle /Soft Launch/ Dark launchなどの仕組みを導入・実装しているか。)	はい / いいえ
アンチパターン	デプロイされたコードに問題が発生した際に、前のバージョンへの切り戻しを意思決定してから5分以内に切り戻すことができない。	はい / いいえ
	開発者のメンバー自身が、権限を持つ人物の承認があっても、自分のコードを本番環境にデプロイすることができない。	はい / いいえ
	デプロイ工程が自動化されておらず、本番反映に1時間以上かかっていたり、特定の時間帯しかできないなどの制約事項がかかっている。	はい / いいえ

システム評価項目 05

API 駆動開発

ネットワーク経由のAPIを基準にシステムを開発することで、他のシステムと連携しやすくなります。

またシステムがレガシー化した際に交換したり、改善したりといった手が打ちやすいものになります。

人が使う見た目の作りだけでなく、エンジニアにとっての作りが質を生み出します。

メトリクスの計測	社内外のAPIの利用者にとってのユーザビリティについてヒアリング/アンケートを行い継続的な改善が行われているか。	はい / いいえ
学習と改善	各APIについて、動作するインタラクティブなドキュメントや管理サービスを持っているか。	はい / いいえ
プラクティスと習慣	プロダクトに対して外部あるいは内部の別のシステムと連携するためのネットワーク経由APIが提供されているか。	はい / いいえ
	APIは何らかのSchema定義言語によって規定され、そこから自動的にクライアントの生成やバリデータの生成が行われているか。	はい / いいえ
	APIに関わる要件は、SDD(スキーマ駆動開発)で開発され、直ちにモックアップサーバーが提供できるか。	はい / いいえ
アンチパターン	ViewやControllerの層に処理が集中しており、機能をAPIに切り出すことが困難な設計になっている。	はい / いいえ
	各APIに対して、ネットワークを経由したE2E(ステージング・本番どちらに対するE2Eテストかは問わない)のテストが存在しておらず、死活監視ができていない。	はい / いいえ
	APIがバージョン管理されておらず、破壊的な変更が利用者から検知できない。	はい / いいえ

システム評価項目 06

疎結合アーキテクチャ

システムの役割は単純であればあるほど、高速に改善しやすくなります。

複雑な問題を解くときに単純な問題野組み合わせにするというのは重要な設計技術です。

疎結合なアーキテクチャとは、このように改善する単位を単純に保つための技術です。

メトリクスの計測	目指すべきアーキテクチャに対してすぐわれない点を洗い出すための仕組みが存在しており、それらの情報をもとに改善を進めているか。（アーキテクチャ適応度関数）	はい / いいえ
学習と改善	システムアーキテクチャの決定・変更・改善に関するドキュメントを管理し継続的な学習機会を設けているか。	はい / いいえ
プラクティスと習慣	ドメインイベントの発火に伴いPublish/Subscribeモデルを利用した仕組みで、関連サービスとの連携が可能か。またその履歴データが保存管理され、これらのイベントリプレイから再突合や監査の自動化が可能か。	はい / いいえ
	結果整合性を考慮したサービスレベルの合意が要件のガイドラインの中に組み込まれているか。 バッチ、ジョブ、プロシージャに対する冪等な設計ガイドラインが存在しており、再送によって整合性が担保できるようなシステムになっているか。	はい / いいえ
アンチパターン	1つのデータベースに対して複数のシステムからの直接的な参照または書き込みがなされていて、それらの依存性が簡単には追跡できない状況になっている。	はい / いいえ
	疎結合なシステムであるが、分散トレーシングの仕組みがなく、問題発生時の原因特定に時間がかかる。 自動テストとスキーマ定義の存在しない外部システムとの依存性が10個以上存在しており、機能開発の影響範囲を特定できない。	はい / いいえ

システム評価項目 07

システムモニタリング

システムの品質は、エラーや障害などから学び改善していくことで生まれます。

そのためには、エラーや障害について検知し、復旧するためのモニタリングとその改善を支援するための文化と技術が必要になります。

ミスを許さない懲罰的な文化のもとでは、重大事故が起きやすくなります。

メトリクスの計測	SLI/SL0/エラーバジェットがビジネスオーナーとエンジニアが協議して合意の上設定され、計測されているか。	はい / いいえ
学習と改善	開発と SRE が共有する障害報告リストがあり、それぞれに有効な再発防止の仕組みが整うようにリソースを割いているか。	はい / いいえ
プラクティスと習慣	APMのためのSaaSやツールが導入され、ユーザーからみた応答速度などについて、要因ごとの影響度を定常的に分析できる状態になっているか。	はい / いいえ
	フォルトインJECTIONテストや、カオスエンジニアリング等の仕組みを導入することで、重大事故につながるかもしれないシステムの欠陥を早期に発見するための試みをおこなっているか。	はい / いいえ
	オートスケールなどの仕組みにより、開発者やSREが介在しなくても、適切なキャパシティコントロールができているか。	はい / いいえ
アンチパターン	障害の発生に対しての罰則や謝罪などの、開発者が萎縮したり障害を隠蔽する方向につながるような慣習が存在するか。	はい / いいえ
	定常的に発生しているサービス上の警告を問題ないものとして無視したり、ログ自体を出さないようにしている。	はい / いいえ
	システム構成要素の構築方法や運用方法が属人化しており、同じインスタンスを構築することができない。	はい / いいえ

システム評価項目 08

セキュリティシフトレフト

ソフトウェアが完成したあとにセキュリティの課題が見つかったと、そのために対応に追われたり、リリースが遅れたりの良いことはありません。

できる限り早い工程でセキュリティの課題を見つけ出す技術と文化を、DevSecOpsあるいはセキュリティのシフトレフトと言います。

メトリクスの計測	CI/CDのパイプラインにソースコードの自動的なセキュリティチェック（静的解析または動的解析）が組み込まれていて、一定の基準を達しないとリリースされない仕組みになっているか。	はい / いいえ
学習と改善	セキュアコーディングについて、開発者を対象にした教育カリキュラムや研修を実施しているか。	はい / いいえ
プラクティスと習慣	専門的なアプリケーションセキュリティの知識を持つメンバーが、専任でセキュリティチームにおり、動向や最新情報をもとに自社サービスをレビュー・改善できているか。	はい / いいえ
	OSSのライブラリやミドルウェアを使用する際、それらの脆弱性情報を自動的モニタリング・警告・パッチ適用するための仕組みまたはサービス等を利用しているか。	はい / いいえ
	4半期から1年の間で定期的に、全体的なアプリケーションとインフラの脆弱性診断を受けているか。	はい / いいえ
アンチパターン	開発速度（デプロイ頻度）を低下させるようなセキュリティルールが、施行されていて現況に合わせたアップデートが行われていない。	はい / いいえ
	ソースコード中に、漏洩してはならない情報がハードコーディングされている。（それらを分離して管理するようなツールまたは仕組みを導入しているか）	はい / いいえ
	開発企画要件の段階で、設計レベルのセキュリティレビューが実施されていない。（Security by Designの未実施）	はい / いいえ



データ駆動のチェック項目



データ駆動

「データの世紀」と呼ばれるように、企業の競争戦略にとってデータの利活用は必要不可欠なものです。

しかし、そもそもデータの取得ができていなかったり、データのリテラシーが低くうまく経営に行かせないということも多くあります。

また、機械学習やデータサイエンスの知見を利用したアプリケーションには、それを支えるビッグデータ処理の仕組みが合わせて必要になります。

1	顧客接点のデジタル化
2	事業活動データの収集
3	データ蓄積・分析基盤
4	データ処理パイプライン
5	データ可視化とリテラシー
6	機械学習プロジェクト管理
7	マーケティング自動化
8	自動的な意思決定

データ評価項目 01

顧客接点のデジタル化

B2CでもB2Bでも、顧客接点がデジタル化していないと、データと顧客を結びつけることが難しくなります。

データ利活用のポイントは、顧客接点に十分コントロールできるシステムが提供できているかが第一歩になります。これは、オンラインに限らず、リアルな接点であっても様々な手段でデジタル化をすることが可能です。

メトリクスの計測	デジタルな手段で行動履歴データを分析可能な形で保存している顧客が全体の7割を超えているか。	はい / いいえ
学習と改善	オンライン・オフラインの両方で、顧客の接点となる行動情報や通知の手段を獲得するためのシステムを開発している組織が社内是否存在するか。	はい / いいえ
プラクティスと習慣	オンライン上で、顧客は自社のサービスを契約したり購入したりできるか。	はい / いいえ
	自社サービスやメディアをスマートフォン用のWebサイトまたはアプリとして提供しているか。	はい / いいえ
	潜在顧客獲得のために自社メディアやSNSを通じたエンゲージメント活動をしているか。	はい / いいえ
アンチパターン	デジタル上でのプッシュマーケティングをEmailのみに頼っている。	はい / いいえ
	自社のリアルな顧客接点からのデータ収集が技術的な課題・社内ルール・オペレーションの問題でできていないままになっている。	はい / いいえ
	顧客接点のサービス開発がうまく機能していないため、改善の速度やデータの取得が遅延している。	はい / いいえ

データ評価項目 02

事業活動データの収集

データを中心とした価値創造を行うためには、顧客と事業の活動データが必要になります。

データ収集がまだまだの状態では、企業活動をデジタル化していくことは困難になります。

この点がどれだけ進んでいるかをチェックします。

メトリクスの計測	事業活動のデータが集計されてから、解析された状態で閲覧可能になるまでのリードタイムは1日以内か。(BIツールで昨日のデータは見られるか。)	はい / いいえ
学習と改善	事業活動の中に潜在的に存在するデータを収集するためのシステム化や業務分析を行うチームは存在するか。	はい / いいえ
プラクティスと習慣	外部事業者との取引情報について構造化されたフォーマットでリアルタイムにデータレイクに保存しているか。	はい / いいえ
	POSや業務システム上のアクセス記録/操作履歴を構造化されたフォーマットでリアルタイムにデータレイクに保存しているか。	はい / いいえ
	音声・動画・文章といった従来構造化できなかったデータソースも事業利活用のために収集しているか。	はい / いいえ
アンチパターン	データ収集プロジェクトが依頼したデータ記入が、現場で実施されていない。	はい / いいえ
	データの活用に関して責任のあるポジションを設置していない。	はい / いいえ
	データ収集に関するプロジェクトが存在しないか、進捗していない。	はい / いいえ

データ評価項目 03

データ蓄積・分析基盤

蓄積された生のデータは、構造化され、整理された上でデータ分析に使いやすい形のクラウドサービスなどのデータ分析基盤に格納されます。

これによって、データ分析のためのビッグデータ処理について知見があまりないエンジニアや関係者であってもデータを取得することができます。

メトリクスの計測	データ分析の基盤実際に操作して数字を取得することができる社内の人数と部署数を増やしていくための活動をしているか。	はい / いいえ
学習と改善	データ分析基盤をデータ分析基盤を職種を問わず使うために、簡単な分析をするためのプログラミングや操作の仕方をエンジニア以外のステークホルダーに対しても教育しているか。	はい / いいえ
プラクティスと習慣	ユーザー理解や仮説検証のためのデータ分析のための環境が整備されており、データサイエンティストだけでなくエンジニア・非エンジニアをとわず、プロダクトのステークホルダーに公開されているか。	はい / いいえ
	イベントストリーム処理の基盤を用いてオンライン情報を利用した分析・サービスでの活用をおこなっているか。	はい / いいえ
	データ分析において、個人情報をもスキミングする機構が存在しているか。	はい / いいえ
アンチパターン	事業システムのデータベースに直接アクセスして、分析用の処理を走らせている。	はい / いいえ
	統一されたデータレイク(未処理のローデータ保管場所)が存在せず、分析基盤ごとにデータを管理している。	はい / いいえ
	利用中の外部サービスにリアルタイムにデータ連携するための機構が存在しない。	はい / いいえ

データ評価項目 04

データ処理パイプライン

データ分析基盤から取得したデータをもとに機械学習や統計処理を利用したアプリケーションを実装するには、試行錯誤の実験フェーズとは異なる知見が必要になります。例えば、スケールして安定したデータ処理を行うワークフローパイプラインの基盤を作るなどです。

このような一般にMLOps/DataOpsと呼ばれる領域に関わる項目をチェックします。

メトリクスの計測	分析・開発や運用のバリューチェーン上の各種サイクルタイムを計測しており、継続的に改善しているか。	はい / いいえ
学習と改善	データレイクから、モデルの実サービス適用までの一連の流れのパフォーマンスモニタ・自動化・効率化を行うエンジニアリングチームが存在するか。	はい / いいえ
プラクティスと習慣	データレイクから、データ分析基盤までのETL処理にも自動テストが存在しており、変換エラーなどがモニタリングされているか。	はい / いいえ
	実運用されているデータ分析・学習のための前処理や学習処理を実行するためのワークフロー処理基盤(ETLの一貫性・冪等性・可用性を確保するための基盤)が存在するか。	はい / いいえ
	学習済みのモデルを検証し、サービスインするまでの処理は自動化されているか。	はい / いいえ
アンチパターン	週次集計や月次集計の処理が特定の日に終わることを想定しており、障害が発生した場合にデータが欠損してしまう。	はい / いいえ
	実験時の環境を実サービス環境に向けてポータブルにするためのコンテナ化やIaC(Infrastructure as Code)が存在しない。	はい / いいえ
	データサイエンス/機械学習/データアプリケーションエンジニアリング/クラウドインフラなどの知見が1名に属人化している。	はい / いいえ

データ評価項目 05

データ可視化とリテラシー

データ駆動経営を行うためには、データの適切な可視化とそれを解釈するリテラシーの両面が必要になります。

しばしば、データの取得や可視化を一部のエンジニアしか行うことができず、高速なサイクルを阻害してしまいます。

そこで、意思決定者自身がデータを直に取り扱えるのかをチェックします。

メトリクスの計測	プロダクトの主要情報のダッシュボードが、常に表示された共用のモニターをチームの座席近辺に配置しているか。(または常に意識されるように運用されているか。)	はい / いいえ
学習と改善	意思決定者は、データの読み取り方や統計の基本的な知識について研修トレーニングを受けているか。	はい / いいえ
プラクティスと習慣	売上などの短期的数値ではなく、長期的な事業価値のための間接的指標 (e.g. 顧客リピート率や予測LTVなど)を主要なKPIとして設定しているか。	はい / いいえ
	データ集計・表示のためのBIツールを導入しており、エンジニア以外でも使うことができているか。	はい / いいえ
	データから得られた推論や仮説が間違っている場合にどのようなデータによって検証可能かをもとにデータ収集や分析が行われているか。	はい / いいえ
アンチパターン	要望ベースでデータの集計を繰り返し、雑多なレポート項目が棚卸しされていない。	はい / いいえ
	簡単なデータ集計であっても、エンジニアを経由しなければ取得することができない。	はい / いいえ
	ダッシュボードは存在するが、データ担当者以外誰も見ておらず形骸化している。	はい / いいえ

データ評価項目 06

機械学習プロジェクト管理

機械学習のプロジェクトは、モデルの発見と顧客価値、サービスデプロイメント時の課題など複数の不確実性に対して適切なマネジメントが要求されます。

このことへの理解や組織学習が進んでいないと、PoCから成果が出なかったり、データエンジニアが離職してしまったりします。

メトリクスの計測	機械学習プロジェクトの具体的な成功指標を持ち、構築したシステムがそれを満たしているかモニタリングしているか。	はい / いいえ
学習と改善	機会学習の知見を、アプリケーション開発者や非エンジニアのスタッフが利用できるように勉強会などを繰り返し開いているか。	はい / いいえ
プラクティスと習慣	継続的再学習のためのモニタリングと、自動的なモデルのアップデート等の効率的な保守を実現するための仕組みを導入しているか。	はい / いいえ
	実運用時の計算量や計算資源の種別（IoT/エッジ利用/クラウド）などを考慮に入れた上で、モデル選定や実験のプロセスが動いているか。	はい / いいえ
	事業価値と実現可能性の両面を同時に仮説検証できるようなプロジェクト管理（PoCとプロトタイプ作成）を行っているか。	はい / いいえ
アンチパターン	機械学習チームと事業担当者との間で、事業理解や背景・目的の共有などの時間を十分に設けていない。	はい / いいえ
	すでに実績のあるクラウドを利用した機械学習ソリューションに対して、検証ができていない。	はい / いいえ
	機械学習プロジェクトのPoCから実運用するためのエンジニアリング能力をチームが持っておらず、そこから実サービス化が進まない。	はい / いいえ

データ評価項目 07

マーケティング自動化

見込み顧客の獲得から、関係性の構築、ナーチャリング、見込み度合いの高い顧客に対してのリテンション、顧客の再訪の促進などのようにマーケティング活動にはさまざまな工程があります。

その一部をデータとシステムを用いて自動化することで、人にしかできない戦略的な事柄にフォーカスできるようになります。

メトリクスの計測	マーケティングのオペレーションの業務の割合と企画・戦略の業務割合を棚卸して、自動化・最適化のためのリソースを割いているか。	はい / いいえ
学習と改善	インハウスのマーケティングチームに自動化や分析を行うエンジニアがおり、指標や自動化をともに勤めているか。	はい / いいえ
プラクティスと習慣	カスタマージャーニーの複数の接点において顧客の行動を把握、理解するため、何らかのDMPツールを導入しているか。	はい / いいえ
	媒体ごとに適切なクリエイティブになるようにA/Bテストやバンディッドアルゴリズムなどで最適化しているか。	はい / いいえ
	各施策ごとに獲得した顧客がその後のどのような購買行動/利用行動ができたかをコーホート分析しているか。	はい / いいえ
アンチパターン	獲得した顧客に対して、一括配信などの画一的な通達を行っており、投資効果の最適化を実施していない。	はい / いいえ
	顧客獲得に対して、一時的な獲得総数のみを目標としており、継続的な利用についてを調査していない。	はい / いいえ
	広告運用の成果報告がフォーマットに沿って自動的に行えるようになっていない。	はい / いいえ

データ評価項目 08

自動的な意思決定

データ駆動経営とは、機械が自動化できることを機械に任せ、人には創造的な仮説の構築や開発などの人にしかできない領域に集中して経営資源を投入することです。

そのためには、作業の単純化だけでなく意思決定も含むビジネスプロセス全体のアーキテクチャ設計が重要な経営上のケイパビリティとなります。

メトリクスの計測	自動化が進捗するために、判断基準が明確な目標を掲げているか。	はい / いいえ
学習と改善	意思決定や業務を自動化していくために、業務プロセスを改善するためのソフトウェアエンジニアを含むチームが存在するか。	はい / いいえ
プラクティスと習慣	意思決定に関する記録を、プロセスマイニングができる形で保存しているか。	はい / いいえ
	意思決定理由について、明確な根拠やガイドラインを作る時間をマネジメントは割いているか。また、ガイドラインの中に倫理規範は含まれているか。	はい / いいえ
	ビジネスプロセスやミーティングを棚卸しし、不要なもの・従来の用途から離れてしまったものを停止・削除しているか。	はい / いいえ
アンチパターン	ビジネスプロセス全体のボトルネックを計測せずに自動化・効率化を各部門に任せてしまう。	はい / いいえ
	業務自動化全体のアーキテクチャ設計を行わず、部署個別にRPAなどの自動化ツールを導入する。	はい / いいえ
	自動化ツールを前提とした組織設計をおこなわず、既存の業務や組織にあわせてツールをカスタマイズする。	はい / いいえ



デザイン思考のチェック項目



デザイン思考

デザイン思考は、ビジネス合理的なものの考え方から一時的に逃れて、利用者との共感と価値、そして概念的な仮説構築などを行い、大胆な発想をしながら、プロトタイピングやユーザビリティテストを通じて仮説の確からしさを徐々に確かめていくというクリエイティブなアプローチです。

このような過程で、デジタルな事業創出をリードしていくプロダクトマネージャーとUI/UXデザイナーはソフトウェアエンジニア・データサイエンティストと並んで、重要な職能となります。

1	ペルソナの設定
2	顧客体験
3	ユーザーインタビュー
4	デザインシステムの管理
5	デザイン組織
6	プロトタイピング
7	ユーザビリティテスト
8	プロダクトマネジメント

デザイン思考 評価項目 01

ペルソナの設定

製品を使ってくれる顧客の具体的なイメージを共有するためにペルソナという仮想の人格を定義するという手法があります。

ペルソナを用いることで、チームの議論が活発になり、エンジニアやデザイナーの事業理解も進みやすくなり、より高速な開発や価値のある開発が可能になります。

メトリクスの計測	少なくとも1つの大きな事業仮説に対して、対応する1つ以上のペルソナが作成されているか。	はい / いいえ
学習と改善	事業、製品のペルソナについて、データや仮説検証で学習したことを受けて定期的に見直しているか。	はい / いいえ
プラクティスと習慣	ペルソナを記述した資料は、施策会議のたびに意識され、参照されているか。	はい / いいえ
	ペルソナについて、チームで繰り返し議論されイメージの共有化をしているか。	はい / いいえ
	B2Bなど顧客における関係者が複数人いる場合、購買プロセスの各担当者など、意思決定に関わるの人物の数だけ必要なペルソナを作っているか。	はい / いいえ
アンチパターン	ペルソナの具体的なライフストーリーが欠如しており、仮説検証のために必要な打ち手につながらない。	はい / いいえ
	ペルソナが個別的過ぎて、サービスの顧客セグメントへの代表性が薄い	はい / いいえ
	ユーザーインタビューなしに勝手なイメージでペルソナを作っている。	はい / いいえ

デザイン思考 評価項目 02

顧客体験

顧客からの問い合わせに対する効率的なマネジメントは、サブスクリプションや継続率を重視するサービス型の事業にとっては短期の売上以上に経営インパクトのある指標です。

そのため、顧客の行動履歴から重要なインサイトの発見や満足度の向上にどれだけ投資できているかはデジタル化のインジケータになります。

メトリクスの計測	顧客に対してNPS（ネットプロモータースコア）や満足度を継続的に測定しているか。	はい / いいえ
学習と改善	顧客からの問い合わせ返信までのリードタイム、問い合わせおよび回答への満足度について定量計測を行い、目標管理しているか。	はい / いいえ
プラクティスと習慣	構造化されたヘルプページがあり、ヘルプに書かれたことの品質を改善するために顧客がフィードバックすることが可能か。	はい / いいえ
	CRM（顧客管理システム）、SFA（営業支援システム）を導入しており、お問い合わせや行動履歴を把握できているか。	はい / いいえ
	顧客が価値を感じるまでの感情的な動きやチャネルを分析したカスタマージャーニーマップを作成しているか。	はい / いいえ
アンチパターン	カスタマーサポートなど顧客接点となるスタッフから、課題の吸い上げができていない。	はい / いいえ
	電話やメールでの対応に対して、柔軟な対応をしすぎてしまい自動化の阻害要因になっている。	はい / いいえ
	顧客体験の向上のための担当エンジニアリングチームが存在せず、システム化や自動化による改善ができていない。	はい / いいえ

デザイン思考 評価項目 03

ユーザーインタビュー

顧客の隠れたニーズや、課題の感覚を拾い上げて数字になかなか現れない事業仮説を構築するというのは、とても難しい創造的な工程です。

それを補助するためには、専門的な知見と実際のインタビュー接点を持ち続けることは必要不可欠な事業ケイパビリティになります。

メトリクスの計測	ユーザーインタビューによって見つけた課題と、実際にリリースした対策と効果について記録しているか。	はい / いいえ
学習と改善	直近、半年以内になんらかのユーザーインタビューを行っているか。	はい / いいえ
プラクティスと習慣	ユーザーインタビューの実施のための稟議やフローは軽量で、一ヶ月以内に行うことができるか。	はい / いいえ
	インタビュー結果のインサイトをまとめて、共感マップなどを作成しているか。	はい / いいえ
	対象となる潜在顧客がリラックスした状態でインタビューを行うことができるように、専用のインタビュースペースを用意できているか。	はい / いいえ
アンチパターン	ユーザーインタビューに関して訓練を受けていないスタッフが実施しており、仮説に対して誘導的すぎたり、クローズドクエッションが多くなったりしている。	はい / いいえ
	顧客接点になっている部門が、ユーザーインタビューを実施させてくれない。	はい / いいえ
	潜在顧客よりも既存顧客を対象にしたユーザーインタビューに偏ってしまい、潜在的な顧客のペインを捉えられない。	はい / いいえ

デザイン思考 評価項目 04

デザインシステムの管理

UIデザイナーとフロントエンドエンジニアの連携の良さは高速なアプリケーション開発にとって重要な要素です。

また、フロントエンドエンジニアとバックエンドエンジニアの連携のしやすさも課題になります。このようにデザインの反映も自動化していくというサイクルが質を生み出していきます。

メトリクスの計測	デザインシステムを用いて、デザイナーの介在なしにフロントエンド開発の5割以上が達成できている。	はい / いいえ
学習と改善	デザインシステムを、完全に網羅的なものを作ろうとせずに継続的に再利用しメンテナンスしていくことができているか。	はい / いいえ
プラクティスと習慣	プロダクトに関わるUIパーツは、動作するライブラリとしてまとめられているか。	はい / いいえ
	UIデザイナーが作成したデザインをソースコードに自動的に反映するためにツールを利用しているか。	はい / いいえ
	プロダクトに関わるUIパーツには、OOUIなどの抽象的な情報設計に基づいた共通のポリシーがあり、それらがドキュメント化されているか。	はい / いいえ
アンチパターン	フロントエンド開発とバックエンドが密結合しており、APIモックサーバーなどを用いて、バックエンド開発と並行して開発をすすめることができない。	はい / いいえ
	各プラットフォーム（iOS/Android/Web）などで過度に統一の体験を設計しようとしすぎてしまい、それぞれの流儀にそぐわないUIになってしまっている。	はい / いいえ
	各画面やパーツごとに組み込まれた細やかなアニメーションや音、振動、トランジションといった動きのあるUI要素がガイドラインに組み込まれていない。	はい / いいえ

デザイン思考 評価項目 05

デザイン組織

デザイナーは、開発者と同じように、希少な人員になるがゆえには、プロジェクトやサービスの専任ではなく、兼務や時間的・心情的なコミットメントが成約されやすい職種です。

一部の工程だけではなく、全体の工程に溶け合うように関わることで相互理解が生まれ、デザインの専門性を事業に反映しやすくなります。

メトリクスの計測	人数ベースで、内製のUI/UXデザイナーがデザインリソースの過半数を占めているか。	はい / いいえ
学習と改善	デザイン組織のリーダーは、自社戦略に必要なデザイナーの人事戦略として立案しており、採用・育成についての権限と責任を負っているか。	はい / いいえ
プラクティスと習慣	全社のクリエイティブや顧客体験デザインを担う専門知識を持った経営幹部がいるか。	はい / いいえ
	内製のデザイン組織を持ち、内製化すべきスキルと外部の専門家集団を利用するガイドラインを持っているか。	はい / いいえ
	事業責任者は顧客体験やデザイン思考のトレーニングを受けているか。	はい / いいえ
アンチパターン	デザイナーがプロジェクトを横断して派遣され、兼務が多くなったり、関わる期間が短くなるため、各プロダクトにおけるユーザへの共感や事業価値の理解が弱くなっている。	はい / いいえ
	デザイナーが、個別のプロダクトや事業チームごとに配置されるようになり、高速にフィードバックを回すことができるが孤独感やキャリアについて不安を感じている。	はい / いいえ
	デザイナーが事業の意思決定に関わることができるフェーズが遅いため、カスタマージャーニー全体に対する価値が発揮しづらい。	はい / いいえ

デザイン思考 評価項目 06

プロトタイピング

マーケットの不確実性の高い現在では、サービスや製品を完成させてから評価する手法は不向きです。

一度、中核価値の部分だけを雑に作成し、それを評価し直して、再度完成品を作り直すというサイクルが重要になります。

失敗をできるかぎり早めるためにどのような投資をしているかをチェックします。

メトリクスの計測	少なくとも毎月1つ以上の戦略仮説に向けたサービスプロトタイプを作成しているか。	はい / いいえ
学習と改善	実際の製品貢献が小さくとも、経営幹部含めて一年に一度以上の頻度で小さなチームでプロトタイプづくりのワークショップを行っているか。	はい / いいえ
プラクティスと習慣	ある課題の発散のフェーズでは、プロトタイプは極端な仮説に基づいて複数個つくられているか。	はい / いいえ
	デザイナーやプロダクトオーナーはプロトタイピング専用ツールを使うことができるか。	はい / いいえ
	作ったプロトタイプは、一度破棄してから製品用の設計を行っているか。	はい / いいえ
アンチパターン	プロトタイプを作るために、プレゼンや大きな稟議が必要になり、プロトタイピングの前に頓挫することが多い。	はい / いいえ
	プロトタイピングに製品版の完成度を求めてしまう。(プロトタイプはどれだけ雑に仮説検証が達成できるかが重要である。)	はい / いいえ
	一度のプロトタイピングで、中止や製品化を判断してしまう。	はい / いいえ

デザイン思考 評価項目 07

ユーザビリティテスト

顧客の使い勝手の評価を、数値的に裏付けを持って改善することで、思い込みをもっ
てしまいやすいデザインの分野においても
数値による評価を行うことができます。

ユーザー調査と異なり、ニーズの検証では
なく。ある程度動作する製品やプロトタイプ
に対しての使いやすさ・使いごちを評価
していきます。

メトリクスの計測	ユーザビリティに関するパフォーマンステストを行い、継続的または周期的にトラッキングしているか。	はい / いいえ
学習と改善	新規・既存顧客について継続的にユーザビリティの変化がないか示すメトリクス（タスクの成功率など）を計測し、それをもとに改善をこなっているか。	はい / いいえ
プラクティスと習慣	重要機能のタスク成功率/タスク実施時間/学習効率などを計測しているか。	はい / いいえ
	実際のリリース後も入力エラーやタスク時間などを計測しているか。	はい / いいえ
	自己申告メトリクスなどを用いて、印象を含めた満足度のテストをおこなっているか。	はい / いいえ
アンチパターン	ユーザー調査（何が課題かの発見）とユーザビリティテスト（使い心地が良いか、どのような印象を抱いたか）を区別せずに同時に行ってしまう。	はい / いいえ
	事業KPIとの関連の薄い些細な項目ばかりに時間を使ってしまう。	はい / いいえ
	3名以下の少なすぎるユーザビリティテスト対象者の結果に振り回されてしまう。	はい / いいえ

デザイン思考 評価項目 08

プロダクトマネジメント

プロダクトマネージャという役職は、プロジェクトマネージャとは異なり、継続的に顧客と繋がり続けるようなサービスのビジョン、仕様決定、マーケティングまでを行うような専門職です。

事業に応じて、多様な専門性と直感が必要になるため、育成の支援とフォローしあえる体制構築が鍵となります。

メトリクスの計測	権限を持ったプロダクトマネージャという役職が存在し、1サービスに専任の1名以上が任命されているか。	はい / いいえ
学習と改善	幹部人材に対して、プロダクトマネジメントのスキルについての継続的な学習機会を提供できているか。	はい / いいえ
プラクティスと習慣	プロダクトへの要求をユーザーストーリーなど開発チームと合意したバックログアイテムの単位で明文化し、その優先順位をつけることができているか。	はい / いいえ
	事業フェーズに応じて、プロダクトマーケティングや詳細な要求分析、UX評価など適切な分業体制を引くことができているか。	はい / いいえ
	プロダクトビジョンや仮説をバリュープロポジションキャンパスや、仮説キャンパスなどで明文化した上で機能定義を行っているか。	はい / いいえ
アンチパターン	プロダクトマネージャがソフトウェアプロダクト開発やデザインに関する知見や関心が薄く、チームの関係性が悪化している。	はい / いいえ
	プロダクトマネージャとビジネスプロセスやマーケット部門との関係性が薄く、それ起因による実運用によるトラブルが、ほぼ毎回発生している。	はい / いいえ
	プロダクトマネージャという役割に予算執行の権限と責任がない。	はい / いいえ



コーポレートのチェック項目



コーポレート

企業のデジタル化には、デジタル人材にとって活躍しやすく、既存の事業人材との共創とイノベーションを引き起こすための環境が必要不可欠です。

そのため、「風通しの良い組織」「開発者にとっての生産性が高い労働環境」「キャリアや自己実現を促進する仕組みや制度」が重要になります。

これらを推進していくためには経営陣のデジタル理解への努力と人材を巻き込んでいくためのコミットメントが何よりの原動力です。

1	スパン・オブ・コントロール
2	開発者環境投資
3	コミュニケーションツール
4	人事制度・育成戦略
5	デジタル人材採用戦略
6	モダンなITサービスの活用
7	経営のデジタルファースト
8	攻めのセキュリティ

コーポレート評価項目 01

スパン・オブ・コントロール

スパン・オブ・コントロールとは、一人あたりが管理する人数のことです。

平均的にこの人数が多いほど、マネジメントは難しくなり、組織としての統制は取りづらくなります。また、小さすぎても組織が階層的になりすぎてしまいます。

これを適正に保つ方法があるかをチェックします。

メトリクスの計測	スパン・オブ・コントロールの基準を(最低4名-最大10名など)設けており、それに外れた部門数などをモニタリングしているか。	はい / いいえ
学習と改善	兼務及びスパン・オブ・コントロールの基準が適切になるように、定期的に改善が行われているか。	はい / いいえ
プラクティスと習慣	人事制度上、管理職位の設定は人事考課上の等級と独立して設置されるようになっているか。(部長職じゃないと、この給与にならないといったような職位と等級が一致するものでないか。)	はい / いいえ
	どのメンバーにとっても業務的な命令を行う上司は1名か。(その原則が崩れているメンバーを列挙して把握しているか。マトリクス組織であっても指揮系統は1つであるか。)	はい / いいえ
	行っている業務と管理部門の役割が一致するように部門のミッションステートメント(業務分掌)を明確に決めて全社に公開しているか。	はい / いいえ
アンチパターン	人事評価者と指示を行う者の不一致がある組織になっている。または、一致するかの確認ができていない。	はい / いいえ
	部下なしもしくは1名の管理職が存在する。	はい / いいえ
	組織のガイドラインが存在しない、もしくはガイドラインが存在していても例外が常態化している。	はい / いいえ

コーポレート評価項目 02

開発環境投資

ソフトウェア開発者やデザイナーなどの創造的な仕事を行う必要があるメンバーにとって、パソコンおよび周辺機器は重要な仕事道具です。

自由にカスタマイズでき、先端のソフトウェアが快適に動作する環境がなければ、高い生産性は発揮しづらくなります。

メトリクスの計測	プロダクト開発に関係する全ステークホルダーに、自社のIT環境の満足度やeNPS SM を測定しているか。	はい / いいえ
学習と改善	働く環境について、事業競合や採用競合ともコミュニケーションする機会を意識的に持ち、従業員の満足度において常に改善を繰り返しているか。	はい / いいえ
プラクティスと習慣	開発者(およびデザイナー)は、職務遂行に十分なスペックの開発マシンを貸与されているか。(開発マシンは、開発者からのアンケートなどを通じて満足が確認されているか。)	はい / いいえ
	従業員が作成したソフトウェアライブラリを、自社のOSSまたは個人のOSSとして公開するためのガイドラインを準備しており、何らかソフトウェアを公開しているか。	はい / いいえ
	オフィス内のWi-Fi速度は安定して20Mbpsを超えており、人数規模に十分なキャパシティを持っているか。	はい / いいえ
アンチパターン	ソフトウェア開発作業を行う場所で、自由にインターネットを使うことができない。(たとえば、SNSをつかわせないなど)	はい / いいえ
	障害対応など予測の困難な業務や、輪番対応等の計画された定時外業務が合っても、定時に会社に出勤することを求めている。	はい / いいえ
	従業員は自身のパソコンへの自由なソフトウェアのインストールを過度に制限している。	はい / いいえ

コーポレート評価項目 03

コミュニケーションツール

ソフトウェア開発はコミュニケーション効率が悪いと途端に遅くなってしまいます。

そのため、気軽に発信・質問ができたり、作業依頼や完了の記録を残していく仕組みの使いやすさは、とても重要な生産性の基盤になります。

これらが適切に導入されているかをチェックします。

メトリクスの計測	チャットツールにおける全社員が閲覧可能な状態でのコミュニケーションの割合を測定しているか。	はい / いいえ
学習と改善	ここ1年以内に管理職以上で、「コミュニケーションの透明性向上」を目的とした対策を検討し実施したか。	はい / いいえ
プラクティスと習慣	チャットサービスは全社で同一のサービスが導入され、チャットを通じた業務上の手続きの自動化(ChatOps)が可能か。	はい / いいえ
	社内ナレッジの管理のために、大多数の従業員が気軽にwebベースの文書が作成できる、Wikiサービス等を導入しているか。	はい / いいえ
	各部門は、自部門の仕事をメールベースではなくチケットツールベースで受け取る環境があるか。	はい / いいえ
アンチパターン	各種ミーティングにおいて、オンラインでアクセスできる共通の議事録をとっていない。	はい / いいえ
	意思決定者や管理者がチャットツールにログインしておらず、コミュニケーションが取れない。	はい / いいえ
	チャットツールを通じた雑談を禁止している。または雑談をやめるように注意喚起を促したことがある。	はい / いいえ

コーポレート評価項目 04

人事制度・育成戦略

新しい知識やツールを常に学習し続けることが求められるソフトウェアエンジニアにとって、そのための支援と評価制度はこれまで以上に重要になります。

また、転職が当たり前の職種ですので、自社での仕事が他社においてもキャリアになることへの投資に対して強い関心があります。この点について調査します。

メトリクスの計測	理想的な自社従業員のスキルセット構成から逆算した採用・育成計画が中長期の計画として定義されているか。	はい / いいえ
学習と改善	全社員のスキルセットやキャリアを管理しているタレントマネジメントシステムを導入していて、データと計画をアップデートしているか。	はい / いいえ
プラクティスと習慣	プロダクト開発に関わる従業員一人あたり年間12万円(月額1万円)以上の教育研修予算があるか。	はい / いいえ
	専門職向けのジョブ型人事制度があり、管理職と同等かそれ以上の給与で従事しているメンバーが存在するか。	はい / いいえ
	リモートワークやフレックスタイムなどの柔軟な働き方を導入しているか。	はい / いいえ
アンチパターン	高度な専門人材に対して、市場変化を考慮した年収額のアップデートができない人事制度になっている。	はい / いいえ
	自己学習のための書籍や、オンライン学習の補助手当がない。	はい / いいえ
	新入社員向けカリキュラムが、自社内でしか通用しないノウハウ、知識に特化したものになっている。	はい / いいえ

コーポレート評価項目 05

デジタル人材採用戦略

ソフトウェアエンジニアに限らず、デジタル人材の獲得は、デジタル化には必要不可欠です。

しかし、新卒一括採用を重視する企業においては、中途採用の戦略や基礎的な知識に欠けることも珍しくありません。

経営レベルでの採用コミットが、自社の高速なDXを促進することになります。

<p>メトリクスの計測</p>	<p>ソフトウェアエンジニアの採用活動を常に行っており、毎年十分なペースでソフトウェアエンジニアを採用できているか。</p>	<p>はい / いいえ</p>
<p>学習と改善</p>	<p>採用管理システムが導入され、採用に関わる全てのステップ、チャンネルからの情報が遅滞なく不足なく集約されているか。</p>	<p>はい / いいえ</p>
<p>プラクティスと習慣</p>	<p>候補者の方との初回面談から、オファーまでのリードタイムを目標とともに管理しているか。（長すぎる選考プロセスは、人材獲得の妨げになる。）</p>	<p>はい / いいえ</p>
	<p>採用したい人材の基準を明確化したJob Description（求人票）が存在し、現場エンジニアとともに表現を見直すためのワークフローが整備されていて、直近一ヶ月以内に更新されているか。</p>	<p>はい / いいえ</p>
	<p>中途採用におけるリファラル採用（社員からの紹介による採用）の比率が全体採用数の30%を超えているか。</p>	<p>はい / いいえ</p>
<p>アンチパターン</p>	<p>採用部門が行使可能な予算金額が年間採用目標人数の年収合計に対して25%以下しかないまたは予算計画が存在しない。</p>	<p>はい / いいえ</p>
	<p>「募集職種：Webエンジニア」のように曖昧な表現しかされておらず、どのようなスキルやキャリアが求められているのかの解像度が低い。</p>	<p>はい / いいえ</p>
	<p>経営や幹部人材が、人材採用に対して責務を負っておらず十分な時間と熱量を費やしていない。</p>	<p>はい / いいえ</p>

コーポレート評価項目 06

モダンなITサービスの活用

IT人材を戦略的に活用するためには、よりコモディティ化していない領域に集中することが最も重要です。

そのために、外部企業が継続的に改善してくれるであろうSaaSの利用は、非常に合理的な選択肢です。

一時的なコストの問題から、自社のIT人材を浪費していないかをチェックします。

メトリクスの計測	従業員の情報システムへの満足度と利用度、申請から承認などのリードタイムなどの指標を継続的に測定し、改善に生かしているか。	はい / いいえ
学習と改善	SaaS間の自動連携を目的としたサービス(iPaaSやLow Code系ツールなど)を導入しており、業務の自動化や効率化をするためのBPR活動を事業部主体で実行しているか。	はい / いいえ
プラクティスと習慣	業務に合わせて社内のシステム開発を行うのではなく、デファクトなSaaSなどの利用を行い、そのツールに業務自体をフィットさせているか。	はい / いいえ
	表計算やプレゼン資料などはすべてクラウド上で共同編集できるようにになっているか。	はい / いいえ
	自身のスマートフォン、または貸与されたスマートフォンを用いて、会社の予定・メール・コミュニケーションツールなどを利用することができるか。	はい / いいえ
アンチパターン	決裁書類がワークフローツールだけで完結しない。	はい / いいえ
	古いバージョンのOSやブラウザでしか動作しないツールが使われている。	はい / いいえ
	業務で利用しているシステムにシステム間連携を目的としたAPIが用意されていない。	はい / いいえ

コーポレート評価項目 07

技術戦略と内部化

最新の技術動向から事業に必要で実践可能な技術戦略を策定することは、専門的な知見と高度な事業理解なしではできません。

また、それを実行するためのCTOやCDOといった経営レベル権限を持った人材がドライブしていかないと現場は改革が実行できずにどんどん疲弊してしまいます。このための、経営的なケイパビリティを調査します。

メトリクスの計測	データとデジタル技術を用いて、どのように事業変革をしているかのビジョンを明文化して経営メッセージとして発信し、それらの推進についての経営指標をもっているか。	はい / いいえ
学習と改善	ソフトウェアエンジニアとしての業務経験のあるCTOないし技術担当取締役が存在し、技術戦略についての策定を主導的に行っているか。	はい / いいえ
プラクティスと習慣	事業部門内にデジタル技術を活用した経営戦略を担うための担当役員（CDO等）が存在するか。	はい / いいえ
	自社システムの戦略（競争領域・非競争領域の定義）を明確化しており、競争領域のプロダクトの開発を内製人材でコントロールできているか。	はい / いいえ
	デジタル事業および人材獲得に向けたM&A・投資戦略を遂行するための部隊が存在するか。	はい / いいえ
アンチパターン	IT予算の過半をソフトウェア資産として計上し、予算決裁のためにリードタイムの長い（1ヶ月以上かかるような）意思決定フローを挟んでいる。	はい / いいえ
	継続的なシステム改善のためのプロダクトマネジメント経験者がいない。そのため、開発チームとの関係が受発注構造になっている。（情報子会社問題）	はい / いいえ
	競争領域ではないシステムについて、業務フローをSaaSに合わせて変更するのではなく、既存の業務フローに合わせてパッケージソフト等をカスタマイズしている。	はい / いいえ

コーポレート評価項目 08

攻めのセキュリティ

情報セキュリティは、安全性と利便性を同時に実現するには、専門的な知見と最新技術のアップデートが必要不可欠になります。

最新動向を追いかけつつ、従業員の生産性や創造性に寄与する攻めのセキュリティは、イノベーションの源泉になる戦略投資です。

メトリクスの計測	セキュリティ担当者の人事評価における評価基準に事業や従業員の生産性改善が組み込まれているか。	はい / いいえ
学習と改善	社内のセキュリティ担当者が、近年のセキュリティ動向やシフトレフトについて理解し、事業部門とともに開発リードタイムや生産性の改善のために必要な措置をおこなっているか。	はい / いいえ
プラクティスと習慣	インシデントレスポンスチームが、社内の専門家と事業責任者を含むチームにより構成されていて、インシデント時の予行練習をおこなっているか。	はい / いいえ
	境界防御モデルではなく、ゼロトラストモデルのセキュリティネットワークを構築しているか。	はい / いいえ
	セキュリティの意思決定は、定期的に行われているリスクアセスメントにともない、リスクとリターンを定量的に評価した上で行われているか。	はい / いいえ
アンチパターン	管理・監視できない領域のITサービスの利用が行われていること(シャドウIT)に対して対策が打てていない。	はい / いいえ
	"パスワードを定期的な変更を強制する"、"メールでのファイル添付でパスワード付きZIP形式で添付し、別途パスワードを送る"などの現在ではセキュリティ価値が低いとされるルールが残っている。	はい / いいえ
	従業員および経営幹部に対しての最新のトレンドなどが取り入れたセキュリティ教育がなされていない。	はい / いいえ

5. 今後の流れ



今後の流れ

GitHubでの公開

PDFおよび、JSONおよびMarkdown形式での本基準をGitHubに公開いたします。

CTO Community のSlackにて議論をしながら、年1程度でアップデートを続けることを検討中です。最新のトレンドなどを取り入れ、当たり前になったもの/古いものを修正していきます。

診断ツールの公開

これらの項目について定期的に測定、可視化できるツールを公開していきます。

これにより、各社診断を実施しやすくなり活用が進むことを期待しています。

協力各社の情報収集

ご協力いただける各社の情報を収集していき、統計データまたは公開データとして、これらの基準を満たす企業がどれだけあるのかなどデジタル組織のベンチマークとしての活用が進むようにしてまいります。

これらの結果もレポートを通じて発表することができればと思っています。

Appendix: 参考文献



参考文献集：チーム調査項目

エイミー・C・エドモンドソン(著), チームが機能するとはどういうことか : 「学習力」と「実行力」を高める実践アプローチ, 野津智子(訳), 東京, 英治出版, 2014. 5

広木大地(著), エンジニアリング組織論への招待 = Engineering Organization Theory : 不確実性に向き合う思考と組織のリファクタリング, 東京, 技術評論社, 2018. 3

ピーター・M. センゲ(著), 学習する組織: システム思考で未来を創造する, 枝廣淳子(訳), 小田理一郎(訳), 中小路佳代子(訳), 東京, 英治出版, 2011. 6

Mike Cohn(著), アジャイルな見積りと計画づくり : 価値あるソフトウェアを育てる概念と技法, 安井力(), 角谷信太郎(訳), 東京, 毎日コミュニケーションズ, 2009. 1

平鍋健児(著), 野中郁次郎(著), アジャイル開発とスクラム : 顧客・技術・経営をつなぐ協調的ソフトウェア開発マネジメント = Agile and scrum : software development that connects customers, engineers and management, 翔泳社, 2013. 1

アンドリュー・S・グローブ(著), High output management : 人を育て、成果を最大にするマネジメント, 小林薫(訳), 日経BP社, 日経BPマーケティング (発売), 2017. 1

ジョン・ドーア(著), メジャー・ホワット・マターズ : 伝説のベンチャー投資家がGoogleに教えた成功手法OKR, 土方奈美(訳), 東京, 日本経済新聞出版社, 2018. 10

ピョートル・フェリクス・グジバチ(著), 世界最高のチーム : グーグル流「最少の人数」で「最大の成果」を生み出す方法, 東京, 朝日新聞出版, 2018. 8

Brian W. Fitzpatrick(著), Ben Collins-Sussman(著), Team Geek : Googleのギークたちはいかにしてチームを作るのか, 角征典(訳), 東京, オライリー・ジャパン, オーム社, 2013. 7

Dennis M. Ahern(著), Aaron Clouse(著), Richard Turner(著), CMMIモデルガイド, 前田卓雄(訳), 日刊工業新聞社, 2002. 9

Nicole Forsgren(著), Jez Humble, Gene Kim(著), LeanとDevOpsの科学 : テクノロジーの戦略的活用が組織変革を加速する : Accelerate, impress top gear, 武舎広幸(訳), 武舎るみ(訳), 東京, インプレス, 2018. 11

市谷聡啓(著), 正しいものを正しくつくる : プロダクトをつくるとはどういうことなのか、あるいはアジャイルのその先について, 東京, ビー・エヌ・エヌ新社, 2019. 6,

Craig Larman(著), Bas Vodde (著), 大規模スクラム = Large-Scale Scrum (LeSS) : アジャイルとスクラムを大規模に実装する方法, 榎本明仁(監訳), 荒瀬中人(訳), 木村卓央(訳), 高江洲睦(訳), 水野正隆(訳), 守田憲司(訳), 東京, 丸善出版, 2019. 1

大野耐一(著), トヨタ生産方式 : 脱規模の経営をめざして, 東京, ダイヤモンド社, 1978. 5

メアリー・ポッペンディーク(著), トム・ポッペンディーク(著), リーン開発の本質 : ソフトウェア開発に活かす7つの原則, 平鍋健児(監訳), 高嶋優子(訳), 天野勝(訳), 東京, 日経BP社, 日経BP出版センター, 2008. 2

宇治川浩一(著), アジャイルCCPM: プロジェクトのマネジメントを少し変えて組織全体のパフォーマンスを大きくのぼす, 株式会社ビーイング, Amazon Services International, Inc., 2017. 8, 30



参考文献集：システム調査項目

リーダブルコード：より良いコードを書くためのシンプルで実践的なテクニック, Dustin
Boswell(著), Trevor Foucher(著), 角征典(訳), 東京, オライリー・ジャパン, オーム
社, 2012. 6

縣俊貴(著), 良いコードを書く技術：読みやすく保守しやすいプログラミング作法, WEB+DB
press plusシリーズ, 東京, 技術評論社, 2011. 5

Kent Beck(著), テスト駆動開発, 和田卓人(訳), 「テスト駆動開発入門」(ピアソン・エデュ
ケーション 2003年刊)の改題、新訳版, 東京, オーム社, 2017. 10

David Scott Bernstein(著), レガシーコードからの脱却：ソフトウェアの寿命を延ばし価値
を高める9つのプラクティス, 吉羽龍太郎(訳), 永瀬美穂(訳), 原田騎郎(訳), 有野雅士(訳), 東
京, オライリー・ジャパン, オーム社, 2019. 9

Neal Ford, Rebecca Parsons(著), Patrick Kua(著), 進化的アーキテクチャ：絶え間ない変
化を支える, 島田浩二(訳),
東京, オライリー・ジャパン, オーム社, 2018. 8

Robert C. Martin(著), Clean Architecture：達人に学ぶソフトウェアの構造と設計, 角征典
(訳), 高木正弘(訳), 東京, ドワンゴ, KADOKAWA, 2018. 7

Brendan Burns(著), 分散システムデザインパターン：コンテナを使ったスケーラブルなサー
ビスの設計, 松浦隼人(訳), 東京, オライリー・ジャパン, オーム社, 2019. 4

Jez Humble(著), David Farley(著), 継続的デリバリー：信頼できるソフトウェアリリース
のためのビルド・テスト・デプロイメントの自動化, 和智右桂(訳), 高木正弘(訳), 東京, アス
キー・メディアワークス, 角川グループパブリッシング, 2012. 3

Vaughn Vernon(著), 実践ドメイン駆動設計：エリック・エヴァンスが確立した理論を実際の
設計に応用する, 高木正弘(訳), Object Oriented SELECTION, 東京, 翔泳社, 2015. 3

Dino Esposito(著), Andrea Saltarello(著). NETのエンタープライズアプリケーションアー
キテクチャ：.NETを例にしたアプリケーション設計原則, 日本マイクロソフト株式会社(監
訳), クイープ(訳), 原タイトル: MICROSOFT.NET-ARCHITECTING APPLICATIONS FOR THE
ENTERPRISE 原著第2版の翻訳, 東京, 日経BP社, 日経BPマーケティング, 2015. 6

ジーン・キム(著), ジェズ・ハンブル(著), パトリック・ドボア(著), ジョン・ウィリス(著),
The DevOpsハンドブック：理論・原則・実践のすべて, 榊原彰(監修), 長尾高弘
(訳), 東京, 日経BP社, 日経BPマーケティング, 2017. 6

Mike Julian(著), 入門監視：モダンなモニタリングのためのデザインパターン, 松浦隼人
(訳), 東京, オライリー・ジャパン, オーム社, 2019. 1

Betsy Beyer(訳), Chris Jones(訳), Jennifer Petoff(訳), Niall Richard Murphy(編), SRE
サイトリライアビリティエンジニアリング：Googleの信頼性を支えるエンジニアリングチー
ム, 澤田武男(監訳), 関根達夫(監訳), 細川一茂(監訳), 矢吹大輔(監訳), 玉川竜司(訳), 東
京, オライリー・ジャパン, オーム社, 2017. 8

Andy Oram(著), Greg Wilson(編), Making Software：エビデンスが変えるソフトウェア開発,
久野 禎子(訳), 久野 靖(訳), 東京, オライリー・ジャパン, オーム社, 2011. 9

Tom Laszewski(著), Kamal Arora(著), Erik Farr(著), Piyum Zonooz(著), クラウドネイティブ
・アーキテクチャ：可用性と費用対効果を極める次世代設計の原則, impress top gear, トッ
プスタジオ(訳), 東京, 2019. 11

Gustavo Franco, SRE チームの評価に役立つレベル別チェック リスト, 2019 年 1 月 26 日に
Google Cloud blog に投稿されたものの抄
訳, [https://cloud.google.com/blog/ja/products/gcp/how-to-start-and-assess-your-sre-j
ourney](https://cloud.google.com/blog/ja/products/gcp/how-to-start-and-assess-your-sre-journey), 2019. 11. 27



参考文献集：データ駆動調査項目

室脇慶彦(著), IT負債：基幹系システム「2025年の崖」を飛び越えろ, 東京, 日経BP, 日経BPマーケティング, 2019. 6

藤井保文(著), 尾原和啓(著), アフターデジタル = Online Merges with Offline : オフラインのない時代に生き残る, 東京, 日経BP社, 日経BPマーケティング, 2019. 3

Martin Kleppmann(著), データ指向アプリケーションデザイン：信頼性、拡張性、保守性の高い分散システム設計の原理, 斉藤太郎(監訳), 玉川竜司(訳), 東京, オライリー・ジャパン, オーム社, 2019. 7

内山悟志(著), デジタル時代のイノベーション戦略 = THE STRATEGY OF DIGITAL INNOVATION, 東京, 技術評論社, 2019. 6

オリバー・ラッセスバーガー(著), モーハン・ソーニー(著), データ駆動型企業：「自律的意思決定」でビジネスを加速する5つのステージ, 日本テラデータ株式会社(監訳), 杉田真(訳), 東京, 日経BP, 日経BPマーケティング, 2019. 10

スコット・W. アンブラー(著), ピラモド・サダラージ(著), データベース・リファクタリング：データベースの体質改善テクニック, 梅澤真史(訳), 越智典子(訳), 小黒直樹(訳), 東京, ピアソン・エデュケーション, 2008. 4

情報処理推進機構AI白書編集委員会(編), AI白書, 東京, 角川アスキー総合研究所, KADOKAWA(発売), 2018. 1

飯塚健太郎(著), 大川徳之(著), keno(著), 古賀理(著), 田中一樹(著), 徳永拓之(著), 西田圭介(著), 森田和孝(著), Practical Developers：機械学習時代のソフトウェア開発, WEB+DB PRESS plusシリーズ, 東京, 技術評論社, 2019. 8

DataKitchen, The DataOps Manifesto, <https://www.dataopsmanifesto.org/>, 2019, 2019. 11. 27

Tetsu Isomura(著), Risa Nishiyama(著), Yu Ito(著), Machine Learning Project Canvas, 1st edition, Digital Transformation Group Mitsubishi Chemical Holdings Corporation, https://www.mitsubishichem-hd.co.jp/news_release/pdf/190718.pdf, 2019. 1. 27

日経クロストrend(編), 日本ディープラーニング協会(監修), ディープラーニング活用の教科書：先進35社の挑戦から読むAIの未来, 東京, 日経BP社, 日経BPマーケティング, 2018. 10

日経クロストrend(編), 日本ディープラーニング協会(監修), ディープラーニング活用の教科書, 実践編, 東京, 日経BP, 日経BPマーケティング, 2019. 10

山本大祐(著), オプティム(監修), 課題解決とサービス実装のためのAIプロジェクト実践読本：第4次産業革命時代のビジネスと開発の進め方, 東京, マイナビ出版, 2019. 3



参考文献集：デザイン思考調査項目

トム・タリス (著), ビル・アルパート(著), ユーザーエクスペリエンスの測定 : UXメトリクス
の理論と実践, 篠原稔和(翻訳), 情報デザインシリーズ, 東京, 東京電機大学出版局, 2014

樽本徹也(著), ユーザビリティエンジニアリング : ユーザエクスペリエンスのための調査,
設計、評価手法, 第2版, 東京, オーム社, 2014. 2

アラ・コルマトヴァ(著), Design Systems : デジタルプロダクトのためのデザインシステム
実践ガイド, 佐藤伸哉(監訳), Bスプラウト(訳), 東京, ボーンデジタル, 2018. 12

アレックス・オスターワルダー(著), イヴ・ピニューール(著), グレッグ・バーナーダ (著), ア
ラン・スミス(著), バリュー・プロポジション・デザイン : 顧客が欲しが
る製品やサービスを創る, 関美和(訳), 東京, 翔泳社, 2015. 4

及川卓也(著), ソフトウェア・ファースト = SOFTWARE FIRST : あらゆるビジネスを一変させ
る最強戦略, 東京, 日経BP, 日経BPマーケティング, 2019. 10

Gayle Laakmann McDowell(著), Jackie Bavaro(著), 世界で闘うプロダクトマネジャーになる
ための本 : トップIT企業のPMとして就職する方法, 小山香織(訳), 小林啓倫(監訳), 東京, マイ
ナビ, 2014. 8

マーティ・ケーガン(著), INSPIRED : 熱狂させる製品を生み出すプロダクトマネジメント, 原
タイトル: INSPIRED原著第2版の翻訳, 佐藤真治(監訳), 関満徳 (監訳) 神月謙一(訳), 東京, 日
本能率協会マネジメントセンター, 2019. 11

大内孝子(編著), 齊藤滋規(著), 坂本啓(著), 竹田陽子(著), 角征典 (著), エンジニアのための
デザイン思考入門 = INTRODUCTION TO DESIGN THINKING FOR ENGINEERS, 東京, 翔泳
社, 2017. 12

スティーブ・ポーチガル(著), ユーザーインタビューをはじめよう : UXリサーチのための「聞
くこと」入門, 安藤貴子(訳),
東京, ビー・エヌ・エヌ新社, 2017. 6

ピーター・メルホルツ(著), クリスティン・スキナー(著), デザイン組織のつくりかた : デザ
イン思考を駆動させるインハウスチームの構築&運用ガイド, 長谷川敦士(監修), 安藤貴子
(訳), 東京, ビー・エヌ・エヌ新社, 2017. 12

リチャード・ベンフィールド(著), デザインリーダーシップ : デザインリーダーはいかにし
て組織を構築し、成功に導くのか?, 三浦和子(訳), 東京, ビー・エヌ・エヌ新社, 2018. 5

Jeff Patton(著), ユーザーストーリーマッピング, 川口恭伸(監訳), 長尾高弘(訳), 東京, オラ
イリー・ジャパン, オーム社, 2015. 7



参考文献集：コーポレート調査項目

チャールズ・A・オライリー(著), マイケル・L・タッシュマン(著), 両利きの経営：「二兎を追う」戦略が未来を切り拓く, 入山章栄(監訳・解説者), 渡部典子(訳), 東京, 東洋経済新報社, 2019. 2

パティ・マッコード(著), NETFLIXの最強人事戦略：自由と責任の文化を築く, 櫻井祐子(訳), 東京, 光文社2018. 8

成毛眞(著), amazon：世界最先端の戦略がわかる, 東京, ダイヤモンド社, 2018. 8

エリック・シュミット(著), ジョナサン・ローゼンバーグ(著), アラン・イーグル(著), How Google works：私たちの働き方とマネジメント, 土方奈美(訳), 日本経済新聞出版社, 2014. 10

青田努(著), 採用に強い会社は何をしているか：52の事例から読み解く採用の原理原則, 東京, ダイヤモンド社2019. 4

桑田耕太郎(著), 田尾雅夫(著), 組織論, 補訂版, 有斐閣, 2010. 3

トレイシー・メイレット(著), マシュー・ライド(著), エンployee・エクスペリエンス：社員のモチベーションを高める新戦略, 和田美樹(訳), 東京, キノブックス2019. 5

人事と心理編集部(著), 構造化された面接:採用の不一致をなくすアメリカ式の新しい採用面接, 人事と心理シリーズ, 人事と心理編集部(訳), 人事と心理出版, 2019. 8. 15

デジタルトランスフォーメーションに向けた研究会(著), DXデジタルトランスフォーメーションレポート～ITシステム「2025年の崖」の克服とDXの本格的な展開～, 2018. 9. 7, https://www.meti.go.jp/shingikai/mono_info_service/digital_transformation/pdf/20180907_03.pdf, (2019. 12. 3)

Evan Gilman, Doug Barth(著), ゼロトラストネットワーク：境界防御の限界を超えるためのセキュアなシステム設計, 鈴木研吾(監訳), 東京, オライリー・ジャパン, 東京, オーム社, 2019. 10

戸部 良一(著), 寺本 義也(著), 鎌田 伸一(著), 杉之尾 孝生(著), 村井 友秀(著), 野中 郁次郎(著), 失敗の本質：日本軍の組織論的研究, 中公文庫, 東京, 中央公論社, 1991. 8

Mary Jo Hatch(著), Ann L. Cunliffe(著), Hatch組織論：3つのパースペクティブ, 大月博司(訳), 日野健太(訳), 山口善昭(訳), 東京, 同文館出版, 2017. 2

マシュー・サイド(著), 失敗の科学：失敗から学習する組織、学習できない組織, 有枝春(訳), 東京, ディスカヴァー・トゥエンティワン, 2016. 12

ペイカレント・コンサルティング(著), ステップで実現するデジタルトランスフォーメーションの実際, 東京, 日経BP社, 日経BPマーケティング, 2017. 12

Claire Agutter(著), Johann Botha(著), Suzanne D. Van Hove(著), Van Haren Publishing(編集), VeriSM™: Unwrapped and Applied, English Edition, 1版, Van Haren Publishing, 2018. 9. 25