

エンジニアのための ドキュメント ライティング

ユーザーの問題解決と
プロダクトの成功を導く

ジャレッド・バーティ
ザッカリー・サラ・コーライセン
ジェン・ランポーン
デービッド・ヌーニェス
ハイディ・ウォーターハウス 著

岩瀬義昌 訳

エンジニアのための ドキュメントライティング

Forkwell Library #19

開発者の生産性 / プロダクトの成長 / ユーザーの採用率 / を改善する

ドキュメントの作成プロセスを伝授!

- ドキュメントがない
- 書き方がわからない
- 内容がわかりづらい

すぐれたドキュメントが 開発を最適化する!

日本能率協会マネジメントセンター

2023年3月
@iwashi86

An Engineer's Field Guide
to Technical Writing

Jared Bhatti
Zachary Sarah Corleissen
Jen Lambourne
David Nunez
Heidi Waterhouse

Foreword by Kelsey Hightower

Apress®

本日の想定対象者

- エンジニアであれば
「ドキュメントを書いておけば良かった」
と思った経験はあるのでは？
- 一方で
「そもそも、ドキュメントの書き方を体系的に学んだことがない」
という方も多いはず
- 本書はそういう方向けの書籍であり、本イベントでその「一部※ + α 」を紹介
※ 特に訳者が重要だと考えている部分

今日のゴール (イベント終了時)

- ドキュメントライティングのスキルが高まっている

今日のゴール (イベント終了時)

- ドキュメントライティングのスキルが高まっている
 - ドキュメントライティングの概要を知っている
 - ドキュメントの「準備」「作成」「運用」で重要なポイントを理解している

今日のゴール (イベント終了時)

- ドキュメントライティングのスキルが高まっている
 - ドキュメントライティングの概要を知っている
 - ドキュメントの「準備」「作成」「運用」で重要なポイントを理解している
- (書籍でカバーされていないが抑えておくと良い)
日本語文章で大事な点の一部および学びの資源を知っている



- 岩瀬 義昌 (@iwashi86) / Yoshimasa Iwase
- 本業
 - 通信企業でプロダクトマネジメントやアジャイル開発の全社支援
 - その他、組織を強くすることなら何でも
- サイドワーク
 - 翻訳者 (本イベントの書籍)
 - ストックマーク Co-VPoE
 - 非常勤講師
 - ポッドキャスター (fukabori.fm)



fukabori.fm



本題

今日のアジェンダ

- 書籍の概要
- 特に意識しておきたい箇所
 - 読み手の理解（1章）
 - ドラフトの執筆（3章）
 - フィードバックの収集（8章）
 - ドキュメントの品質測定（9章）
- 時間があれば日本語特有の観点（+α）

今日のアジェンダ

- **書籍の概要**

- 特に意識しておきたい箇所

- 読み手の理解（1章）
- ドラフトの執筆（3章）
- フィードバックの収集（8章）
- ドキュメントの品質測定（9章）

- 時間があれば日本語特有の観点（+α）

書籍の構成

PART I

ドキュメント作成の準備

1章 読み手の理解

2章 ドキュメントの計画

書籍の構成

PART I

ドキュメント作成の準備

1章 読み手の理解

2章 ドキュメントの計画

PART II

ドキュメントの作成

3章 ドキュメントのドラフト

4章 ドキュメントの編集

5章 サンプルコードの組み込み

6章 ビジュアルコンテンツの追加

書籍の構成

PART I

ドキュメント作成の準備

1章 読み手の理解

2章 ドキュメントの計画

PART II

ドキュメントの作成

3章 ドキュメントのドラフト

4章 ドキュメントの編集

5章 サンプルコードの組み込み

6章 ビジュアルコンテンツの追加

PART III

ドキュメントの公開と運用

7章 ドキュメントの公開

8章 フィードバックの収集と組み込み

9章 ドキュメントの品質測定

10章 ドキュメントの構成

11章 ドキュメントの保守と非推奨化

書籍の構成

PART I

ドキュメント作成の準備

1章 読み手の理解

2章 ドキュメントの計画

PART II

ドキュメントの作成

3章 ドキュメントのドラフト

4章 ドキュメントの編集

5章 サンプルコードの組み込み

6章 ビジュアルコンテンツの追加

PART III

ドキュメントの公開と運用

7章 ドキュメントの公開

8章 フィードバックの収集と組み込み

9章 ドキュメントの品質測定

10章 ドキュメントの構成

11章 ドキュメントの保守と非推奨化

書籍の構成

PART I

ドキュメント作成の準備

1章 読み手の理解

2章 ドキュメントの計画

PART II

ドキュメントの作成

3章 ドキュメントのドラフト

4章 ドキュメントの編集

5章 サンプルコードの組み込み

6章 ビジュアルコンテンツの追加

PART III

ドキュメントの公開と運用

7章 ドキュメントの公開

8章 フィードバックの収集と組み込み

9章 ドキュメントの品質測定

10章 ドキュメントの構成

11章 ドキュメントの保守と非推奨化

- 1章：読み手であるユーザーを理解する（詳細後述）
- 2章：ユーザーに提供すべきドキュメントタイプ計画する
 - スタートガイド、チュートリアル、APIリファレンス、リリースノート、etc…
 - プロダクト全体におけるユーザージャーニーに沿って作るドキュメントを計画する

書籍の構成

PART I

ドキュメント作成の準備

1章 読み手の理解

2章 ドキュメントの計画

PART II

ドキュメントの作成

3章 ドキュメントのドラフト

4章 ドキュメントの編集

5章 サンプルコードの組み込み

6章 ビジュアルコンテンツの追加

PART III

ドキュメントの公開と運用

7章 ドキュメントの公開

8章 フィードバックの収集と組み込み

9章 ドキュメントの品質測定

10章 ドキュメントの構成

11章 ドキュメントの保守と非推奨化

- 3章：まずはドラフトを書く（後述）
- 4章：編集を通じて精度を高める
 - 例：どんな観点でレビューすべきか？（してもらうべきか？）
- 5-6章：ユーザーの課題をより効果的に解決するために
 - サンプルコードやビジュアルコンテンツ（図表・画像）を追加していく
 - 例：どんな図表が効果的なのか？スクリーンショットはどう使えばいい？

書籍の構成

PART I

ドキュメント作成の準備

1章 読み手の理解

2章 ドキュメントの計画

PART II

ドキュメントの作成

3章 ドキュメントのドラフト

4章 ドキュメントの編集

5章 サンプルコードの組み込み

6章 ビジュアルコンテンツの追加

PART III

ドキュメントの公開と運用

7章 ドキュメントの公開

8章 フィードバックの収集と組み込み

9章 ドキュメントの品質測定

10章 ドキュメントの構成

11章 ドキュメントの保守と非推奨化

- 7-9章：ドキュメントを公開してフィードバックを得て
プロダクトやドキュメントを改善する
- 10章：ドキュメントがスケールしたら（量が増えたら）全体を構成し直す
- 11章：ドキュメントが機能していない or/and 役目を果たしたら非推奨化して廃止する

プロダクト開発と同じ流れ

1 ユーザーを理解して
ジャーニーを描く

2 ジャーニーの中で
ドキュメントが
こう役立つのでは
という仮説を立てる

PART II

ドキュメントの作成

1章 ドキュメントのドラフト

2章 ドキュメントの編集

3章 サンプルコードの組み込み

4章 ビジュアルコンテンツの追加

PART III

ドキュメントの公開と運用

7章 ドキュメントの公開

8章 フィードバックの収集と組み込み

9章 ドキュメントの品質測定

10章 ドキュメントの構成

11章 ドキュメントの保守と非推奨化

プロダクト開発と同じ流れ

ユーザーを理解して
ジャーニーを描く

ジャーニーの中で
ドキュメントが
どう役立つのでは
という仮説を立てる

仮説検証のために実装
(=ドキュメント実装)

設計してコードを書いて
レビューするのと一緒に

PART III

ドキュメントの公開と運用

7章 ドキュメントの公開

8章 フィードバックの収集と組み込み

9章 ドキュメントの品質測定

10章 ドキュメントの構成

11章 ドキュメントの保守と非推奨化

プロダクト開発と同じ流れ

ユーザーを理解して
ジャーニーを描く

ジャーニーの中で
ドキュメントが
こう役立つのでは
という仮説を立てる

仮説検証のために実装
(=ドキュメント実装)

設計してコードを書いて
レビューするのと一緒に

リリースして反応を見る

ユーザーから
フィードバックを集め
プロダクトを改善する

不要な機能※は減らす

※ ドキュメントはプロダクトの機能の1部

プロダクト開発と同じ流れ

ユーザーを理解して
ジャーニーを描く

ジャーニーの中で
ドキュメントが
こう役立つのでは
という仮説を立てる

仮説検証のために実装
(=ドキュメント実装)

設計してコードを書いて
レビューするのと一緒に

リリースして反応を見る

ユーザーから
フィードバックを集め
プロダクトを改善する

不要な機能※は減らす

※ドキュメントはプロダクトの機能の1部

書籍の補足・注意点

- エンジニアがよく作る・利用するドキュメントに集中した書籍

書籍の補足・注意点

- エンジニアがよく作る・利用するドキュメントに集中した書籍
 - そのため、ロジカルライティングや
パラグラフライティングは説明されていない
 - ドキュメントの種類のうち
コンセプトガイド等では
ロジカルシンキングが効果的

書籍の補足・注意点

- エンジニアがよく作る・利用するドキュメントに集中した書籍
 - そのため、ロジカルライティングやパラグラフライティングは説明されていない
 - ドキュメントの種類のうちコンセプトガイド等ではロジカルシンキングが効果的



今日のアジェンダ

- 書籍の概要
- 特に意識しておきたい箇所
 - 読み手の理解（1章）
 - ドラフトの執筆（3章）
 - フィードバックの収集（8章）
 - ドキュメントの品質測定（9章）
- 時間があれば日本語特有の観点（+α）

ドキュメントを書く上で最も大事なこと

- 読み手となるユーザーを理解すること

ドキュメントを書く上で最も大事なこと

- 読み手となるユーザーを理解すること
 - ユーザーは誰なのか？

ドキュメントを書く上で最も大事なこと

- 読み手となるユーザーを理解すること
 - ユーザーは誰なのか？
 - ユーザーは何を達成したいのか？
 - ユーザーはドキュメントが読みたいのではなく
目の前にある課題やニーズを解決したいだけ

ドキュメントを書く上で最も大事なこと

- 読み手となるユーザーを理解すること
 - ユーザーは誰なのか？
 - ユーザーは何を達成したいのか？
 - ユーザーはドキュメントが読みたいのではなく
目の前にある課題やニーズを解決したいだけ
 - ドキュメントにより、どんな課題を解きたいのか？

ドキュメントを書く上で最も大事なこと

- 読み手となるユーザーを理解すること
 - ユーザーは誰なのか？
 - ユーザーは何を達成したいのか？
 - ユーザーはドキュメントが読みたいのではなく
目の前にある課題やニーズを解決したいだけ
 - ドキュメントにより、どんな課題を解きたいのか？
- なぜか？
 - ユーザーの理解を外すと無価値なドキュメントが爆誕する

それ、ビルドトラップの兆候かも？

ビルドトラップの兆候

- ❖ 約束されたロードマップ、長すぎるバックログ
- ❖ 競合他社との機能比較
- ❖ 必要性の分からない機能
- ❖ 生産性やベロシティに対する課題ばかりが表出
- ❖ ステークホルダーの干渉
- ❖ 権限のないプロダクトチーム、意思のないプロダクトチーム
- ❖ 利用状況などのメトリクスを取っていない
- ❖ 役に立たないKPI
- ❖ 使い切らなければいけない予算
- ❖ ……(思いつくものを是非挙げてください!!)

どうすればユーザーを理解できるのか？

どうすればユーザーを理解できるのか？

チャットでの会話ログ

プロダクトの設計メモ

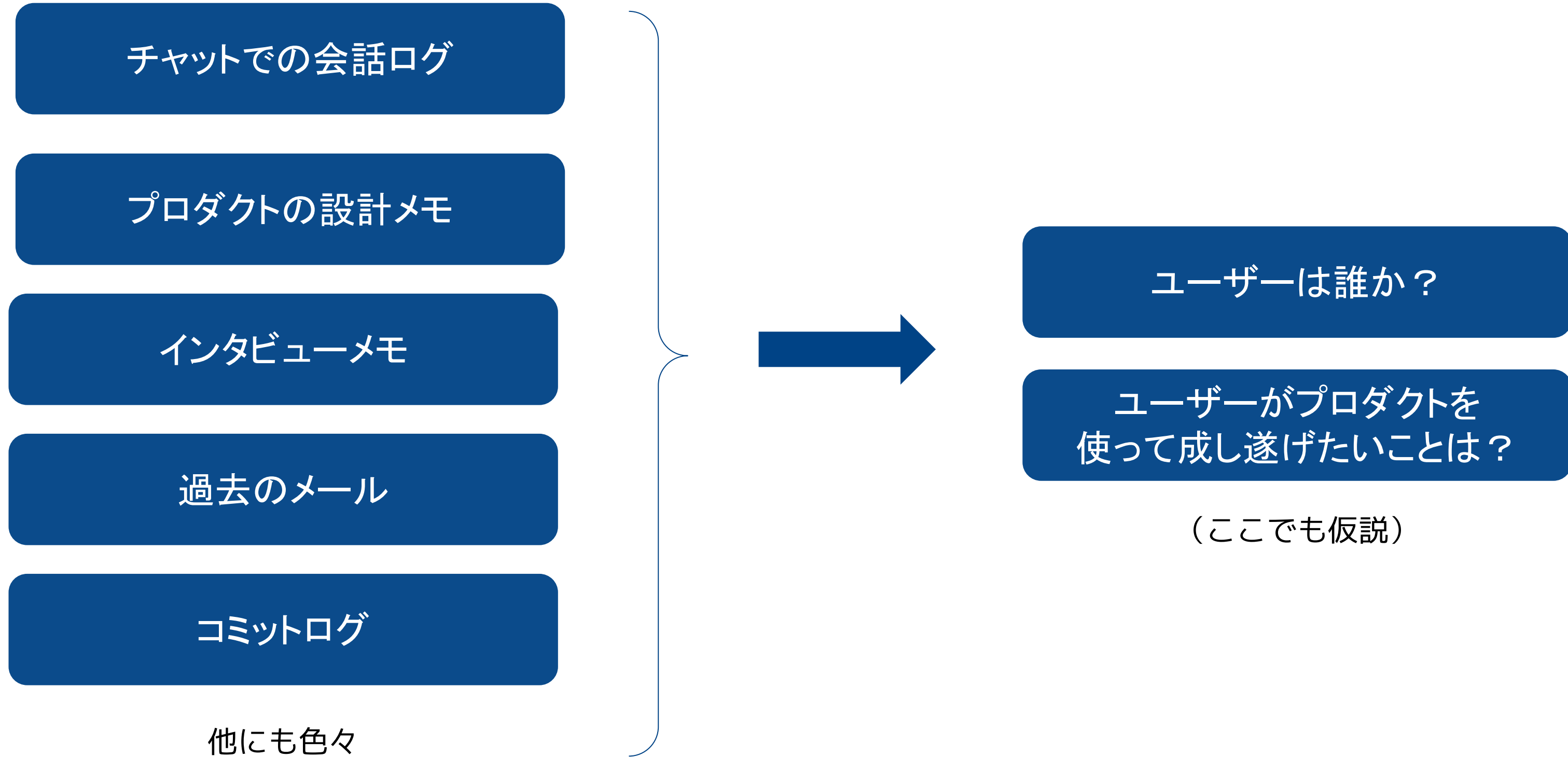
インタビューメモ

過去のメール

コミットログ

他にも色々

どうすればユーザーを理解できるのか？



検証：そのユーザー理解はあっているのか？

検証：そのユーザー理解はあっているのか？

- ユーザーの理解を検証する最強の方法
 - ユーザーと直接対話してみる

検証：そのユーザー理解はあっているのか？

- ユーザーの理解を検証する最強の方法
 - ユーザーと直接対話してみる
- どうやって直接対話するか？
 - 既存チャンネルがあればそれを活用（例：カスタマーサクセス）
 - サポートチケット
 - インタビュー
 - アンケート

対話したら知見をまとめていく

- 代表的なまとめ方
 - ユーザーペルソナ
 - ユーザーストーリー
 - ユーザージャーニーマップ（カスタマージャーニーマップ）

対話したら知見をまとめていく

- 代表的なまとめ方
 - ユーザーペルソナ
 - ユーザーストーリー
 - ユーザージャーニーマップ（カスタマージャーニーマップ）
- なぜまとめておくのか？
 - すぐに忘れてしまうため（覚えておけない）

ユーザーペルソナの例 (chatGPTのペルソナ)



プロダクトマネージャーとして、以下のようなユーザーペルソナを提案します。

1. ユーザーペルソナ：高橋 知恵 (たかはし ちえ)

- 年齢：35歳
- 職業：マーケティングマネージャー
- 居住地：東京都
- 使用デバイス：スマートフォン、ノートパソコン

高橋さんは、中規模の企業でマーケティングマネージャーとして働いています。彼女は、企業のオンラインプレゼンスを向上させ、顧客エンゲージメントを高めるために日々努力しています。そのため、chatgpt を使って、顧客対応や広告戦略の効果を高めたいと考えています。また、プレゼンテーションやレポートの作成にも、より効率的な方法を求めています。

ユーザーストーリーの例



ユーザーストーリー: 高橋 知恵さん (マーケティングマネージャー)

「マーケティングマネージャー」として、「効果的な広告コンテンツ」を作成するために、「chatgptを活用してターゲットオーディエンスへのアピール力を高めるキャッチコピーを生成」したい。

ユーザージャーニーマップの例

	認識・認知	興味・関心	評価・検討	決定・購入	実行・利用	継続利用・推奨
疑問	chatgptはどんな機能があるの？	chatgptは私の業務に役立つの？	chatgptの価格は手頃か？	chatgptは使いやすいか？	chatgptの使い方は？	chatgptは期待通りの性能か？
インタラクション (ユーザー体験)	ウェブサイトや業界ニュースを閲覧	ブログ記事や専門家の意見を読む	chatgptの公式サイトで価格情報を調べる	chatgptの無料トライアルを利用	チュートリアルやヘルプセンターを参照	chatgptを定期的に使い、効果を評価
感情	好奇心が刺激され、興味を持つ	興味が高まるが、実際に役立つか不安	比較的リーズナブルと感じるが、自分にとって価値があるか懸念	使い始めは戸惑いがあるが、興味を持ち続ける	最初は不慣れでストレスを感じるが、徐々に慣れてくる	最初は不安だが、使い続けるうちに満足度が向上する
改善候補	わかりやすいプロダクト紹介ページの改善	より具体的なユースケースの紹介	さまざまなプランの提示	トライアル期間の延長	より詳細な使い方ガイドの提供	フィードバック機能の充実

ユーザージャーニーマップの例

	認識・認知	興味・関心	評価・検討	決定・購入	実行・利用	継続利用・推奨
疑問	chatgptはどんな機能があるの？	chatgptは私の業務に役立つの？	chatgptの価格は手頃か？	chatgptは使いやすいか？	chatgptの使い方は？	chatgptは期待通りの性能か？
インタラクション (ユーザー体験)	ウェブサイトや業界ニュースを閲覧	ブログ記事や専門家の意見を読む	chatgptの公式サイトで価格情報を調べる	chatgptの無料トライアルを利用	チュートリアルやヘルプセンターを参照	chatgptを定期的に使い、効果を評価
感情	好奇心が刺激され、興味を持つ	興味が高まるが、実際に役立つか不安	比較的リーズナブルと感じるが、自分にとって価値があるか懸念	使い始めは戸惑いがあるが、興味を持ち続ける	最初は不慣れでストレスを感じるが、徐々に慣れてくる	最初は不安だが、使い続けるうちに満足度が向上する
改善候補	わかりやすいプロダクト紹介ページの改善	より具体的なユースケースの紹介	さまざまなプランの提示	トライアル期間の延長	より詳細な使い方ガイドの提供	フィードバック機能の充実

感情の起伏
ラインを
書くことも

知識の呪い (認知バイアス)

知識の呪い (認知バイアス)

- こんな経験は？
 - 同僚が、耳慣れない専門用語で話している
 - 同僚が、開発環境を作る手順を教えてくれなかった
 - デバッグでエラーメッセージを教えてくれたものの、他に必要な背景情報は伝えてくれなかった

知識の呪い（認知バイアス）

- こんな経験は？
 - 同僚が、耳慣れない専門用語で話している
 - 同僚が、開発環境を作る手順を教えてくれなかった
 - デバッグでエラーメッセージを教えてくれたものの、他に必要な背景情報は伝えてくれなかった
- 悪気があったのではなく、
おそらく知っていると思っていて伝えていない（可能性が高い）

知識の呪いをどう断ち切るか？

知識の呪いをどう断ち切るか？

- ユーザーへの共感が必要
 - そのためにユーザーの理解をしてきた

知識の呪いをどう断ち切るか？

- ユーザーへの共感が必要
 - そのためにユーザーの理解をしてきた
- フリクションログを作る
 - フリクション = 原義は「摩擦」や「抵抗」
 - 転じて、プロダクトを使う際に
 - 上手くつかえなかったことや違和感などを示す
 - フリクションログはそれを記録したもの

フリクションログの例

ステップ	フリクション	影響
1	サインアップページが見つげにくい	高橋さんは迷ってしまい、サインアップへの興味が減退する
2	メールアドレス入力時に誤りがあってもエラーメッセージが表示されない	高橋さんは何度も試行し、イライラが募る
3	パスワードの要件が明確でない	高橋さんは適切なパスワードを作成できず、繰り返し入力することになる
4	利用不可能な文字列をパスワードに入れてしまったが、エラーメッセージが表示されない	高橋さんはどのような問題があるのか理解できず、ストレスが高まる
5	サインアップ時のキャプチャ認証が複雑で繰り返し解かされる	高橋さんはイライラし、サインアップの途中でやめたくなる
6	サインアップ完了後のメール確認が遅い	高橋さんは不安になり、問題が発生したのかと思い始める

書籍の補足・注意点

- 1章を通読するとストーリーや具体例などから

「顧客へプロダクトを提供するときのドキュメントのみが対象？」

「開発者向けのAPIの話？」

と思うことがある



書籍の補足・注意点

- 1章を通読するとストーリーや具体例などから
「顧客へプロダクトを提供するときのドキュメントのみが対象？」
「開発者向けのAPIの話？」
と思うことがある
- 実際にはほとんどのドキュメントで応用可能
 - 例えば「開発環境の構築手順書」であるなら
「顧客(読み手) = 社内の開発者」となる



今日のアジェンダ

- 書籍の概要
- 特に意識しておきたい箇所
 - 読み手の理解（1章）
 - ドラフトの執筆（3章）
 - フィードバックの収集と品質測定（8-9章）
- 時間があれば日本語特有の観点（+α）

ドキュメントを書く上で最も難しいこととは？

- 白紙から脱却すること = 書き始めること

ドキュメントを書く上で最も難しいこととは？

- 白紙から脱却すること = 書き始めること
- 手を進めるコツ
 - 手に馴染んでいるツールを使うこと
ドキュメント専用に新規に何かを学ばなくても良い
 - 紙とペンでも、ホワイトボードでも何でも良い

ドキュメントを書く上で最も難しいこととは？

- 白紙か
- 手を進
- 手は
- ドキ
- 紙と



https://www.youtube.com/watch?v=JV3KOJ_Z4Vs

余談:書きやすくするためのスキルはありつつも、桜井さんの動画のとおり「いいからやる」のも重要だと思います。

ドラフトを書く上で最初に考えるべきこと

- 冒頭で以下を整理する
 - 「誰が」「何のために」読みにくるのか？
 - 「どのタイプ」のドキュメントが適切か？

ドラフトを書く上で最初に考えるべきこと

- 冒頭で以下を整理する
 - 「誰が」「何のために」読みにくるのか？
 - 「どのタイプ」のドキュメントが適切か？
- 次にタイトルとアウトラインを決める
 - タイトル = ドキュメントを読んで達成できるゴールの要約
 - 例：カスタム会話型チャットボットの開発 など
 - アウトライン = ゴールに必要な大まかな要素や手順
 - 例：開発環境、API認証の方法 など

ユーザーと目的を整理したら肉付けする

- 肉付けに使える要素
 - 見出し
 - 段落
 - リスト
 - コールアウト

ユーザーと目的を整理したら肉付けする

- 肉付けに使える要素
 - 見出し
 - ドキュメント内の道しるべとして機能
 - ユーザーに最も重要な情報を一番上に
 - 段落
 - リスト
 - コールアウト

ユーザーと目的を整理したら肉付けする

- 肉付けに使える要素
 - 見出し
 - 段落
 - ドキュメントの詳細の理解に役立つ文章
 - もっとも情報量が多いが、もっとも読むのが大変
 - リスト
 - コールアウト

ユーザーと目的を整理したら肉付けする

- 肉付けに使える要素
 - 見出し
 - 段落
 - リスト
 - さっと読みやすい要素
 - 順不同だが、重要順やアルファベット順でソートすると読みやすい
 - コールアウト

ユーザーと目的を整理したら肉付けする

- 肉付けに使える要素
 - 見出し
 - 段落
 - リスト
 - コールアウト（例）

⚠ Warning: You might not see **Start iOS Simulator** option when you click **No Devices** in VS Code. If you're on Mac, then you might have to run following command in terminal to launch a simulator.

```
$ open -a simulator
```

On Windows or Linux, it's not possible to launch an iOS simulator.

ドキュメントにまつわる真実

- ユーザーは情報を探してドキュメントにたどり着く



- ユーザーは書いてある内容をほとんど読まない

ユーザーはFパターンでドキュメントを読む



Eyetracking by Nielsen Norman Group nngroup.com NN/g

何がユーザーにとって大事か？

- ~~せっかく書いたから読んで欲しい~~
- ユーザーが必要な情報をすばやく見つけられるのが一番良い
 - そのために「流し読み」しやすい構成にする

どうすれば流し読みしやすくなるか？

- 最も重要な情報を冒頭で述べる
 - たとえば手順書であれば、手順書完了時点で達成できることを書く

We want your thoughts on implementing deep links in Flutter.
By taking part in this study, you'll directly impact Flutter's deep linking support.
[Sign up for a chance to be included.](#)

Get started ▾

Samples & tutorials ▾

Development ▲

▾ User interface

Introduction to widgets

▾ Building layouts

Layouts in Flutter

TutorialCreating adaptive and
responsive apps

Building adaptive apps

Understanding
constraints

Box constraints

Adding interactivity

Assets and images

Material Design

▶ Navigation & routing

▶ Animations

▶ Advanced UI

Widget catalog

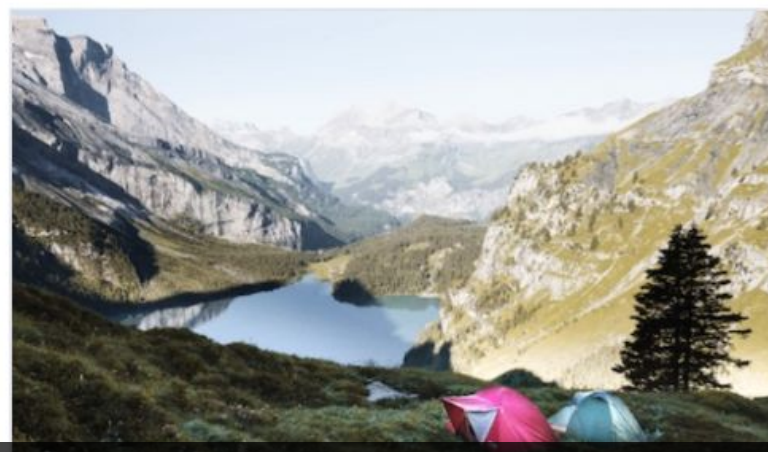
Building layouts

[UI](#) > [Layout](#) > Tutorial

What you'll learn

- How Flutter's layout mechanism works.
- How to lay out widgets vertically and horizontally.
- How to build a Flutter layout.

This is a guide to building layouts in Flutter. You'll build the layout for the following app:



Contents

[Step 0: Create the app base code](#)[Step 1: Diagram the layout](#)[Step 2: Implement the title row](#)[Step 3: Implement the button row](#)[Step 4: Implement the text section](#)[Step 5: Implement the image section](#)[Step 6: Final touch](#)

冒頭で達成できることが書いてある

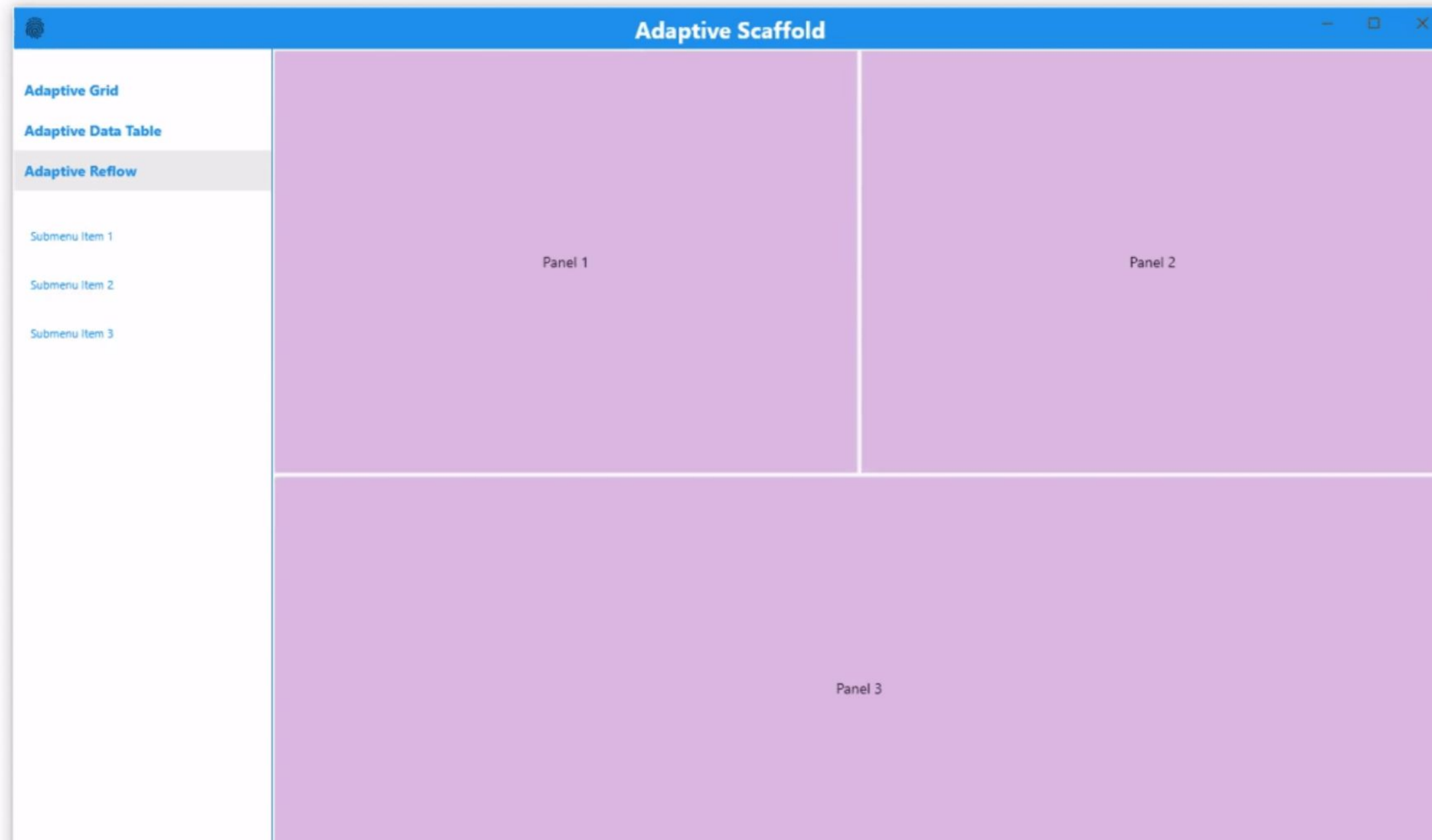
どうすれば流し読みしやすくなるか？

- 最も重要な情報を冒頭で述べる
 - たとえば手順書であれば、手順書完了時点で達成できることを書く
- 大きな文章のかたまりを分割する
 - 長い段落は流し読みが困難
 - 見出し・リスト・サンプルコード・図表 で分割する

Visual density

Different input devices offer various levels of precision, which necessitate differently sized hit areas. Flutter's `VisualDensity` class makes it easy to adjust the density of your views across the entire application, for example, by making a button larger (and therefore easier to tap) on a touch device.

When you change the `VisualDensity` for your `MaterialApp`, `MaterialComponents` that support it animate their densities to match. By default, both horizontal and vertical densities are set to 0.0, but you can set the densities to any negative or positive value that you want. By switching between different densities, you can easily adjust your UI:



段落と図表(実際は動画)で分割している例

To set a custom visual density, inject the density into your `MaterialApp` theme:

<https://docs.flutter.dev/development/ui/layout/building-adaptive-apps>

どうすれば流し読みしやすくなるか？

- 最も重要な情報を冒頭で述べる
 - たとえば手順書であれば、手順書完了時点で達成できることを書く
- 大きな文章のかたまりを分割する
 - 長い段落は流し読みが困難
 - 見出し・リスト・サンプルコード・図表 で分割する
- 複数の方法が含まれているなら方法ごとに分割する
 - 例：GUIを使うパターン、CUIを使うパターン

どうすれば流し読みしやすくなるか？

- 最も重要な情報を冒頭で述べる
 - たとえば手順書であれば、手順書完了時点で達成できることを書く
- 大きな文章のかたまりを分割する
 - 長い段落は流し読みが困難
 - 見出し・リスト・**サンプルコード**・図表 で分割する

(超重要。5章まるまるサンプルコードのみ。詳細は書籍にて)
- 複数の方法が含まれているなら方法ごとに分割する
 - 例：GUIを使うパターン、CUIを使うパターン

今日のアジェンダ

- 書籍の概要
- 特に意識しておきたい箇所
 - 読み手の理解（1章）
 - ドラフトの執筆（3章）
 - フィードバックの収集（8章）
 - ドキュメントの品質測定（9章）
- 時間があれば日本語特有の観点（+α）

ドキュメントも仮説検証対象

- プロダクトに開発した機能がユーザーに刺さるとは限らない
 - 分からないので仮説検証する

ドキュメントも仮説検証対象

- プロダクトに開発した機能がユーザーに刺さるとは限らない
 - 分からないので仮説検証する
- ドキュメントは、プロダクトの機能の一部
 - 他機能と同様に「フィードバック(FB)」をベースに検証が必要
 - FBをどうやって集めればいい？

ユーザーFBの集め方

- フィードバックを受け取るための経路を用意する
- 例
 - ドキュメントのページへformなどを埋め込む

ユーザーFBの集め方

- フィードバックを受け取るための経路を用意する
- 例
 - ドキュメントのページへformなどを埋め込む
 - サポートチケットから読み取る

ユーザーFBの集め方

- フィードバックを受け取るための経路を用意する
- 例
 - ドキュメントのページへformなどを埋め込む
 - サポートチケットから読み取る
 - ドキュメントに対する感情を集める

このページはお役に立ちましたか? *

👍 はい

👎 いいえ

詳しくお聞かせください。

お客様の経験を共有してください。ただし、機密情報や個人情報は含めないでください。

送信

[プライバシー ポリシー](#)

ユーザーFBの集め方

- フィードバックを受け取るための経路を用意する
- 例
 - ドキュメントのページへformなどを埋め込む
 - サポートチケットから読み取る
 - ドキュメントに対する感情を集める
 - ユーザーサーベイを実施する

ユーザーFBの集め方

- フィードバックを受け取るための経路を用意する
- 例
 - ドキュメントのページへformなどを埋め込む
 - サポートチケットから読み取る
 - ドキュメントに対する感情を集める
 - ユーザーサーベイを実施する
 - ユーザーコミュニティ(会)を作り、そこで質問する

ユーザーFBの活用方法

- 集め始めるとFBが大量※に蓄積される
 - Quick Fixできるものから、詳細な検討が必要なものまで色々
(※ FBに見せかけたサポート依頼や、バグ報告まで最初はなんでも混ざっている)

ユーザーFBの活用方法

- 集め始めるとFBが大量に蓄積される
 - Quick Fixできるものから、詳細な検討が必要なものまで色々
- そこで、そもそも有効なFBかどうか、また優先度をトリアージする
 - その課題は有効か？
 - その課題は修正可能か？（重複してない？再現可能？ など）
 - どのぐらい重要か？

ユーザーFBの活用方法

- 集め始めるとFBが大量に蓄積される
 - Quick Fixできるものから、詳細な検討が必要なものまで色々
- そこで、そもそも有効なFBかどうか、また優先度をトリアージする
 - その課題は有効か？
 - その課題は修正可能か？（重複してない？再現可能？ など）
 - どのぐらい重要か？

→ kubernetes の例で見てください！



Search this site...

Documentation

Getting Started

Making your First Contribution

Contributing to Kubernetes

Pull Request Process

GitHub Workflow

Coding Conventions

Help Wanted and Good First Issue Labels

Issue Triage Guidelines

Non-code

Contributions

Adding Release Notes

OWNERS Files

Community

Expectations

Contributor Cheatsheet

New Contributor Course

Communication Platform Guidelines and Policies

Issue Triage Guidelines

These guidelines serve as a primary document for triaging incoming issues to Kubernetes. SIGs and projects are encouraged to use this guidance as a starting point, and customize to address specific triaging needs.

Table of Contents

- [Scope](#)
- [What Is Triaging?](#)
- [Why Is Triaging Beneficial?](#)
- [How to Triage: A Step-by-Step Flow](#)
 - [Triage-Related Tools](#)
 - [Permissions and the Bot](#)
 - [Gubernator](#)
 - [Triage Party](#)
 - [GitHub Project Boards](#)
 - [DevStats](#)
 - [Process Pointers and Advice from SIGs](#)
 - [Running a Triage Meeting: Tips from api-machinery](#)
 - [Triage Guide by cluster-lifecycle](#)
- [Step One: Review Newly Created Open Issues](#)
 - [Conducting Searches](#)
- [Step Two: Triage Issues by Type](#)
 - [Support Requests](#)
 - [Abandoned or Wrongly Placed Issues](#)
 - [Needs More Information](#)

← 次ページでこの辺を拡大します

- Table of Contents
- Scope
- What Is Triaging?
- Why Is Triaging Beneficial?
- Triage-Related Tools
 - Permissions and the Bot
 - Gubernator
 - Triage Party
 - GitHub Project Boards
 - DevStats
- Process Pointers and Advice from SIGs
 - Running a Triage Meeting: Tips from api-machinery
 - Triage Guide by cluster-lifecycle
- Step One: Review Newly Created Open Issues
 - Conducting Searches
- Step Two: Triage Issues by Type
 - Support Requests
 - Abandoned or Wrongly Placed Issues
 - Needs More Information

- Step One: Review Newly Created Open Issues
 - Conducting Searches
- Step Two: Triage Issues by Type
 - Support Requests
 - Abandoned or Wrongly Placed Issues
 - Needs More Information
 - Bugs
 - Help Wanted/Good First Issues
 - Kind Labels
- Step Three: Define Priority

新規Issueの確認

Issueを分類

優先度付け

Priority label	What it means	Examples
<code>priority/critical-urgent</code>	Team leaders are responsible for making sure that these issues (in their area) are being actively worked on—i.e., drop what you're doing. Stuff is burning. These should be fixed before the next release.	user-visible bugs in core features broken builds tests and critical security issues
<code>priority/important-soon</code>	Must be staffed and worked on either currently or very soon—ideally in time for the next release. Important, but wouldn't block a release.	[XXXX]
<code>priority/important-longterm</code>	Important over the long term, but may not be currently staffed and/or may require multiple releases to complete. Wouldn't block a release.	[XXXX]
<code>priority/backlog</code>	General agreement that this is a nice-to-have, but no one's available to work on it anytime soon. Community contributions would be most welcome in the meantime, though it might take a while to get them reviewed if reviewers are fully occupied with higher-priority issues—for example, immediately before a release.	[XXXX]
<code>priority/awaiting-more-evidence</code>	Possibly useful, but not yet enough support to actually get it done.	Mostly placeholders for potentially good ideas, so that they don't get completely forgotten, and can be referenced or deduped every time they come up

Priority label	What it means	Examples
priority/critical-urgent	Team leaders are responsible for making sure that these issues (in their area) are being actively worked on—i.e., drop what you're doing. Stuff is burning. These should be fixed before the next release.	user-visible bugs in core features broken builds tests and critical security issues
priority/important-soon	Must be worked on, but not necessarily as soon—ideally in time for the next release. Important, but wouldn't block a release.	[XXXX]
priority/important-longterm	Important over the long term, but may not be currently staffed and/or may require multiple releases to complete. Wouldn't block a release.	[XXXX]
priority/backlog	General agreement that this is a nice-to-have, but no one's available to work on it anytime soon. Community contributions would be most welcome in the future. May be difficult to get them reviewed if reviewers are fully occupied with higher-priority issues—for example, immediately before a release.	[XXXX]
priority/awaiting-more-evidence	Possibly useful, but not yet enough support to actually get it done.	Mostly placeholders for potentially good ideas, so that they don't get completely forgotten, and can be referenced or deduped every time they come up

超重要

なるべく早くやる。できれば次のリリースで。

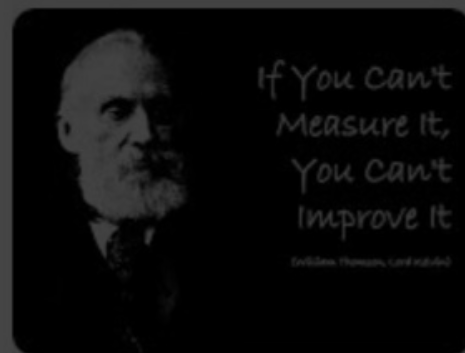
長期的には重要

あったらいいけどすぐやらない

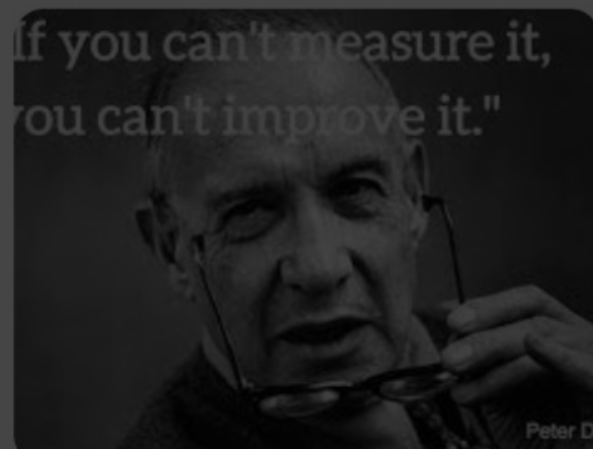
たぶん有用だけどまだ十分な支持がない

今日のアジェンダ

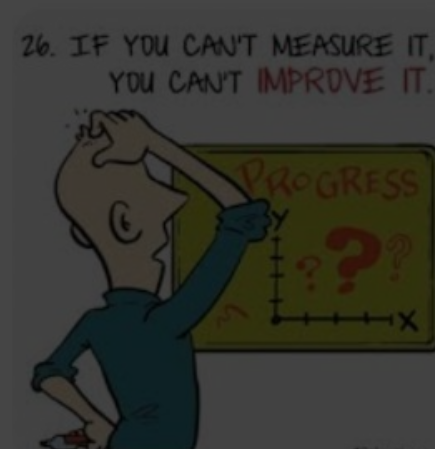
- 書籍の概要
- 特に意識しておきたい箇所
 - 読み手の理解（1章）
 - ドラフトの執筆（3章）
 - フィードバックの収集（8章）
 - ドキュメントの品質測定（9章）
- 時間があれば日本語特有の観点（+α）



LinkedIn What do Lord Kelvin and Peter Drucker have in common?



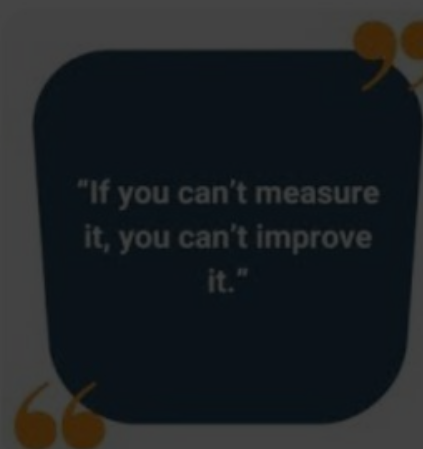
AC Mentoring If you can't measure it, you can't impro...



Facebook Peter Diamandis - If you can't...



LinkedIn You can't improve what you can't measure

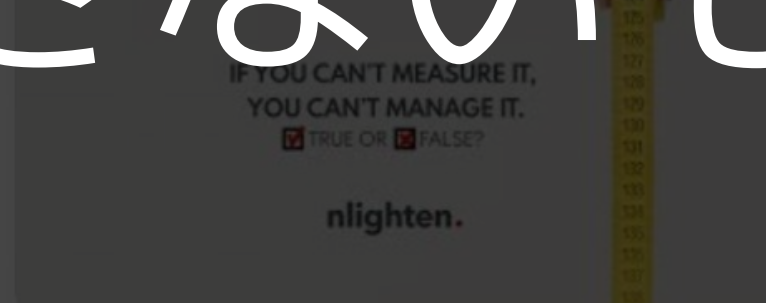


gt Growththink The Two Most Important Quo...

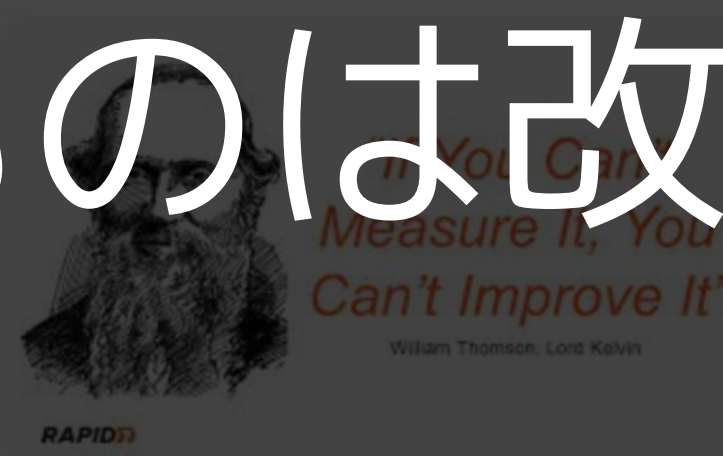
計測できないものは改善できない



Quotefancy Ed Seykota Quote: "If you can't measure it, yo...



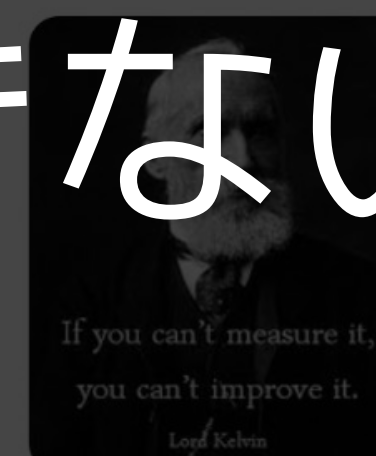
n. nlightencx True or false: If you can't measure it, you can't man...



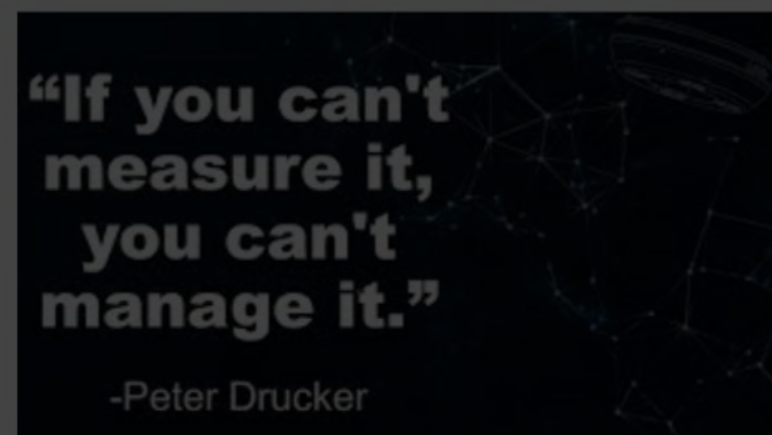
AC Mentoring If you can't measure it, you can't improve it - AC Me...



www.gapingvoidart.com If You Can't Measure It - gapingvoid ...



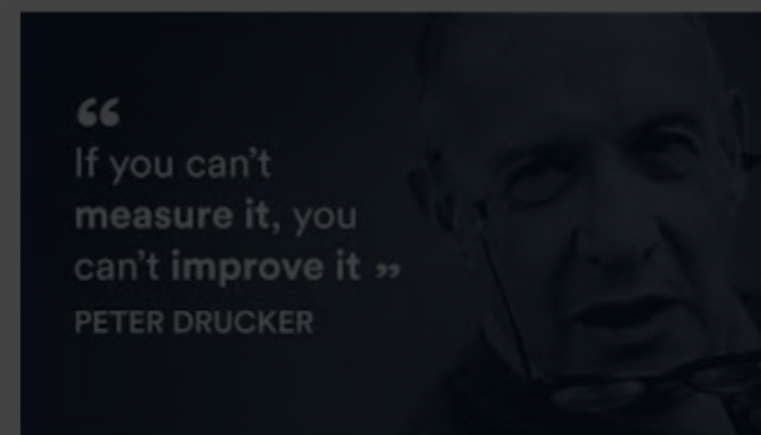
Twitter Lorena Torres, Ph.Dさん...



Coherix 'If You Can't Measure It, You Can't Manage It' - Coherix



Facebook Pasquesi Partners LLC - "If you can't measure it, you ca...



Data-driven VC Data-driven VC #23: How to measure productivity and i...



GuavaBox If You Can't Measure It, You Can't Improve It | Les...

優れたドキュメントって何だろう？

- ドキュメントの品質の定義

「ドキュメントが優れているのは目的にかなっている場合である。」

優れたドキュメントって何だろう？

- ドキュメントの品質の定義

「ドキュメントが優れているのは目的にかなっている場合である。」



ユーザーの特定の行動を促進すること + 組織のゴールを達成すること

ドキュメントの品質の構成要素

- 機能品質
 - ドキュメントの目的やゴールが達成されているか？

ドキュメントの品質の構成要素

- 機能品質
 - ドキュメントの目的やゴールが達成されているか？
- 構造品質
 - ドキュメント自体がうまく書かれているか？
 - うまく構成されているか？

機能品質

- アクセシビリティがあること
- 目的があること
- 見つけやすいこと
- 正確なこと
- 完全であること

構造品質 (3C)

- Clear (明確な)
- Concise (簡潔な)
- Consistent (一貫している)

ドキュメントの品質の構成要素

- 機能品質

- ドキュメントの目的やゴールが達成されているか？

- 構造品質

- ドキュメント自体がうまく書かれているか？
- うまく構成されているか？

さて、どちらの品質が重要でしょうか？🤔

ドキュメントの品質の構成要素

- 機能品質

- ドキュメントの目的やゴールが達成されているか？

- 構造品質

- ドキュメント自体がうまく書かれているか？
- うまく構成されているか？

さて、どちらの品質が重要でしょうか？🤔

→ 機能品質 ∵ どれだけ最高の表現でもゴール達成しないと無価値

ドキュメントのメトリクス

- Web解析ツールを使えば様々なメトリクスを測定可能
 - PV数
 - ユニークユーザー数
 - 直帰率
 - TTHW (Time to Hello World)
 - etc...

ドキュメントのメトリクス

- Web解析ツールを使えば様々なメトリクスを測定可能
 - PV数
 - ユニークユーザー数
 - 直帰率
 - TTHW (Time to Hello World)
 - etc...
- 使い切れないほどあるため、確認するメトリクスの絞り込みが重要

メトリクス活用のTips (の一部)

- メトリクス活用の計画を作る
 - なぜ測定したいのか？
 - その情報を使って何をするのか？
 - その努力で、どのように組織のゴールが前に進められるのか？

メトリクス活用のTips (の一部)

- メトリクス活用の計画を作る
 - なぜ測定したいのか？
 - その情報を使って何をするのか？
 - その努力で、どのように組織のゴールが前に進められるのか？
- 基準値を確立する
 - ベースラインがあれば何らかの変更の前後で比較可能になる
(=計測して基準値があるから改善できるようになる)

今日のアジェンダ

- 書籍の概要
- 特に意識しておきたい箇所
 - 読み手の理解（1章）
 - ドラフトの執筆（3章）
 - フィードバックの収集（8章）
 - ドキュメントの品質測定（9章）
- 時間があれば日本語特有の観点（+α）

日本語の作文技術に関する書籍は多く存在



タイトルが「ヘルプサイトの作り方」であるが作文技術が多く言及されている

事実と意見をはっきり区別する

- 手順書などのテクニカルドキュメントに意見は紛れこみにくい
- 一方で設計の思想が現れるドキュメントは主観が入る
 - ここで事実と意見が混ざると理解しやすさ・信憑性が低下する



事実と意見が曖昧な例

“近頃の学生は整った文章を書く能力がないという声をよく聞くが、私はこれは主に理科系の学生に関していわれていることだと思う。

理科系の学生がきちんとした文章を書けないことにふしぎはない。

彼らの本領は文学ではないからである。”



事実と意見が曖昧な例

“近頃の学生は整った文章を書く能力がないという
声をよく聞くが、私はこれは主に理科系の学生に関
していわれていることだと思う。

} 意見

理科系の学生がきちんとした文章を書けないことに
ふしぎはない。

彼らの本領は文学ではないからである。”



事実と意見が曖昧な例

“近頃の学生は整った文章を書く能力がないという声をよく聞かすが、私はこれは主に理科系の学生に関していわれていることだと思う。

意見

理科系の学生がきちんとした文章を書けないことにふしぎはない。

突然、事実になったため議論の土台がグラグラに

彼らの本領は文学ではないからである。”



中公新書
624

修飾の順序

- 例：以下の紙を「言葉」で表現する



白い

厚手

横線がある

修飾の順序

- 節（述語を含む）を先に、句を後に

白い横線の引かれた厚手の紙

厚手の横線の引かれた白い紙

横線の引かれた厚手の白い紙

わかりやすい文はどれ？

修飾の順序

- 節（述語を含む）を先に、句を後に
 - ✗ 白い横線の引かれた厚手の紙 （＝横線が白い ことになる）
 - ✗ 厚手の横線の引かれた白い紙 （＝横線が厚手 に捉えられる）
 - ✓ 横線の引かれた厚手の白い紙
述語がある

修飾の順序

- 節（述語を含む）を先に、句を後に

✗ 白い横線の引かれた厚手の紙 （＝横線が白い ことになる）

✗ 厚手の横線の引かれた白い紙 （＝横線が厚手 に捉えられる）

✓ 横線の引かれた厚手の白い紙

- その他の原則

- 長い修飾句ほど先に、短いほどあとに
- 大状況から小状況へ、重大なものから重大でないものへ
- 親和度（なじみ）の強弱による配置転換

動詞に「方法」「こと」などを付けて安易に名詞化しない

- 冗長な、くどい日本語になるため



動詞に「方法」「こと」などを付けて安易に名詞化しない

- 冗長な、くどい日本語になるため
- 例：
 - ✗ 機能Aを使用するという方法もあります
 - ✓ 機能Aも使用できます



動詞に「方法」「こと」などを付けて安易に名詞化しない

- 冗長な、くどい日本語になるため
- 例：
 - ✗ 機能Aを使用するという方法もあります
 - ✓ 機能Aも使用できます
 - ✗ Bにより改善することができます
 - ✓ Bにより改善できます



と、いくつかの書籍から説明したように日本語の作文技術も学んでおくのがおすすめです

ということでは

今日お話ししたこと

- 書籍の概要
- 特に意識しておきたい箇所
 - 読み手の理解（1章）
 - ドラフトの執筆（3章）
 - フィードバックの収集（8章）
 - ドキュメントの品質測定（9章）
- 時間があれば日本語特有の観点（+α）

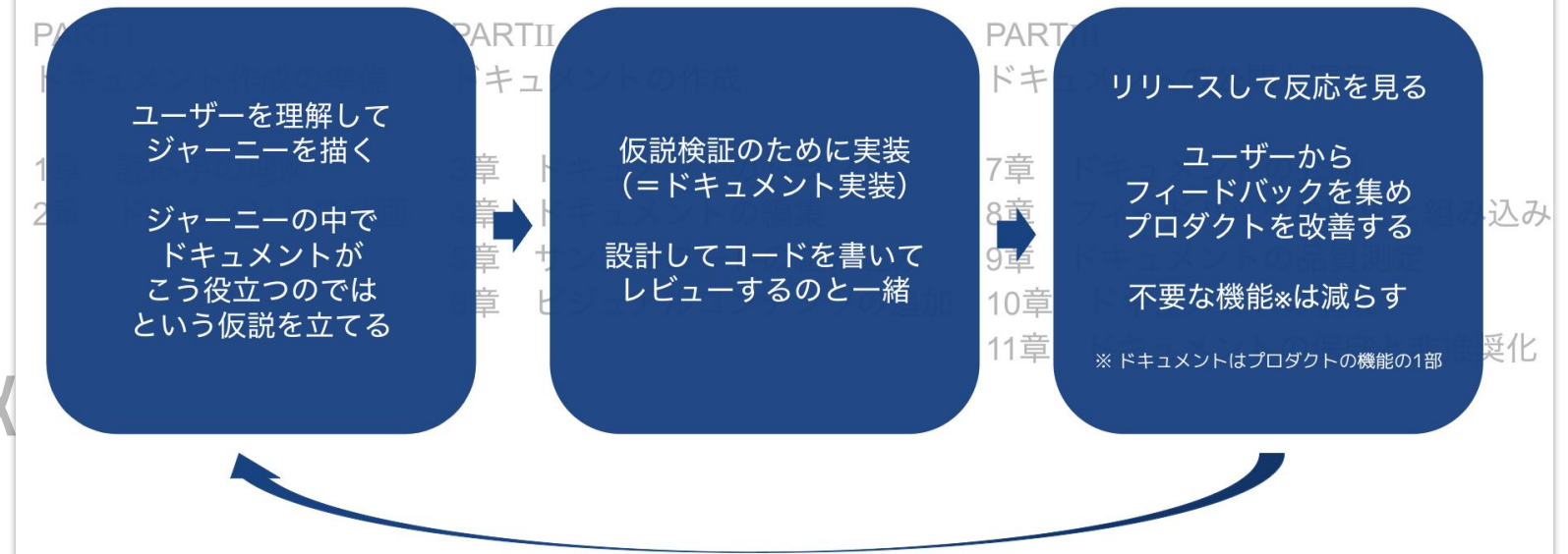
今日お話ししたこと

- 書籍の概要
- 特に意識しておきたい箇所
 - 読み手の理解（1章）
 - ドラフトの執筆（3章）
 - フィードバックの収集（8章）
 - ドキュメントの品質測定（9章）
- 時間があれば日本語特有の観点（+α）

書籍の構成

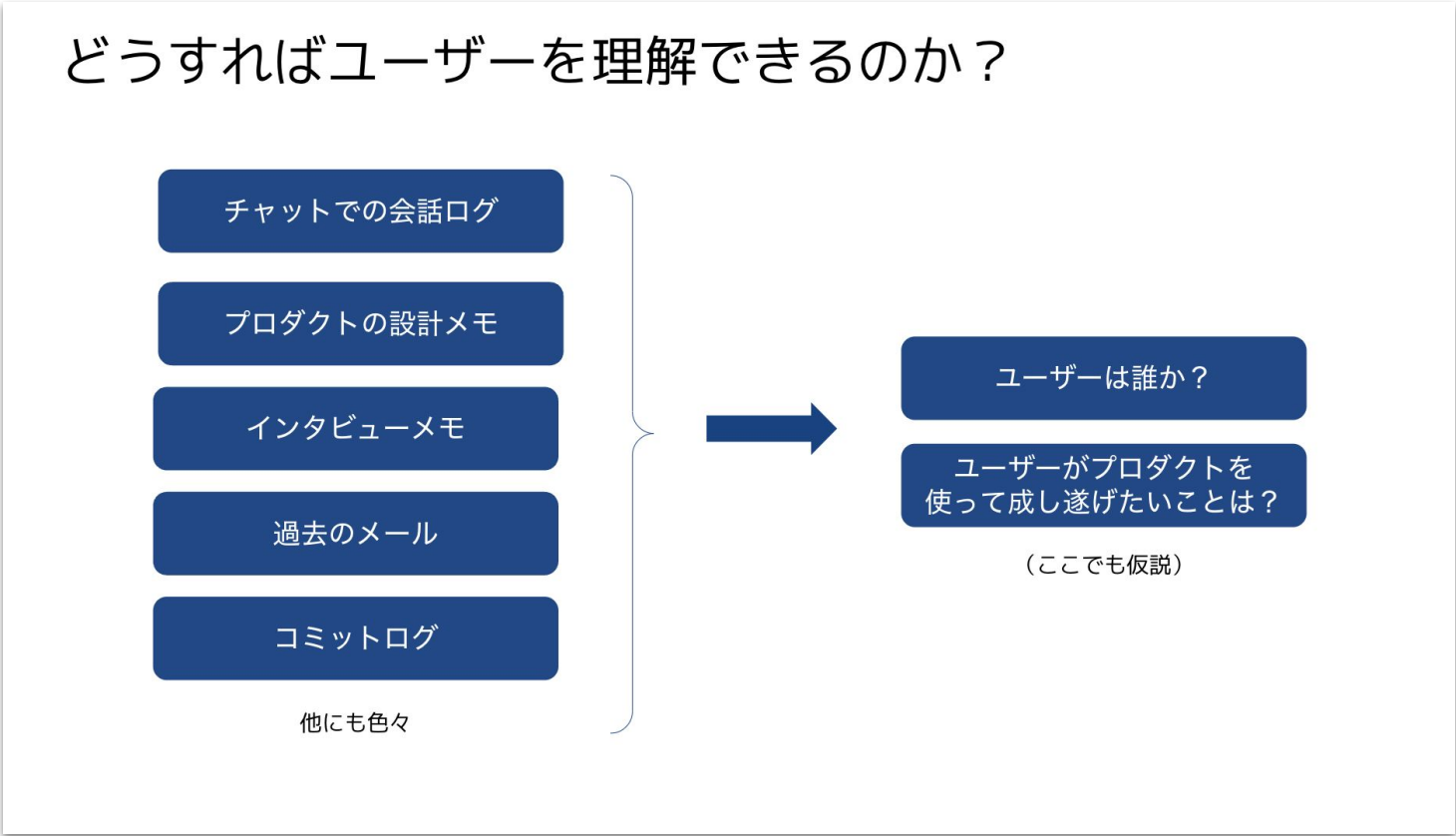
PART I ドキュメント作成の準備	PART II ドキュメントの作成	PART III ドキュメントの公開と運用
1章 読み手の理解	3章 ドキュメントのドラフト	7章 ドキュメントの公開
2章 ドキュメントの計画	4章 ドキュメントの編集	8章 フィードバックの収集と組み込み
	5章 サンプルコードの組み込み	9章 ドキュメントの品質測定
	6章 ビジュアルコンテンツの追加	10章 ドキュメントの構成
		11章 ドキュメントの保守と非推奨化

プロダクト開発と同じ流れ



今日お話ししたこと

- 書籍の概要
- 特に意識しておきたい箇所
 - 読み手の理解（1章）
 - ドラフトの執筆（3章）
 - フィードバックの収集（8章）
 - ドキュメントの品質測定（9章）
- 時間があれば日本語特有の観点（+α）



ユーザージャーニーマップの例

	認識・認知	興味・関心	評価・検討	決定・購入	実行・利用	継続利用・推奨
疑問	chatgptはどんな機能があるの？	chatgptは私の業務に役立つの？	chatgptの価格は手頃か？	chatgptは使いやすいか？	chatgptの使い方は？	chatgptは期待通りの性能か？
インタラクション (ユーザー体験)	ウェブサイトや業界ニュースを閲覧	ブログ記事や専門家の意見を読む	chatgptの公式サイトで価格情報を調べる	chatgptの無料トライアルを利用	チュートリアルやヘルプセンターを参照	chatgptを定期的に使い、効果を評価
感情	好奇心が刺激され、興味を持つ	興味が高まるが、実際に役立つか不安	比較的リーズナブルと感じるが、自分にとって価値があるか懸念	使い始めは戸惑いがあるが、興味を持ち続ける	最初は不慣れでストレスを感じるが、徐々に慣れてくる	最初は不安だが、使い続けるうちに満足度が向上する
改善候補	わかりやすいプロダクト紹介ページの改善	より具体的なユースケースの紹介	さまざまなプランの提示	トライアル期間の延長	より詳細な使い方がガイドの提供	フィードバック機能の充実

感情の起伏ラインを書くことも

今日お話ししたこと

- 書籍の概要
- 特に意識しておきたい箇所
 - 読み手の理解（1章）
 - ドラフトの執筆（3章）
 - フィードバックの収集（8章）
 - ドキュメントの品質測定（9章）
- 時間があれば日本語特有の観点（+0）

ドラフトを書く上で最初に考えるべきこと

- 冒頭で以下を整理する
 - 「誰が」「何のために」読みにくるのか？
 - 「どのタイプ」のドキュメントが適切か？
- 次にタイトルとアウトラインを決める
 - タイトル = ドキュメントを読んで達成できるゴールの要約
 - 例：カスタム会話型チャットボットの開発 など
 - アウトライン = ゴールに必要な大まかな要素や手順
 - 例：開発環境、API認証の方法 など

ドキュメントにまつわる真実

- ユーザーは情報を探してドキュメントにたどり着く
 - ↓
 - ユーザーは書いてある内容をほとんど読まない
- だから流し読みしやすくする

今日お話ししたこと

- 書籍の概要
- 特に意識しておきたい箇所
 - 読み手の理解（1章）
 - ドラフトの執筆（3章）
 - フィードバックの収集（8章）
 - ドキュメントの品質測定（9章）
- 時間があれば日本語特有の観点（+0）

ユーザーFBの集め方

- フィードバックを受け取るための経路を用意する
- 例
 - ドキュメントのページへformなどを埋め込む
 - サポートチケットから読み取る
 - ドキュメントに対する感情を集める

このページはお役に立ちましたか? ^x

詳しくお聞かせください。

お客様の経験を共有してください。ただし、機密情報や個人情報は含めないでください。

[プライバシー ポリシー](#)

<https://learn.microsoft.com/ja-jp/azure/virtual-machines/linux/quick-create-di>

Priority label	What it means	Examples
priority/critical-urgent	Te th St	超重要 user-visible bugs in core features broken builds tests and critical security issues
priority/important-soon	M in	なるべく早くやる。できれば次のリリースで。 [XXXX]
priority/important-longterm	Im an re	長期的には重要 [XXXX]
priority/backlog	General agreement that this is a nice-to-have, but no one's available to to we reviewed if reviewers are fully occupied with higher priority issues— for example, immediately before a release.	あったらいいけどすぐやらない [XXXX]
priority/awaiting-more-evidence	Possibly useful, but not yet enough support to actually get it done.	たぶん有用だけどまだ十分な支持がない Mostly placeholders for potentially good ideas, so that they don't get completely forgotten, and can be referenced or deduped every time they come up

<https://www.kubernetes.dev/docs/guide/issue-triage/>

今日お話ししたこと

- 書籍の概要
- 特に意識しておきたい箇所
 - 読み手の理解（1章）
 - ドラフトの執筆（3章）
 - フィードバックの収集（8章）
 - **ドキュメントの品質測定（9章）**
- 時間があれば日本語特有の観点（+0）

優れたドキュメントって何だろう？

- ドキュメントの品質の定義

「ドキュメントが優れているのは**目的**にかなっている場合である。」



ユーザーの特定の行動を促進すること + 組織のゴールを達成すること

ドキュメントの品質の構成要素

- 機能品質
 - ドキュメントの目的やゴールが達成されているか？
- 構造品質
 - ドキュメント自体がうまく書かれているか？
 - うまく構成されているか？

さて、どちらの品質が重要でしょう？🤔

→ 機能品質 ∵ どれだけ最高の表現でもゴール達成しないと無価値

今日お話ししたこと

- 書籍の概要
- 特に意識しておきたい箇所
 - 読み手の理解（1章）
 - ドラフトの執筆（3章）
 - フィードバックの収集（8章）
 - ドキュメントの品質測定（9章）
- 時間があれば日本語特有の観点（+α） → 良書で補完！

ゴールのおさらい

今日のゴール (イベント終了時)

- ドキュメントライティングのスキルが高まっている
 - ドキュメントライティングの概要を知っている
 - ドキュメントの「準備」「作成」「運用」で重要なポイントを理解している
- (書籍でカバーされていないが抑えておくと良い)
日本語文章で大事な点を知っている

とはいえ、まだまだ話していない内容がたくさんある！

- 良いサンプルコードってなんだろう？
- 良いビジュアルコンテンツ、図表ってなんだろう？
- 作ったとして、どうやってレビューすればいいんだろう？
- 良いフィードバックって何だろう？
- ドキュメントが増えてきたらどうやって全体構成すればいいんだろう？
- ドキュメントの廃止 (deprecate) ってどうやればいいんだろう？
- etc…

まだまだ話していない内容がたくさん

- 良いサンプルコードってなんだろう？
- 良いビジュアルコンテンツ、図表ってなん

書籍に全部書いてあります！

- よろしくお願いします！
- ドキュメントが増えてきたらどうやって全
- ドキュメントの廃止 (deprecate) ってど
- etc...

エンジニアのための ドキュメント ライティング

ジャレッド・バーティ
ザッカリー・サラ・コーライセン
ジェン・ランボン
デービッド・ヌーニェス
ハイディ・ウォーターハウス 著

岩瀬義昌 訳

Docs for Developers

An Engineer's Field Guide
to Technical Writing



開発者の生産性 / プロダクトの成長 / ユーザーの採用率 / を改善する

ドキュメントの作成プロセスを伝授!

ドキュメントがない

書き方がわからない

内容がわかりづらい

よくある
問題を
解決!

すぐれたドキュメントが 開発を最適化する!

ユーザーの問題解決と
プロダクトの成功を導く