

Webサイト制作で デザイナーがエンジニアと 円滑にコミュニケーションを 取る方法は？

2025.02.13 @ Adobe MAX Japan 2025

セッション中の

動画の撮影・配信はNGですが

写真撮影 (+SNS等への投稿) は可能です。



unshift Inc. / Nauts (<https://nauts.co.jp/>)

Developer

長谷川 巧

1人法人 / Webフロントエンド / WebGL雰囲気やってる /
3歳の娘 (かわいい) / クリエイティブコーディング /
バスケットボール / SIGMA fp

Website <https://unshift.co.jp/>

X (Twitter) [@_unshift](https://twitter.com/_unshift)

Instagram [@_unshift](https://www.instagram.com/_unshift)

クリエイティブコーディングギャラリー
<https://unshift.notion.site/creativecodinggallery>

その他

- [STUDIO DETAILS エンジニアunit 外部アドバイザー](#)
- 2025年に武蔵野美術大学で特別講師

X (@_unshift) に[Notionのプロフィールページ](https://unshift.notion.site/creativecodinggallery)のリンクを投稿しました。

Intro

**このセッションは、基本的に
デザイン専門 (実装はしない)のデザイナー
向けのセッションではあるんですが...**

Intro

**いろいろな属性の方々にも
共感・新たな発見などを
提供できたらいいなと思っています。**

Intro

ではどんな属性ががあるかというと…

A. デザイン専門のデザイナー

B. デザイン→実装を一人で行うデザイナー

C. エンジニア

D. デザイナーとエンジニアを見守る人

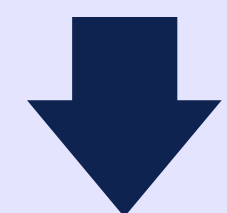
(ディレクターやプロジェクトマネージャーなど)

A. デザイン専門のデザイナー

今回のメインターゲット

デザイン専門のデザイナーからは見えづらい

エンジニア目線の情報を得られる！



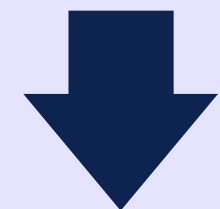
その結果…

実装のポイントをおさえつつ設計できるので

プロジェクト全体のコミュニケーションのコストが減らせる！

B. デザイン→実装を一人で行うデザイナー

普段自分がやっている (やっていないかもしれない) ことを
改めて言語化・整理した情報が得られる！



その結果…

自分の制作時に見落としていた点の発見や
制作フローの見直しにつながる (かも)！

分業で行うことになった際の注意点なども得られる！

C. エンジニア

他のエンジニアが気にしているポイントを得られる！



**自分とはタイプの違うエンジニアの情報を得ることにより
新たな気づきを得られる (かも)！**

(※共感したらめっちゃくちゃ頷いてください！登壇者が喜びます。)

D. デザイナーとエンジニアを見守る人

(ディレクターやプロジェクトマネージャーなど)

デザイナーとエンジニアがやり取りする際に必要な情報を得られる！



情報設計の際に役立つ！

**エンジニアとデザイナーが見落としているポイントを
助言できるようになる！**

クライアントに説明する際の武器が増える (かも)！

Intro

みんなハッピーになってほしい

Index

1

まずは、プロジェクトを成功させるという心構えが必要

2

エンジニアがWebサイトを実装するときに気にする点

3

エンジニアとの共通言語を増やすための基礎知識

4

やり取りの際に意識したほうがよいこと

Index

1 | **まずは、プロジェクトを成功させるという心構えが必要**

2 | **エンジニアがWebサイトを実装するときに気にする点**

3 | **エンジニアとの共通言語を増やすための基礎知識**

4 | **やり取りの際に意識したほうがよいこと**

1. まずは、プロジェクトを成功させるという心構えが必要

**Webサイト制作のプロジェクトにおいて
デザイナーにとってのゴールは
デザイン通りに実装してもらおうこと
ではなく…**

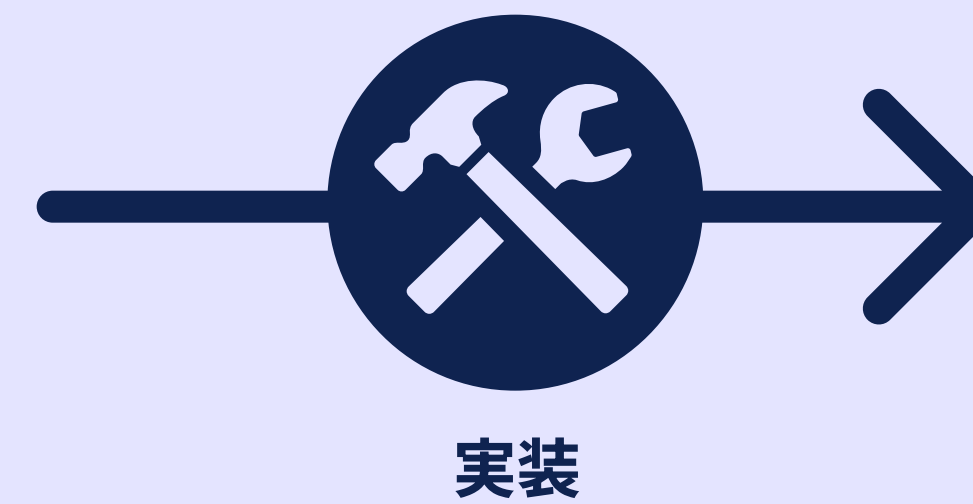
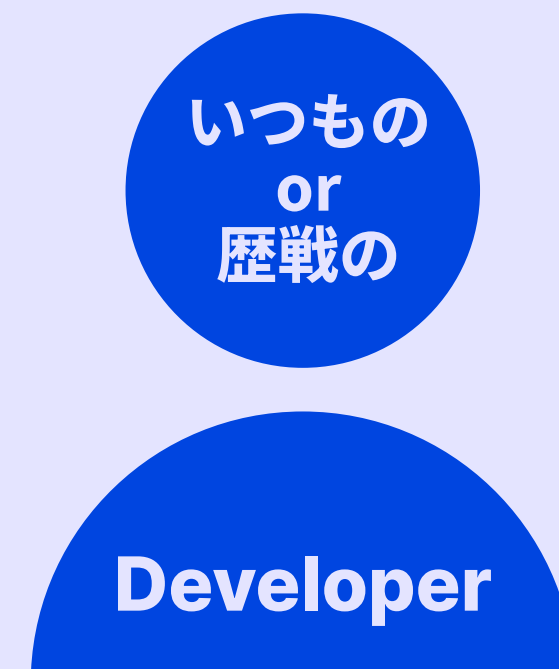
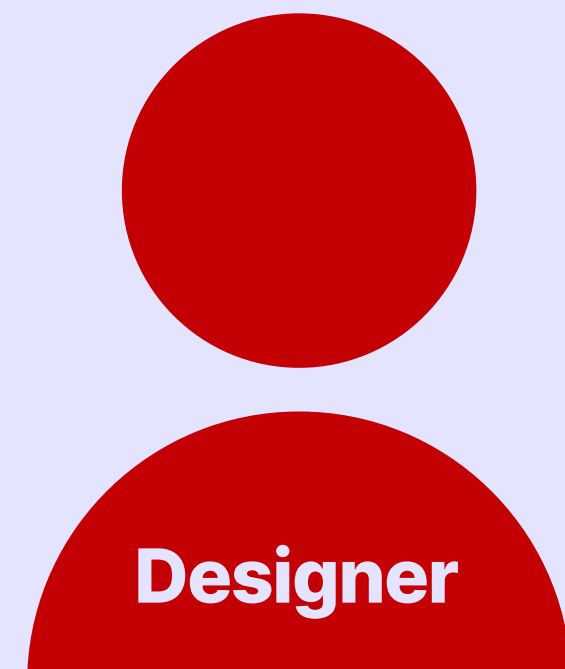
1. まずは、プロジェクトを成功させるという心構えが必要

**スケジュール通りに予算内で
制作チーム※が納得したものを制作し
かつ使用するユーザーも
満足できるWebサイトをつくること。**

※クライアントワークの場合はクライアントも含む

1. まずは、プロジェクトを成功させるという心構えが必要

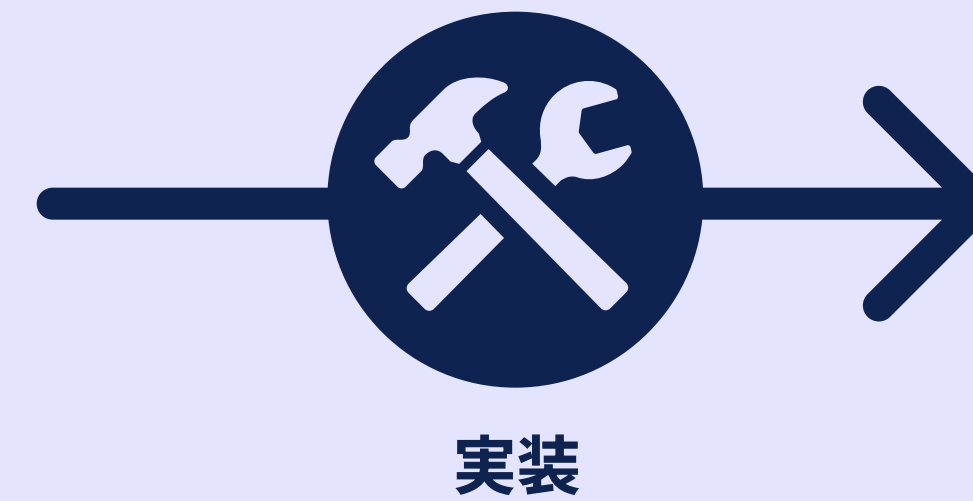
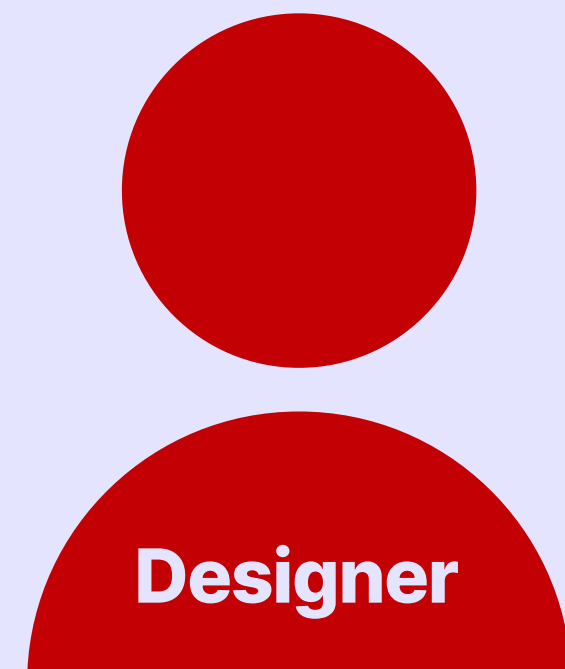
普段いつも一緒に仕事をしているエンジニアや
経験豊富なエンジニアであれば
余計なやり取りもあまりなく
スムーズに制作できても…



正直いつもこんなチームでできたら最高!

1. まずは、プロジェクトを成功させるという心構えが必要

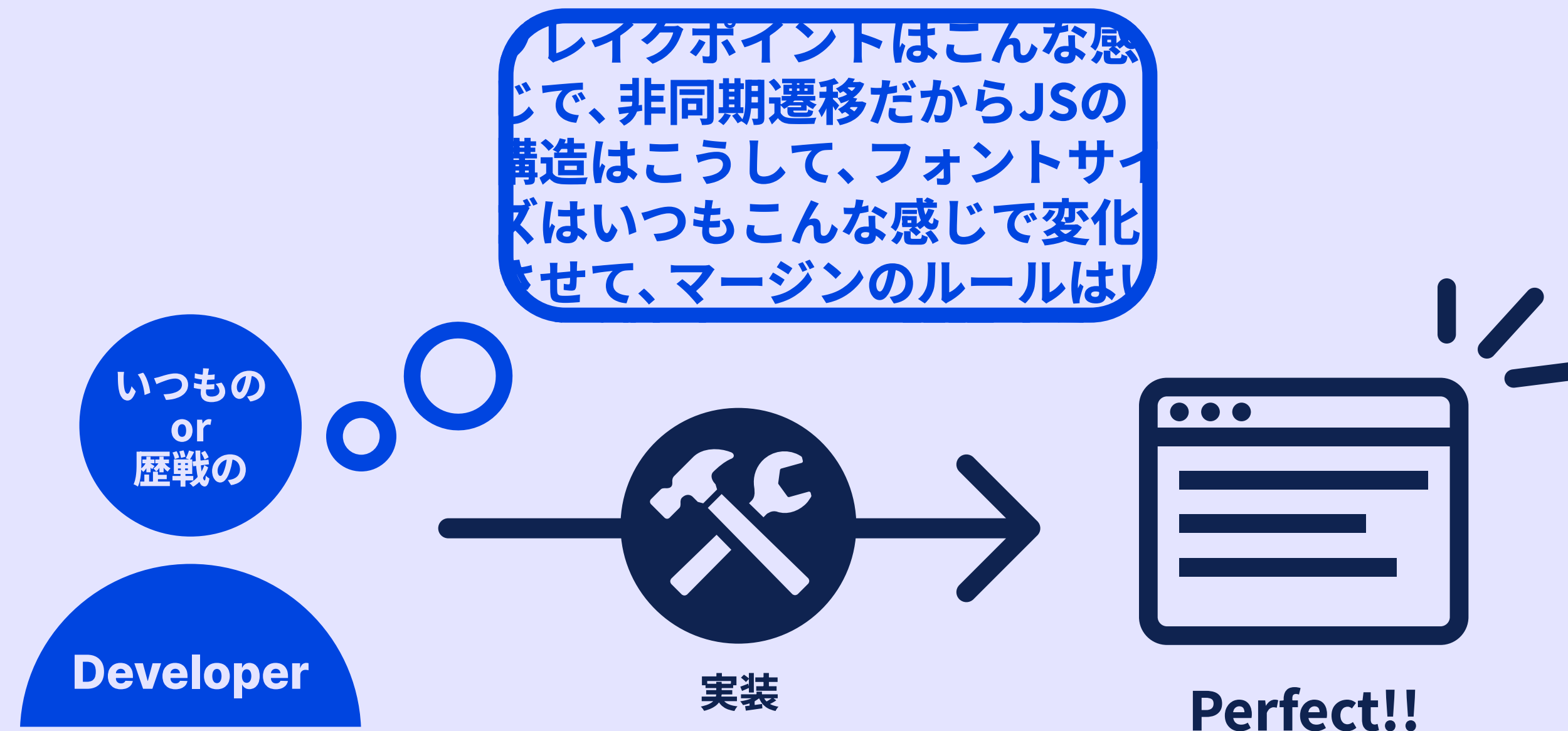
初めて一緒に制作するエンジニアや
経験の浅いエンジニアに実装してもらおうと
いつもどおりのやり方で
うまくいかない！！



そんなときはもしかしたら…

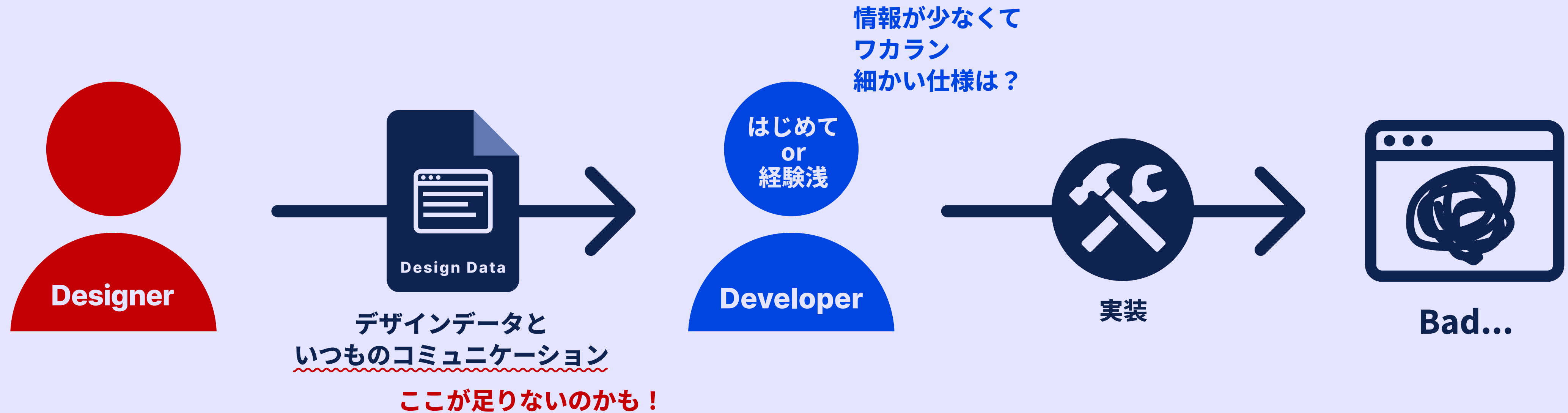
1. まずは、プロジェクトを成功させるという心構えが必要

普段いつも一緒に仕事をしているエンジニアや
経験豊富なエンジニアが
宙ぶらりんになっている仕様を
いい感じに決めてくれているのかも…



1. まずは、プロジェクトを成功させるという心構えが必要

デザイナーからエンジニアに渡す 情報が足りないこと (コミュニケーション不足)を まずは疑ってみる。

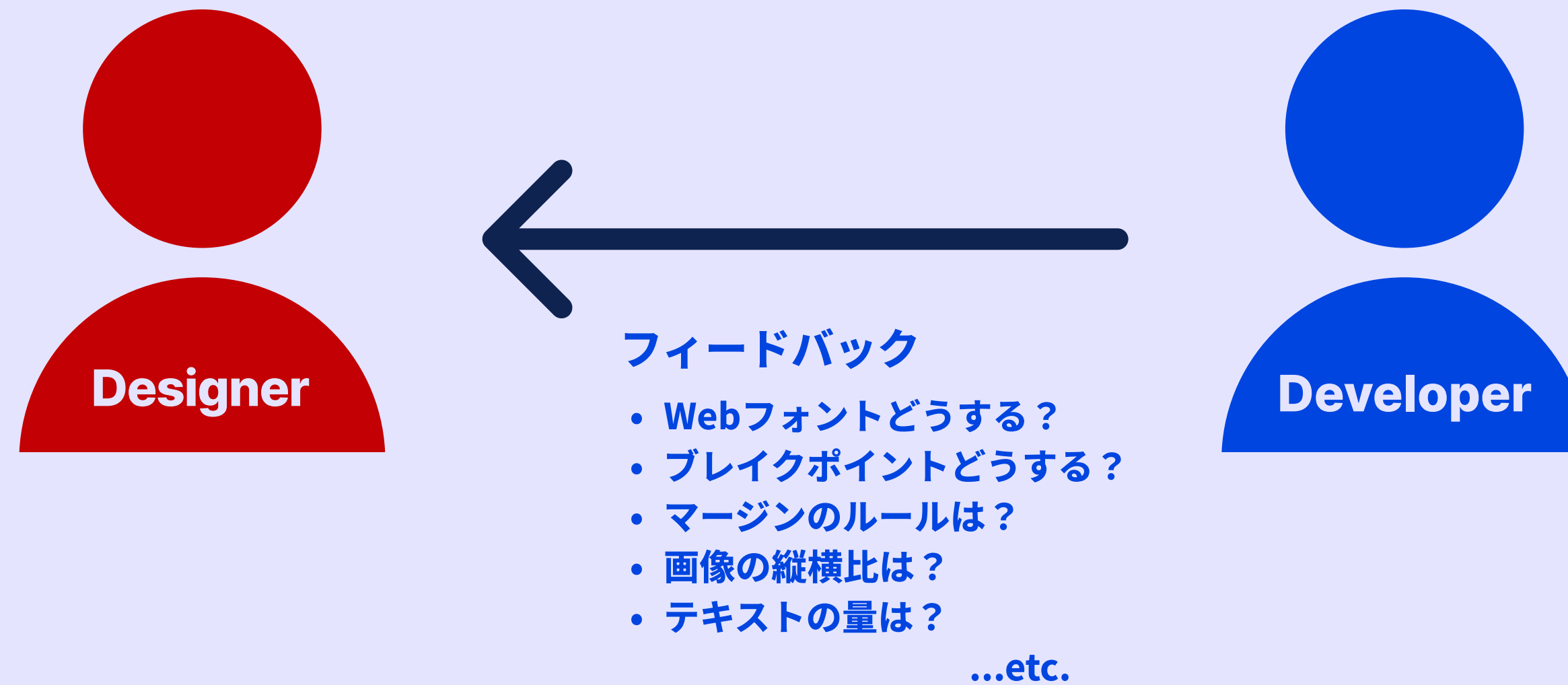


1. まずは、プロジェクトを成功させるという心構えが必要

逆もまた然り。

実装するための情報が足りなければ

エンジニアからの積極的なフィードバックも必要。



1. まずは、プロジェクトを成功させるという心構えが必要

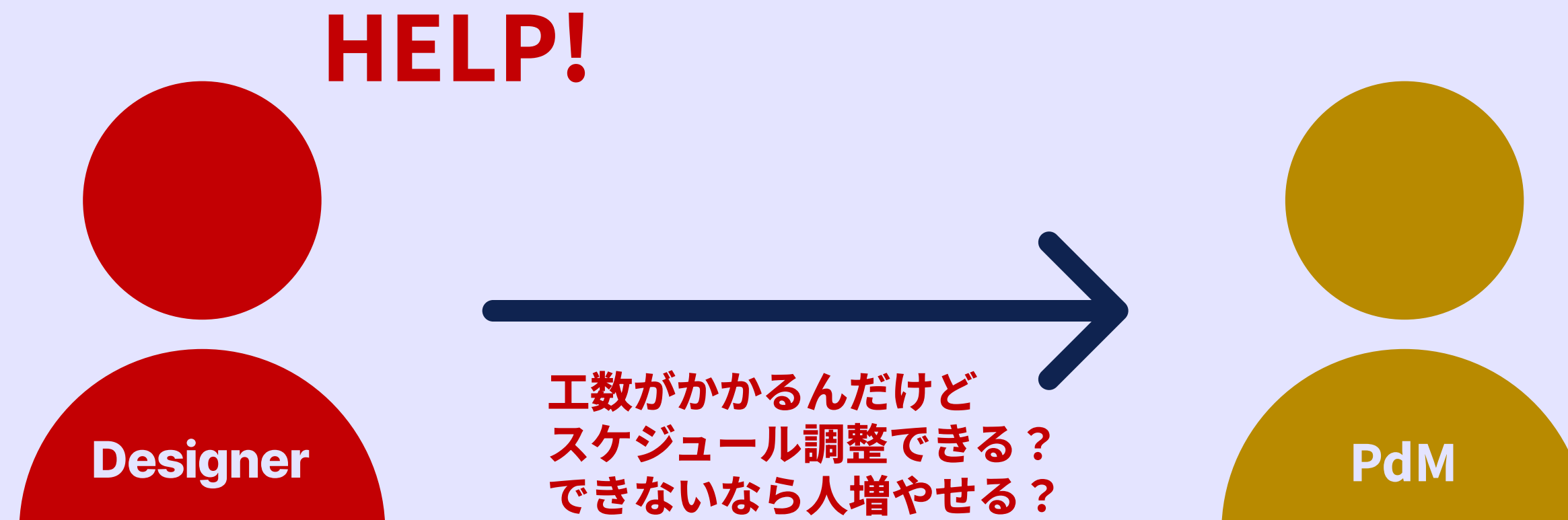
**ないものを嘆くのではなく
双方が歩み寄ることで
プロジェクト進行をスムーズにしたほうが
生産的。**

ただ、そうは言っても…

1. まずは、プロジェクトを成功させるという心構えが必要

コミュニケーションのコストがかかるのであれば
その分の工数をプロジェクト責任者に報告する必要がある。

場合によってはアサインのミスの可能性もあるため
交代もしくは増員も検討しなければならない。



1. まずは、プロジェクトを成功させるという心構えが必要

まとめ

**足りないものをチームで補い合い
現実的な落とし所を探りつつ
プロジェクトを進行することを
常に頭に入れておく。**

Index

1

まずは、プロジェクトを成功させるという心構えが必要



2

エンジニアがWebサイトを実装するときに気にする点

3

エンジニアとの共通言語を増やすための基礎知識

4

やり取りの際に意識したほうがよいこと

2. エンジニアがWebサイトを実装するときに気にする点

**じゃあ実際
エンジニアがWebサイト実装時に
気にする点ってなに？**

2. エンジニアがWebサイトを実装するときに気にする点

- a. ルールと共通パーツ・UI
- b. フォント・タイポグラフィ
- c. ウィンドウのサイズによるレイアウトの変化
- d. コンテンツの量・種類によるレイアウトの変化
- e. インタラクション
- f. リッチなグラフィック表現
- g. UI・コンテンツの状態
- h. その他

2. エンジニアがWebサイトを実装するときに気にする点

a. ルールと共通パーツ・UI

2. エンジニアがWebaサイトを実装するときに気にする点

2-a. ルールと共通パーツ・UI

余白やフォントサイズに
規則性があるか

=ルール化できるか

早く決めたいレベル ★★★★★・

タイトル・リード・本文の
組み合わせのときは
このルールっぽい。

font-size: 32px
下層ページタイトル
margin: 10px
リード文のフォントサイズはタイトルよりも
少し小さくなるよ。 font-size: 20px
margin: 20px
本文のテキストサイズは全部こうなるよ。本文のテキストサイ
ズは全部こうなるよ。本文のテキストサイズは全部こうなる
よ。本文のテキストサイズは全部こうなるよ。本文のテキスト
サイズは全部こうなるよ。本文のテキストサイズは全部こうな
るよ。本文のテキストサイズは全部こうなるよ。
font-size: 16px

h1 タイトル

h2 タイトル

h3 タイトル

テキスト周りは全部
このルールになってそう。

本文のテキストサイズは全部こうなるよ。本文のテキストサイズは全部こうな
るよ。本文のテキストサイズは全部こうなるよ。本文のテキストサイズは全部
こうなるよ。本文のテキストサイズは全部こうなるよ。本文のテキストサイ
ズは全部こうなるよ。本文のテキストサイズは全部こうなるよ。

- 番号なしリストのスタイルはこうなるよ
 - 番号なしリストのスタイルはこうなるよ
 - 番号なしリストのスタイルはこうなるよ
 - 番号なしリストのスタイルはこうなるよ
1. 番号なしリストのスタイルはこうなるよ
 2. 番号なしリストのスタイルはこうなるよ
 3. 番号なしリストのスタイルはこうなるよ
 4. 番号なしリストのスタイルはこうなるよ

2. エンジニアがWebaサイトを実装するときに気にする点

2-a. ルールと共通パーツ・UI

使い回せるUIやパーツがあるか
=コンポーネント化できるか

早く決めたいレベル ★★★★★・

About us →

詳細を見る →

サイズと色、フォントが違うけど
色んなところで使っているので
共通コンポーネント化できそう。

2. エンジニアがWebサイトを実装するときに気にする点

b. フォント・タイポグラフィ

2. エンジニアがWebサイトを実装するときに気にする点

2-b. フォント・タイポグラフィ

Webフォントを使用するか
するならどう使用するか
(サービス or セルフホスティング)

早く決めたいレベル ★★★★★・

クライアントの本番サーバーと
弊社のプレビューサーバー
どちらでもWebフォントが
表示されるようにしておかないと。



Adobe Fonts
Google Fonts
etc...

or



→
upload

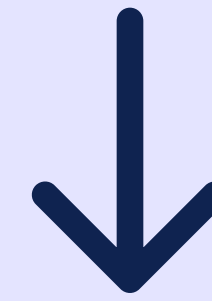


Web Server

2. エンジニアがWebサイトを実装するときに気にする点

2-b. フォント・タイポグラフィ

あのイーハトーヴォのすきとおった風、夏でも底に冷たさをもつ青いそら、うつくしい森で飾られたモリーオ市、郊外のぎらぎらひかる草の波。



約物をつめるかどうか

早く決めたいレベル ★★・・・

CSSの設定だけでいけるかな。
約物半角フォントが必要かな。

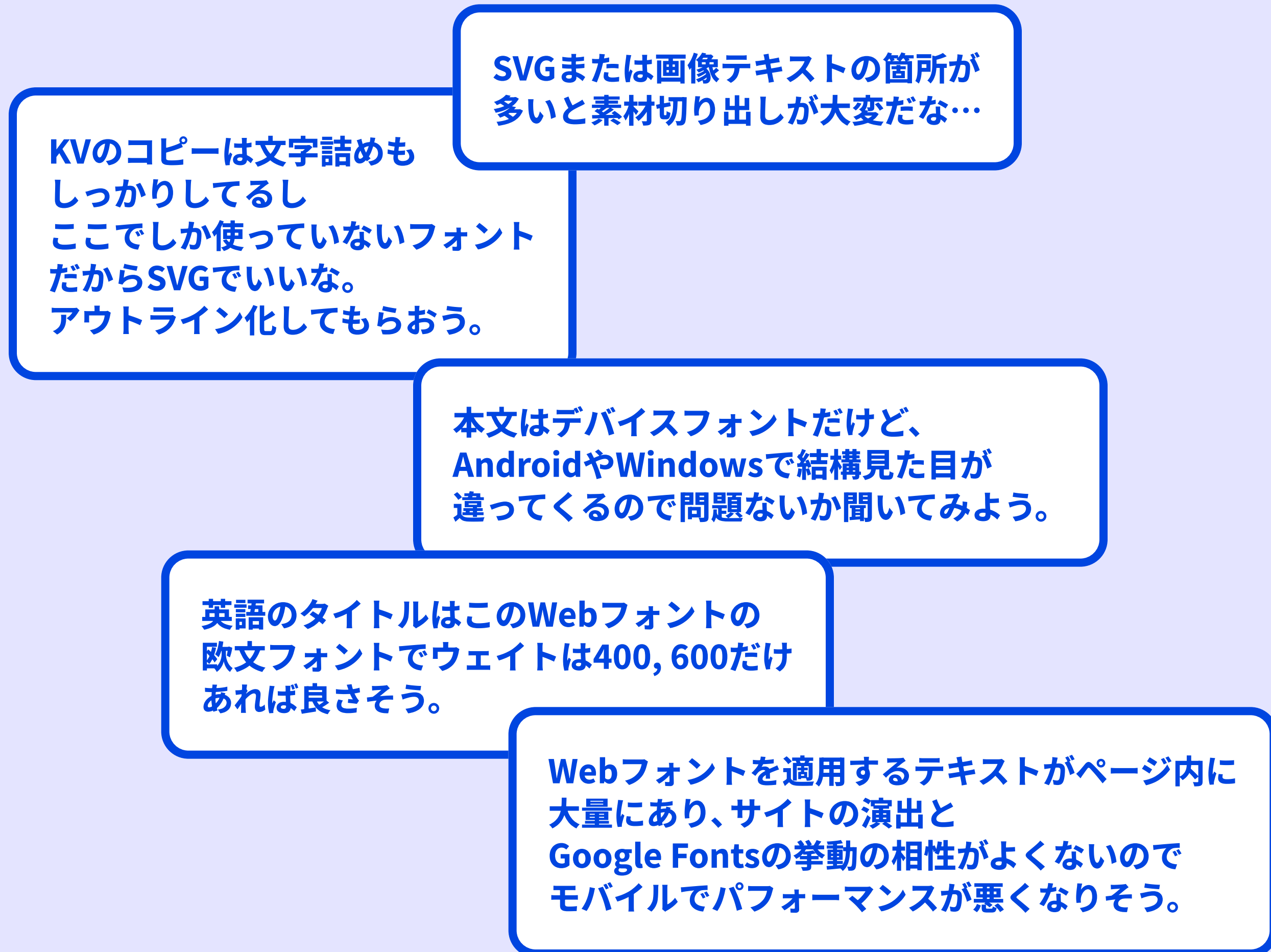
あのイーハトーヴォのすきとおった風、夏でも底に冷たさをもつ青いそら、うつくしい森で飾られたモリーオ市、郊外のぎらぎらひかる草の波。

2. エンジニアがWebサイトを実装するときに気にする点

2-b. フォント・タイポグラフィ

(個別のテキストごと)
Webフォントなのか
デバイスフォントなのか
画像テキストなのか

早く決めたいレベル ★★★★★・



2. エンジニアがWebサイトを実装するときに気にする点

c. ウィンドウのサイズによる レイアウトの変化

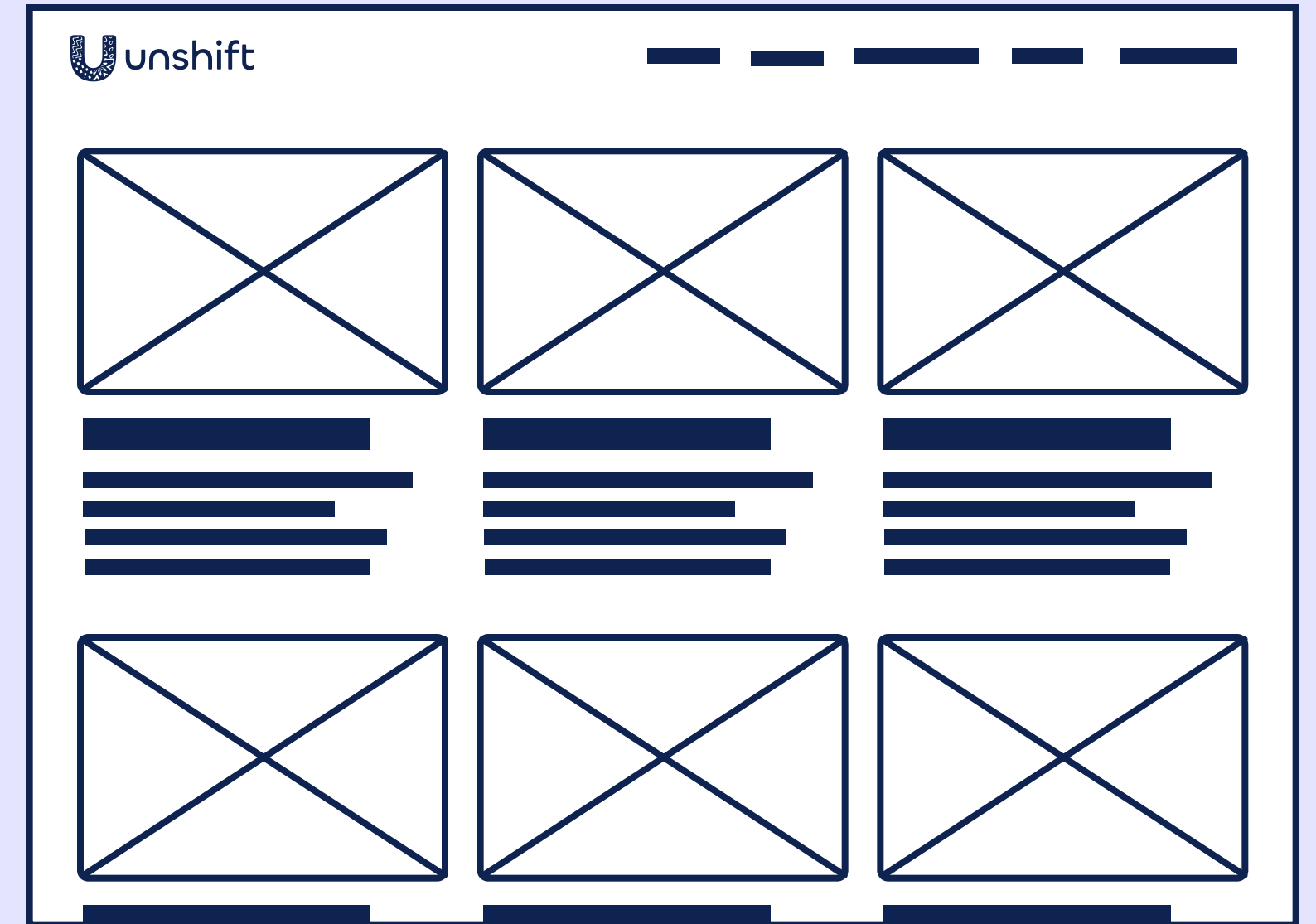
2. エンジニアがWebサイトを実装するときに気にする点

2-c. ウィンドウのサイズによるレイアウトの変化

S size
~768px

M size
768px~1024px

L size
1024px~



ブレイクポイント はどうするか

早く決めたいレベル ★★★★★・

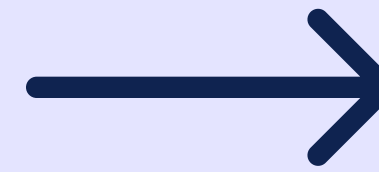
プロジェクトメンバーと相談して
今回のサイト用に最適化して決定するか
それが特になければいつものやつにしよう。

2. エンジニアがWebサイトを実装するときに気にする点

2-c. ウィンドウのサイズによるレイアウトの変化

フォントサイズの変化の仕方 はどうなるか (固定 or 変動)

早く決めたいレベル ★★★★★



ウィンドウサイズ
が広がっても
フォントサイズは固定



or



ウィンドウサイズに
合わせて
フォントサイズが変動



フォントサイズを変動させるなら
KVエリアの高さもフォントサイズに合わせて
変動させないと。

※参照: Responsive Samples (<https://responsive-samples.unshift.co.jp/>)

2. エンジニアがWebサイトを実装するときに気にする点

**d. コンテンツの量・種類による
レイアウトの変化**

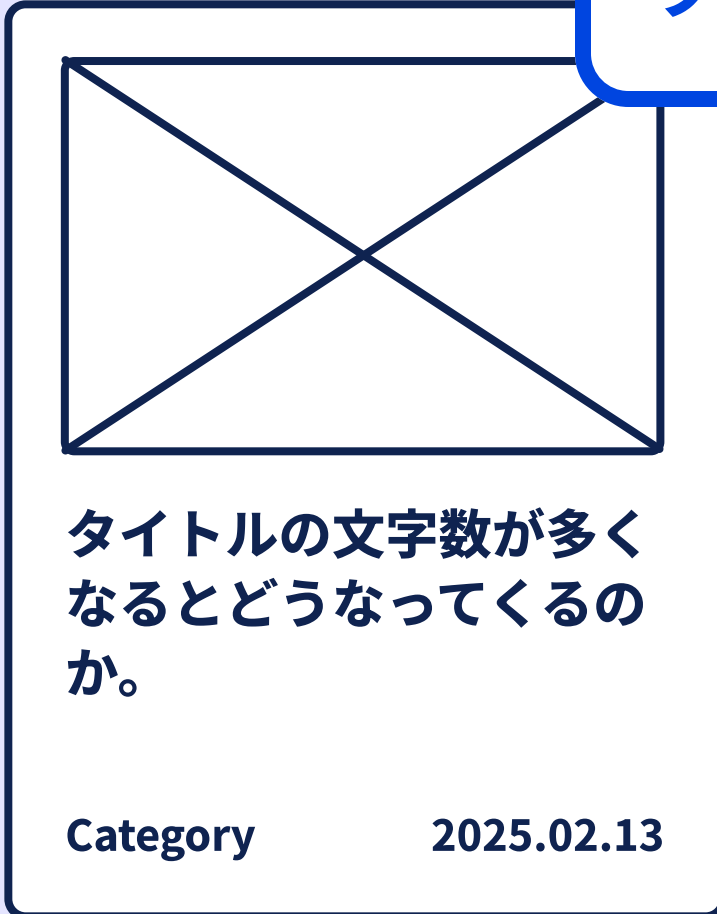
2. エンジニアがWebサイトを実装するときに気にする点

2-d. コンテンツの量・種類によるレイアウトの変化

テキスト量が多くなると
どうなるのか
(折り返す or 切り取るなど)

早く決めたいレベル ★★・・・

成り行きでテキストを表示したとき
各項目の高さも合わせるなら
フッター要素は下付きになる？



or

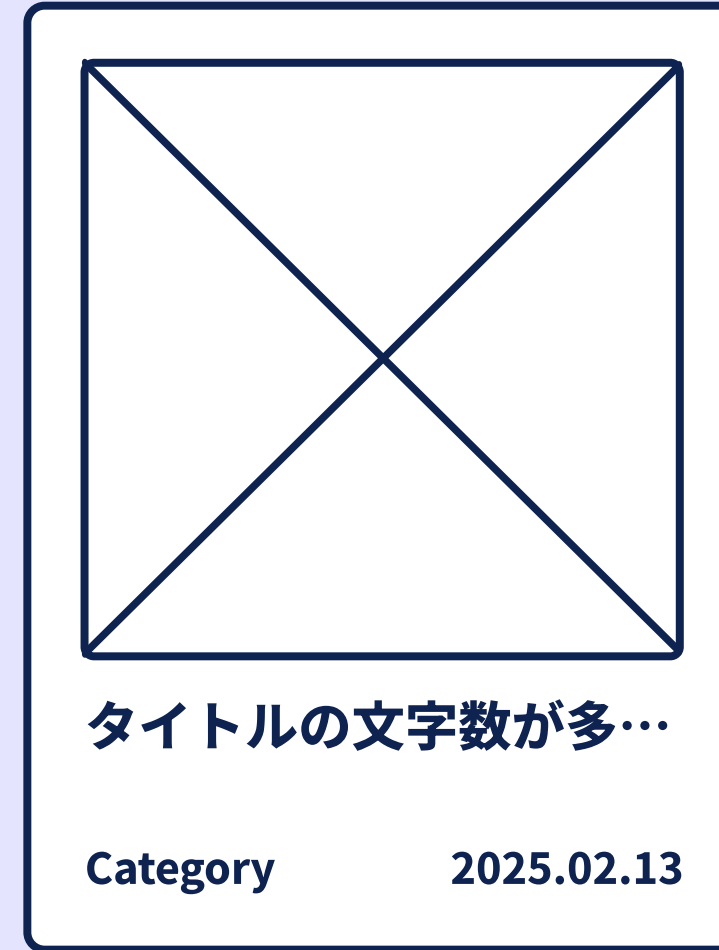


2. エンジニアがWebサイトを実装するときに気にする点

2-d. コンテンツの量・種類によるレイアウトの変化

画像の縦横比は どうなるのか

早く決めたいレベル ★★ . . .



同じコンポーネントで
画像の縦横比を変えるなら
高さはばらばらになる？
揃える？



表示領域は揃えて
余白を設ける？
余白は背景色あり？なし？

2. エンジニアがWebサイトを実装するときに気にする点

e. インタラクシヨソ

2. エンジニアがWebサイトを実装するときに気にする点

2-e. インタラクション

マウスオーバーの表現 はどうなるのか

早く決めたいレベル ★★★★★・

default

詳細を見る



hover

詳細を見る



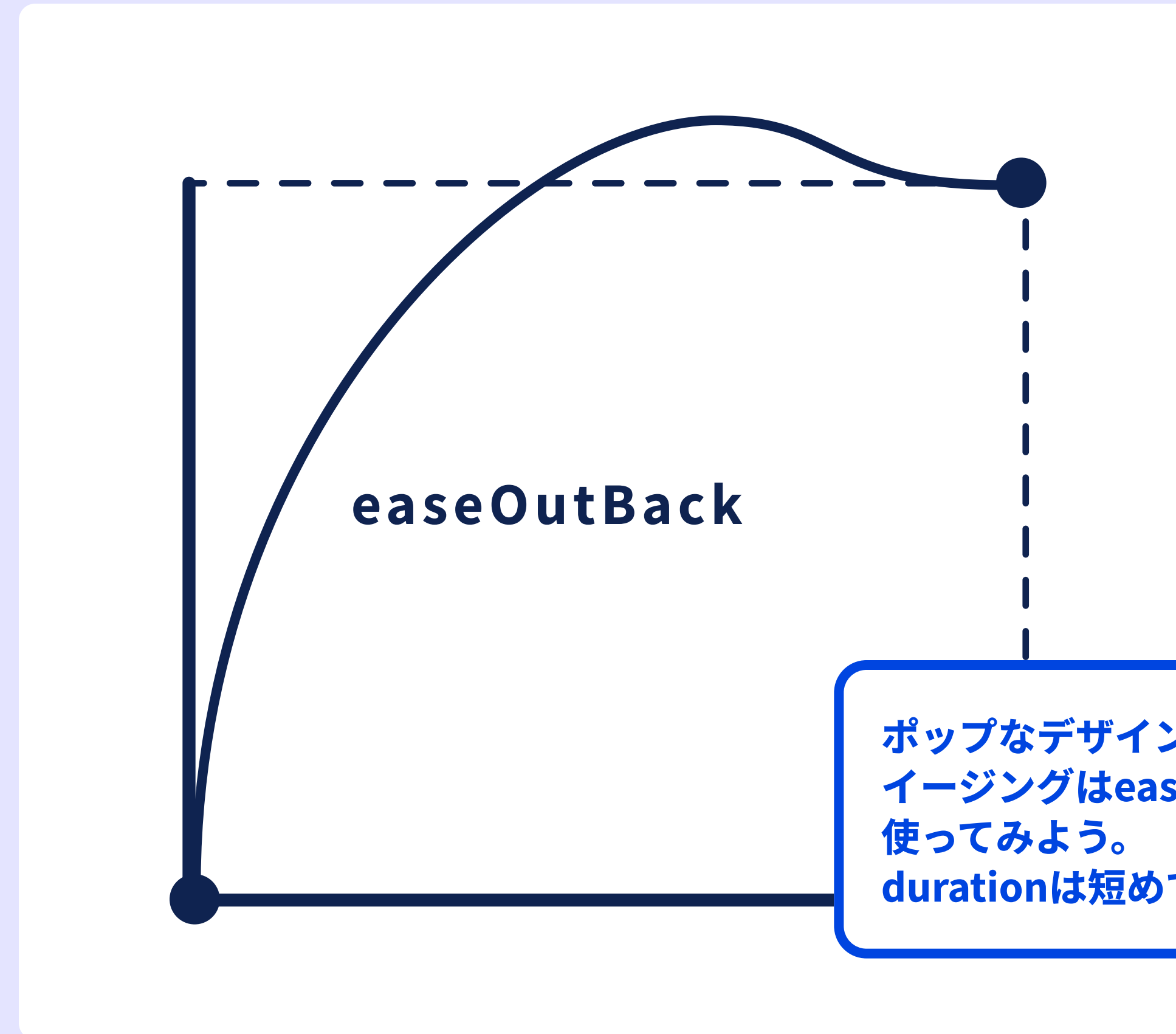
演出によっては
特殊な構造にしないと
いけなかったりするので
早めに決まっていると嬉しい。

2. エンジニアがWebサイトを実装するときに気にする点

2-e. インタラクション

アニメーションの雰囲気 (主にeasing)はどうか

早く決めたいレベル ★★★★★・



2. エンジニアがWebサイトを実装するときに気にする点

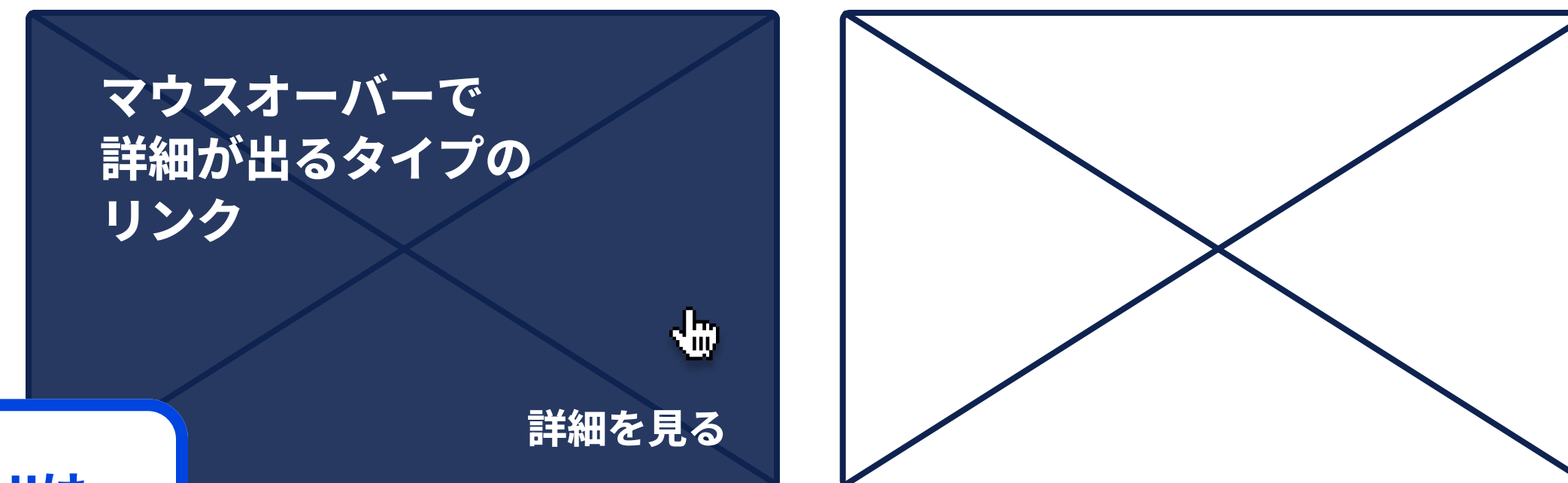
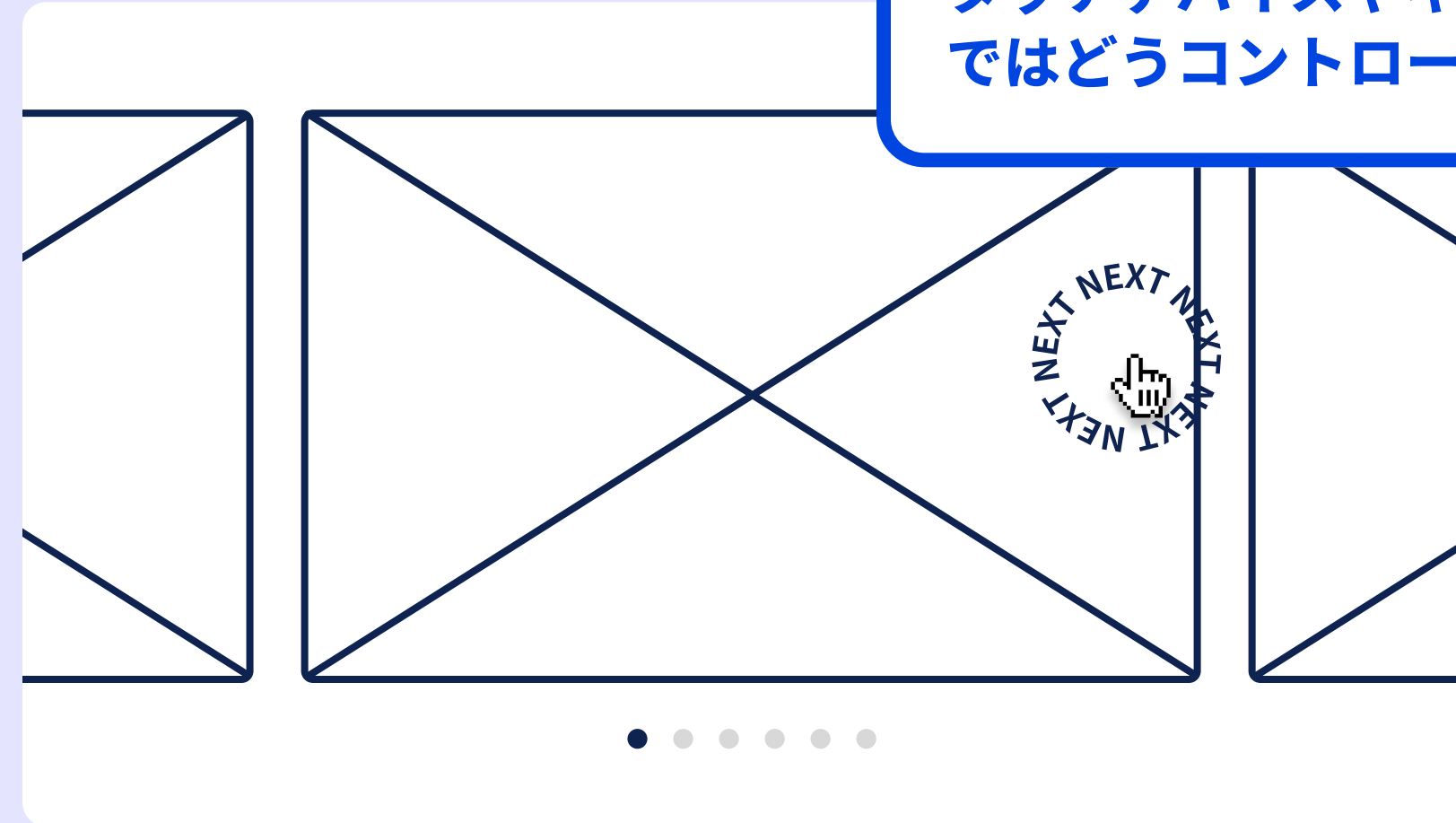
2-e. インタラクション

操作の違い

(マウス/タッチ/キーボード)に どう対応するか

早く決めたいレベル ★★・・・

カルーセルUIの左右移動の
ナビゲーションがマウススローカータイプで
マウス位置によって左右が切り替わる場合
タッチデバイスやキーボード
ではどうコントロールする？



マウスオーバーで情報を出すUIは
タッチやキーボードではどうやって
実現する？

2. エンジニアがWebサイトを実装するときに気にする点

f. リッチなグラフィック表現

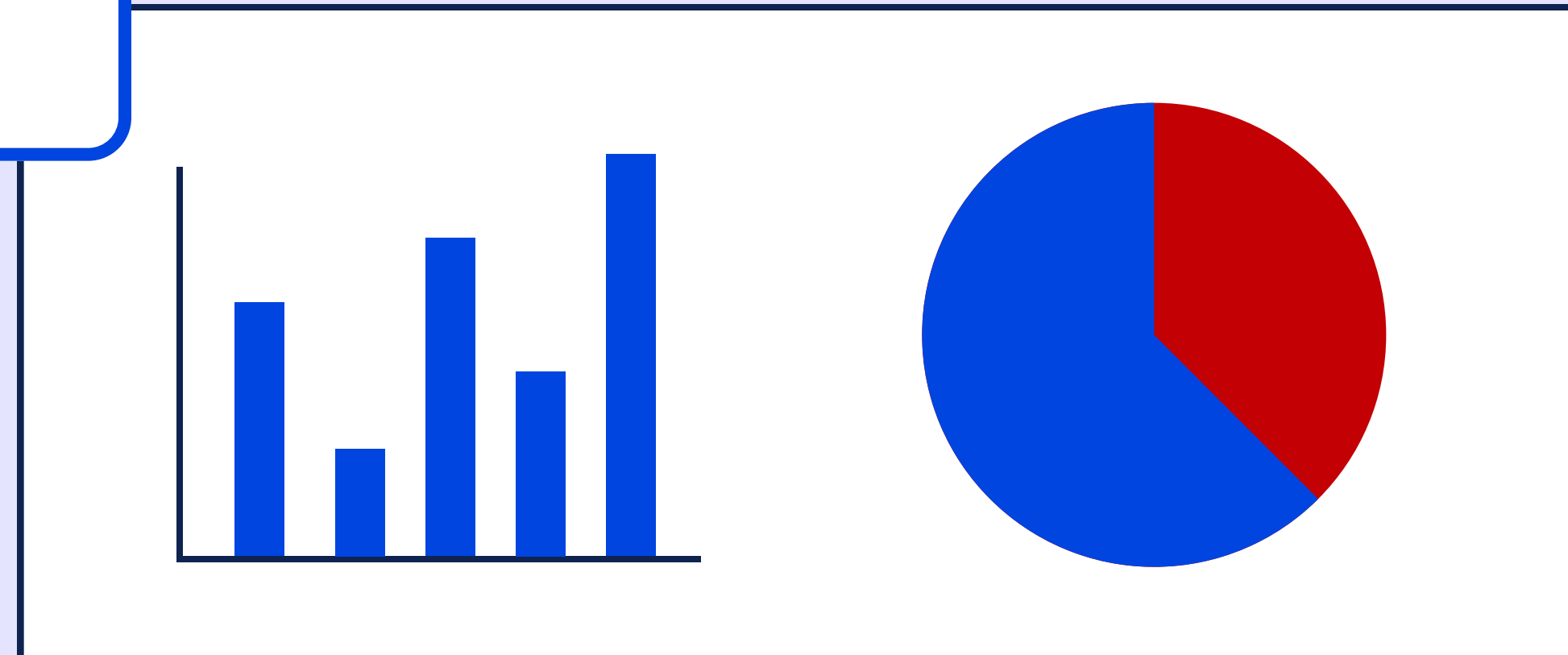
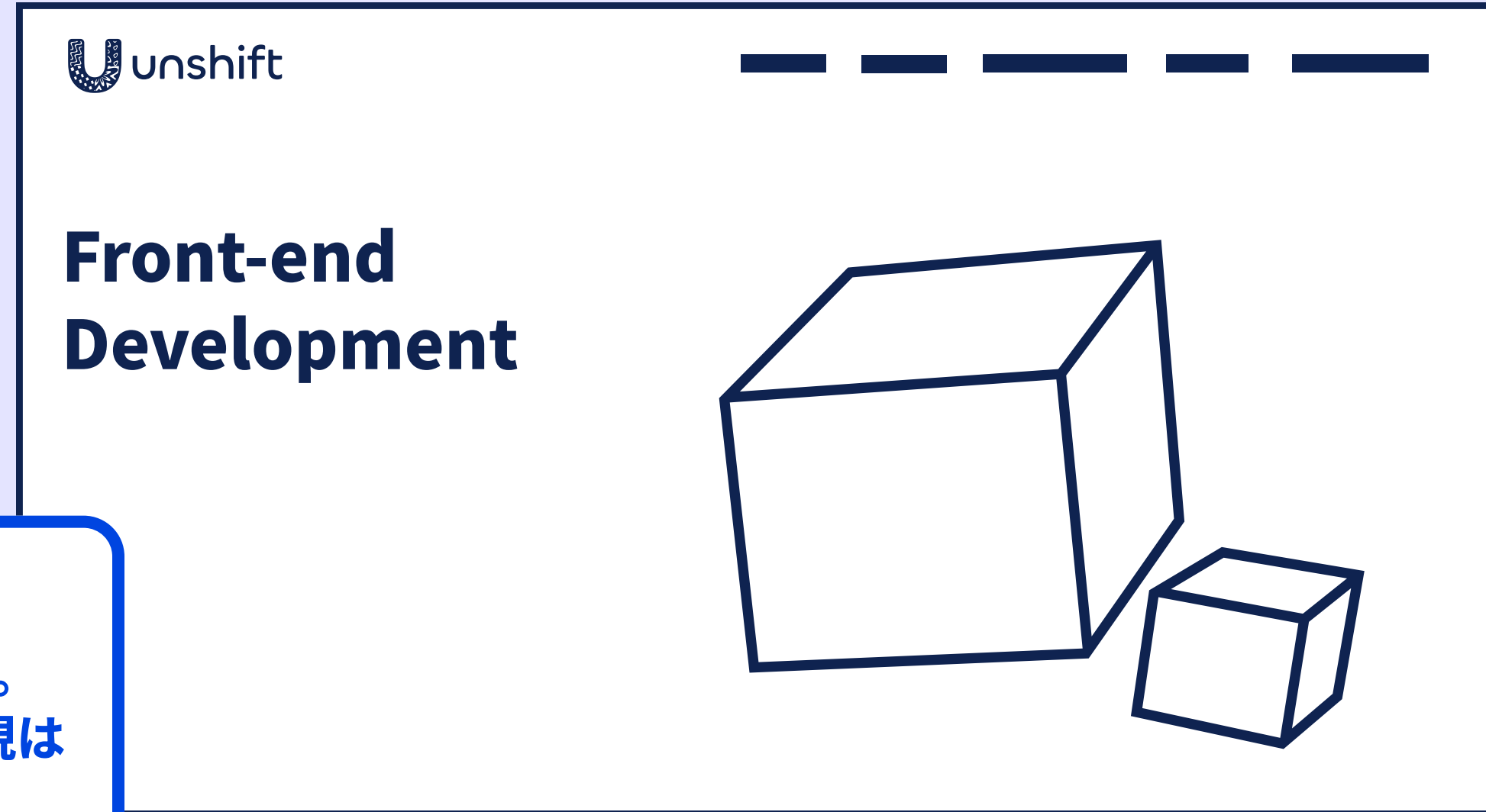
2. エンジニアがWebサイトを実装するときに気にする点

2-f. リッチなグラフィック表現

どうやって再現するか
(WebGL, Canvas2D,
SVG, CSSなど)

早く決めたいレベル ★★★★★・

3Dをリアルタイムに
動かしたいならWebGL。
動的なグラフなどの表現は
Canvas2D, SVG。
単純な表現ならCSS。
静的であれば画像。

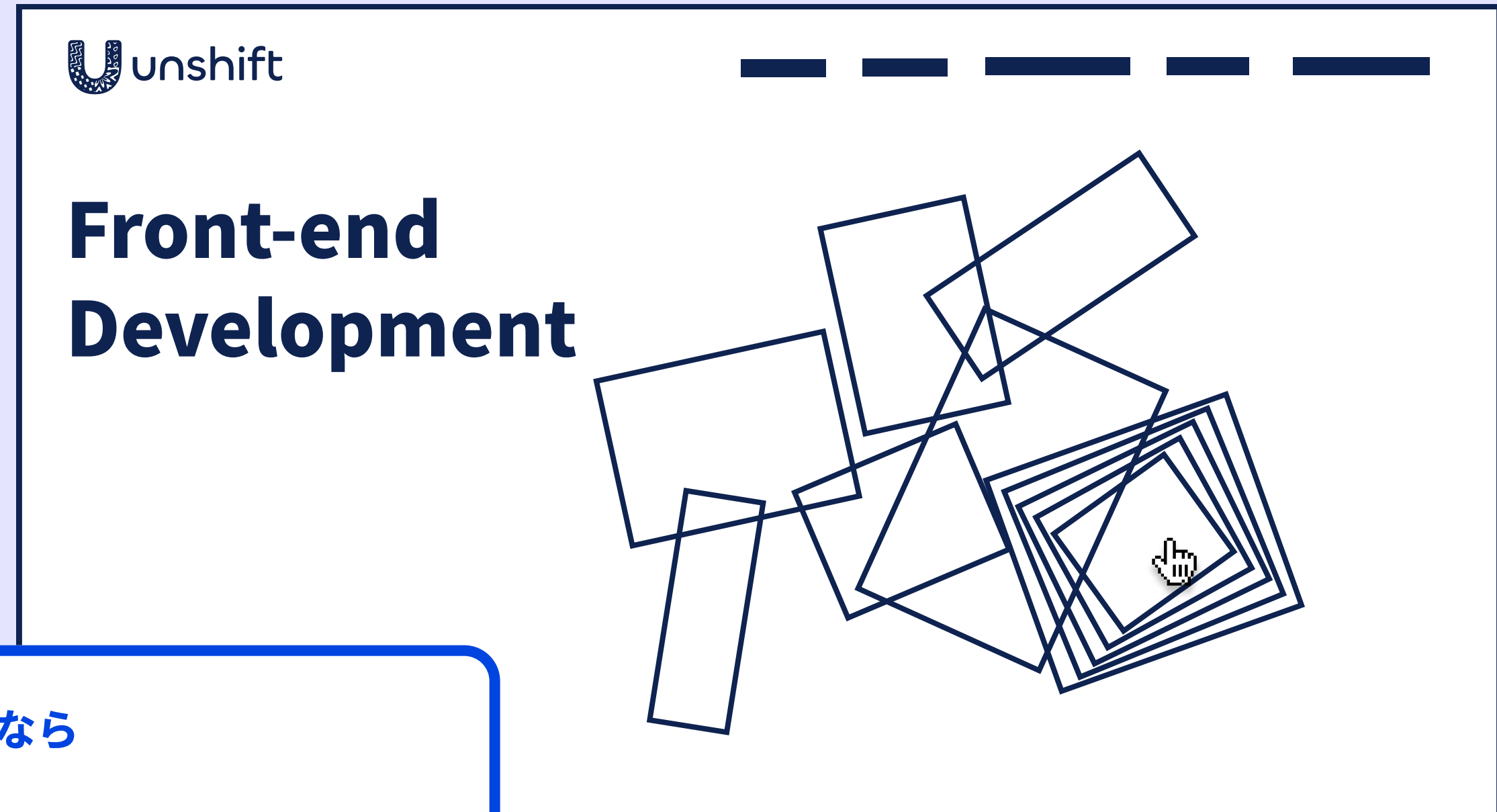


2. エンジニアがWebサイトを実装するときに気にする点

2-f. リッチなグラフィック表現

インタラクションはあるのか

早く決めたいレベル ★★★★★



インタラクションがないのなら
動画でもいいのでは？
ただ、動画を作れる人を追加でアサインしないと。

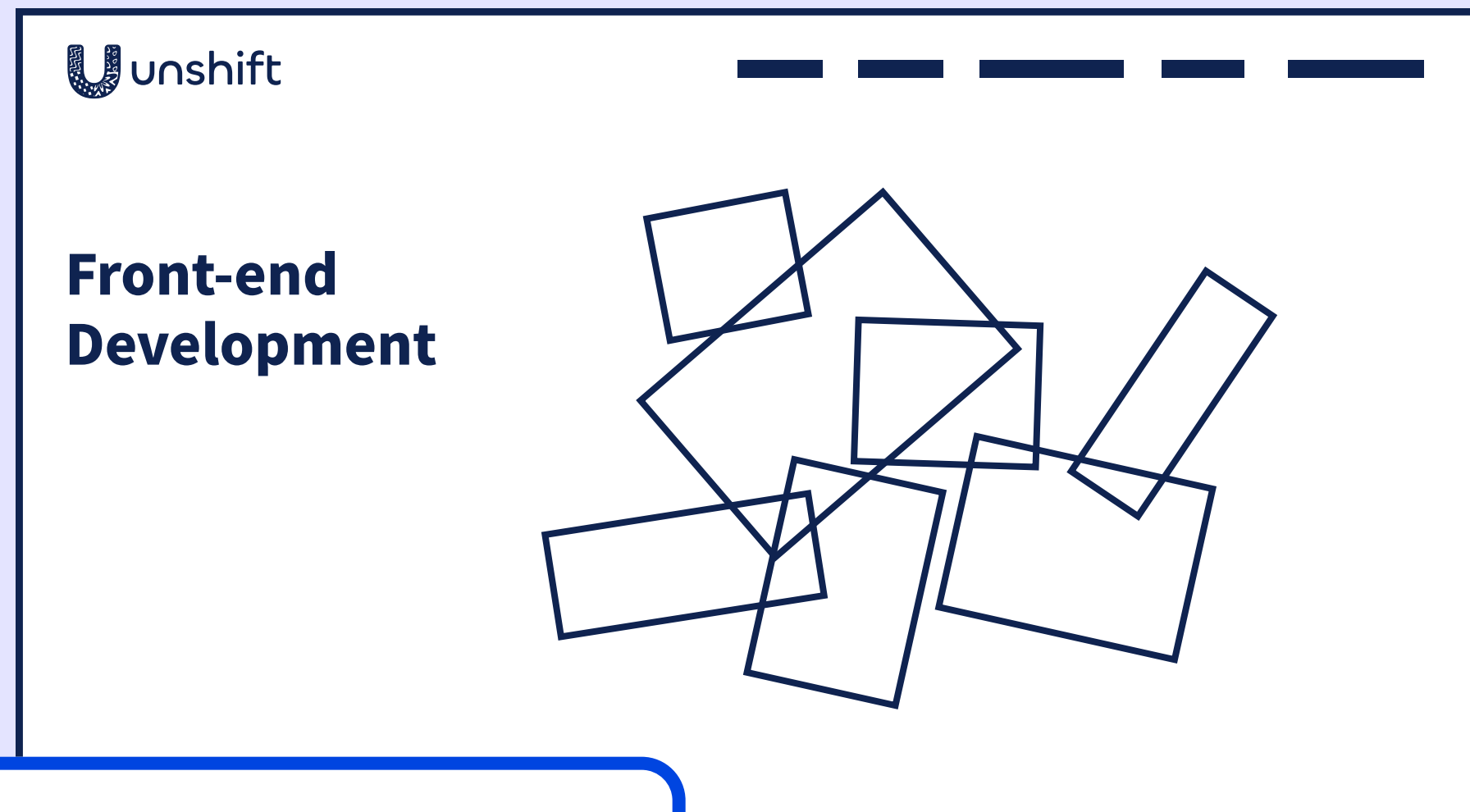
2. エンジニアがWebサイトを実装するときに気にする点

2-f. リッチなグラフィック表現

スマホでも
同じ表現をするのか

早く決めたいレベル ★★★★★

PC



Mobile



同じ演出をしつつも
モバイルのときは少し演出を
簡易的にする必要がありそう？
そもそもモバイルではやらないほうがいい？

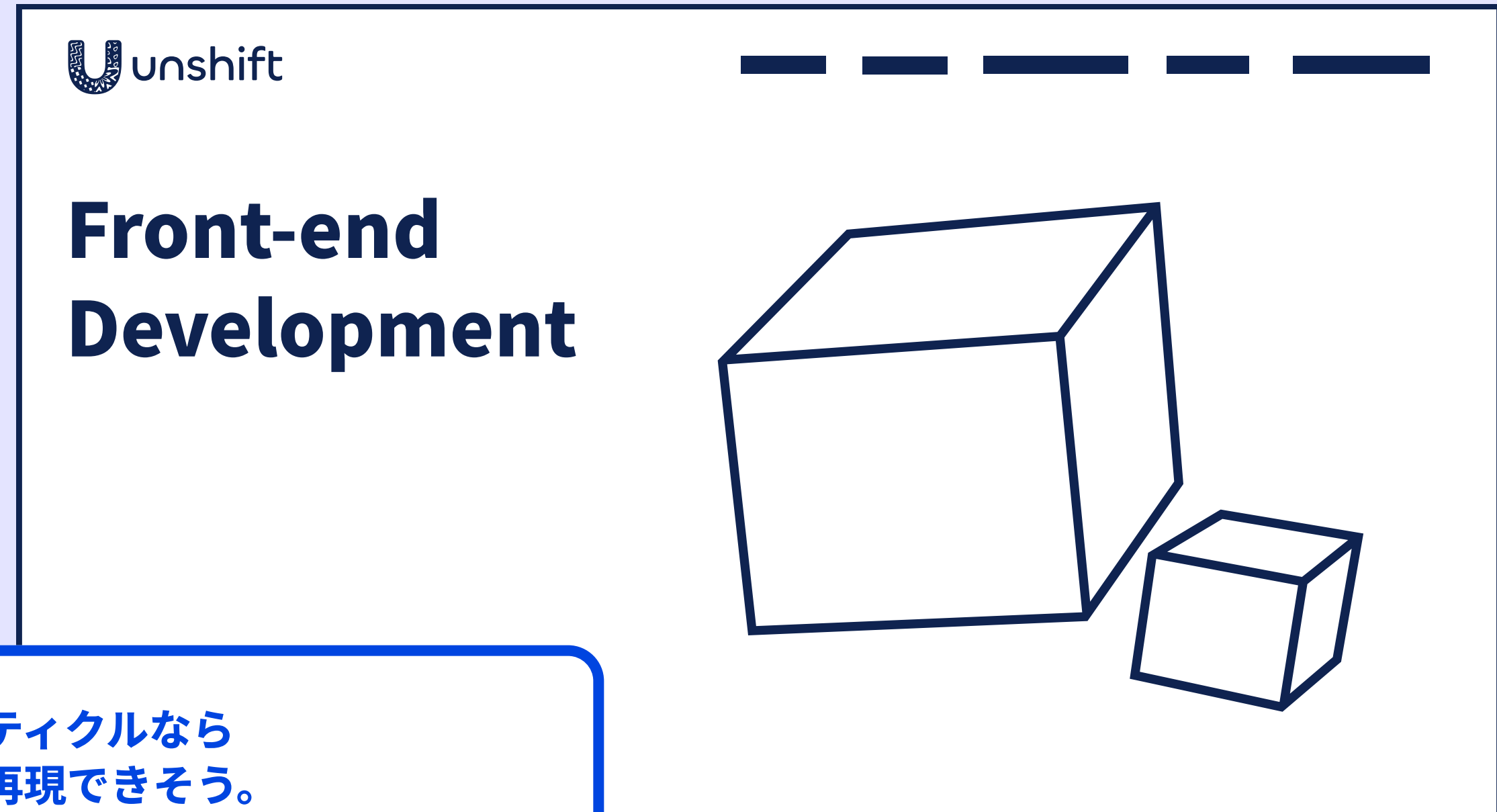
2. エンジニアがWebサイトを実装するときに気にする点

2-f. リッチなグラフィック表現

3D表現がある場合

3Dモデルが必要になるのか

早く決めたいレベル ★★★★★



幾何学表現やパーティクルなら
プログラミングで再現できそう。
具体的なモチーフがあるなら3Dモデルが必要。
板ポリに画像を貼りつけてそれを3D空間で
動かすなら、3Dモデルは不要そう。

2. エンジニアがWebサイトを実装するときに気にする点

g. UI・コンテンツの状態

2. エンジニアがWebサイトを実装するときに気にする点

2-g. UI・コンテンツの状態

フォームがある場合 バリデーションのエラー表示は どうなるのか

早く決めたいレベル ★★ . . .

エラーメッセージや
バリデーションの使用も決める必要あり。
エラーがあるときは
送信ボタンもdisabledに。

お名前 (必須)

山田 太郎

メールアドレス (必須)

aaaaaaaaaa

正しい形式で入力してください。

送信する

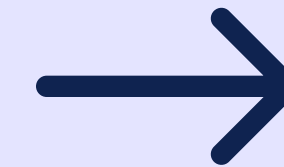
2. エンジニアがWebサイトを実装するときに気にする点

2-g. UI・コンテンツの状態

**フォームがある場合
確認画面や
完了画面があるか**

早く決めたいレベル ★★★★★

The screenshot shows a contact form on the Unshift website. At the top left is the Unshift logo, and at the top right is a hamburger menu icon. The form consists of three input fields, each with a red label indicating it is required: "お名前 (必須)" (Name), "お名前 (必須)" (Name), and "お問い合わせ内容 (必須)" (Inquiry content). Below the fields is a dark blue button labeled "送信する" (Send).



The screenshot shows a confirmation screen on the Unshift website. At the top left is the Unshift logo, and at the top right is a hamburger menu icon. The text reads "お問い合わせ
ありがとうございます" (Inquiry received, thank you). Below this is a dark blue button labeled "トップページへ戻る" (Return to top page).

送信完了は完了画面があるのか。
確認画面はあるのか。
送信後完了画面へ
非同期で遷移するなら、
フォーム送信用APIが必要、など。

2. エンジニアがWebサイトを実装するときに気にする点

2-g. UI・コンテンツの状態

読み込み中や
読み込み完了などの
表示があるか

早く決めたいレベル ★★★★★

送信中

The screenshot shows a mobile app interface for 'Unshift' with a dark blue background. At the top left is the 'Unshift' logo, and at the top right is a hamburger menu icon. The form contains three input fields, each with a red label '(必須)'. The first two are labeled 'お名前 (必須)'. The third is labeled 'お問い合わせ内容 (必須)'. A white loading spinner is centered over the third field, with the text '送信中' below it. At the bottom, a dark blue button labeled '送信中...' is visible.

送信ボタンのラベルが変更になる。
オーバーレイでステータスが表示される。
トースト表示される。など。

送信完了 (下部のトースト通知)

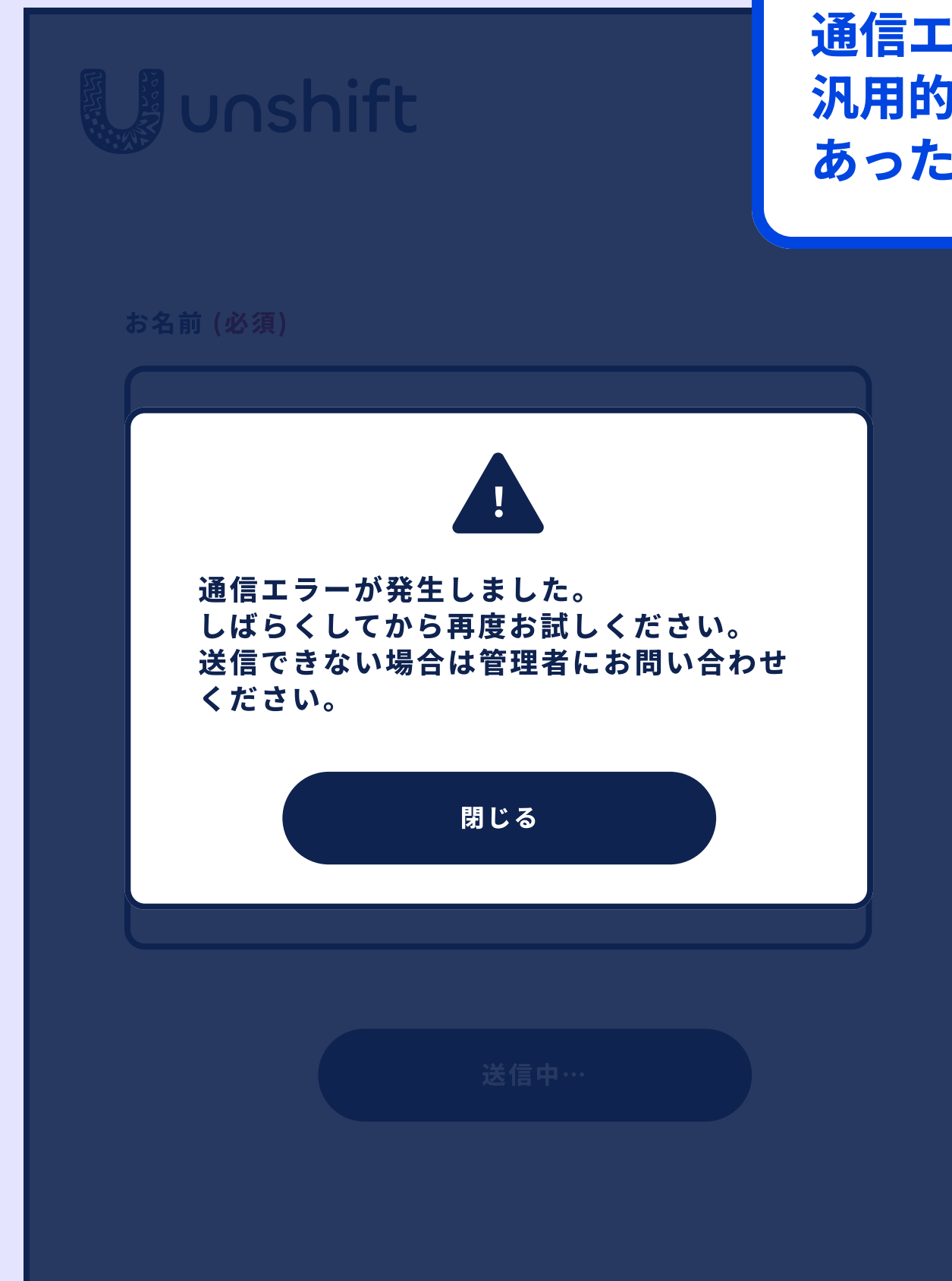
The screenshot shows the same mobile app interface as the previous one, but with a light blue background. The form fields are now empty. The button at the bottom is now labeled '送信する'. A light blue toast notification is displayed at the bottom of the screen, containing the text '送信が完了しました' and a close button '×'.

2. エンジニアがWebサイトを実装するときに気にする点

2-g. UI・コンテンツの状態

通信が発生する場合 通信エラーが発生した場合の 表示があるか

早く決めたいレベル ★★ . . .



通信エラーなど
汎用的に使える異常通知は
あったほうがいい。

2. エンジニアがWebサイトを実装するときに気にする点

h. その他

2. エンジニアがWebサイトを実装するときに気にする点

2-h. その他

**既存サイト内の
一部のコンテンツ制作の場合
共通ヘッダ・フッタがあるか**

早く決めたいレベル ★★★★★・

**CMSで管理する箇所がある場合
コンテンツがCMSの要件に合うか
運用上破綻しないか**

早く決めたいレベル ★★★★★・

**Webアクセシビリティには
どのレベルまでどう対応するか**

早く決めたいレベル ★★★・・

非同期でページ遷移するか

早く決めたいレベル ★★★★★・

**スクロールバーは
デフォルトの見た目がいいのか
独自実装のものなのか**

早く決めたいレベル ★★★・・

2. エンジニアがWebサイトを実装するときに気にする点

まとめ

**デザインカンプは特定の状態のスナップショット※でしかないので
さまざまな変化を
網羅的に仕様に落とし込んで
実装する必要があることを理解しておく。**

※ある特定の瞬間を切り取った静的な画像や記録のこと

Index

1

まずは、プロジェクトを成功させるという心構えが必要

済

2

エンジニアがWebサイトを実装するときに気にする点

済

3

エンジニアとの共通言語を増やすための基礎知識

4

やり取りの際に意識したほうがよいこと

3. エンジニアとの共通言語を増やすための基礎知識

**エンジニアと
仕様を詰める会話をする際に
共通言語を使えると
やり取りがよりスムーズに**

3. エンジニアとの共通言語を増やすための基礎知識

- a. OS・デバイス・ブラウザの違いによる挙動や見た目の違い
- b. Webサイトで使用するフォント
- c. アニメーション
- d. CSSの単位
- e. 3Dの基礎知識 (おまけ)

3. エンジニアとの共通言語を増やすための基礎知識

a. OS・デバイス・ブラウザの違いによる 挙動や見た目の違い

3. エンジニアとの共通言語を増やすための基礎知識

3-a. OS・デバイス・ブラウザの違いによる挙動や見た目の違い

- **スクロールバーの違い**
- **操作方法の違い**
(トラックパッド、タッチパネル、マウスなど)
- **プリインストールされているフォントが異なる (後述)**

3. エンジニアとの共通言語を増やすための基礎知識

3-a. OS・デバイス・ブラウザの違いによる挙動や見た目の違い

スクロールバーの違い

3. エンジニアとの共通言語を増やすための基礎知識

3-a. OS・デバイス・ブラウザの違いによる挙動や見た目の違い

- スクロールバーの違い



コンテンツにオーバーレイするタイプ



ビューポート幅を圧迫するタイプ

3. エンジニアとの共通言語を増やすための基礎知識

3-a. OS・デバイス・ブラウザの違いによる挙動や見た目の違い

- スクロールバーの違い

**スクロールバーは
OSで異なるのはもちろんだが
OSの設定、ブラウザでも
見た目・挙動がかなり異なる。**

3. エンジニアとの共通言語を増やすための基礎知識

3-a. OS・デバイス・ブラウザの違いによる挙動や見た目の違い

操作方法の違い
(トラックパッド、タッチパネル、マウスなど)

3. エンジニアとの共通言語を増やすための基礎知識

3-a. OS・デバイス・ブラウザの違いによる挙動や見た目の違い

- 操作方法の違い

**マウスオーバーでなにか要素を
表示できるようなUIを考えるときは
タッチパネルでも問題なく
操作できるUIになっている必要がある。
(実演: AppleのWebサイトのメガメニュー)**

3. エンジニアとの共通言語を増やすための基礎知識

b. Webサイトで使用するフォント

3. エンジニアとの共通言語を増やすための基礎知識

3-b. Webサイトで使用するフォント

デバイスフォント

- デバイスにプリインストールされているフォントのこと
- OSによってプリインストールされているフォントが異なる
- Androidに至ってはデバイスによっても異なる
- AndroidにはSerif体 (明朝体) のフォントが入っていないことがある。
- macOSは、プリインストールフォントとして入っているはずのフォントが有効になっていない場合がある。

3. エンジニアとの共通言語を増やすための基礎知識

3-b. Webサイトで使用するフォント

Webフォント

- **Google FontsやAdobe Fontsのような配信サービスを使用するか自分でサーバにフォントファイルをアップロード(セルフホスティング)して使用する。**
- **セルフホスティングの場合はライセンスに注意(商用として使えるか、Webフォントとして使えるかなど)。**
- **日本語Webフォントは便利だが、一般的にはファイルサイズが大きいため読み込みに時間がかかる。**
- **Google Fontsの日本語フォントは、読み込みに時間がかからないように最適化されている。**

3. エンジニアとの共通言語を増やすための基礎知識

3-d. 3Dの基礎知識

**日本語Webフォントは便利だが
一般的には読み込みに時間がかかるなどの
デメリットも大きいいため
「とりあえず入れておけばよい」
というものではない。**

3. エンジニアとの共通言語を増やすための基礎知識

3-d. 3Dの基礎知識

また、同じフォントでもOSによって見た目が異なる場合があるので注意。

3. エンジニアとの共通言語を増やすための基礎知識

3-d. 3Dの基礎知識

**デバイスフォントとWebフォントの特性を
理解して使い分ける。**

3. エンジニアとの共通言語を増やすための基礎知識

c. アニメーション

3. エンジニアとの共通言語を増やすための基礎知識

3-c. アニメーション

CSSでの transition の指定の仕方

※JSのアニメーションライブラリなどでも同じ用語を使う。

```
.foo {
```

```
  transition: all 0.4s ease-out 0.2s;
```

```
}
```

duration
(再生時間)

easing function
(イージング関数)

delay
(待ち時間)

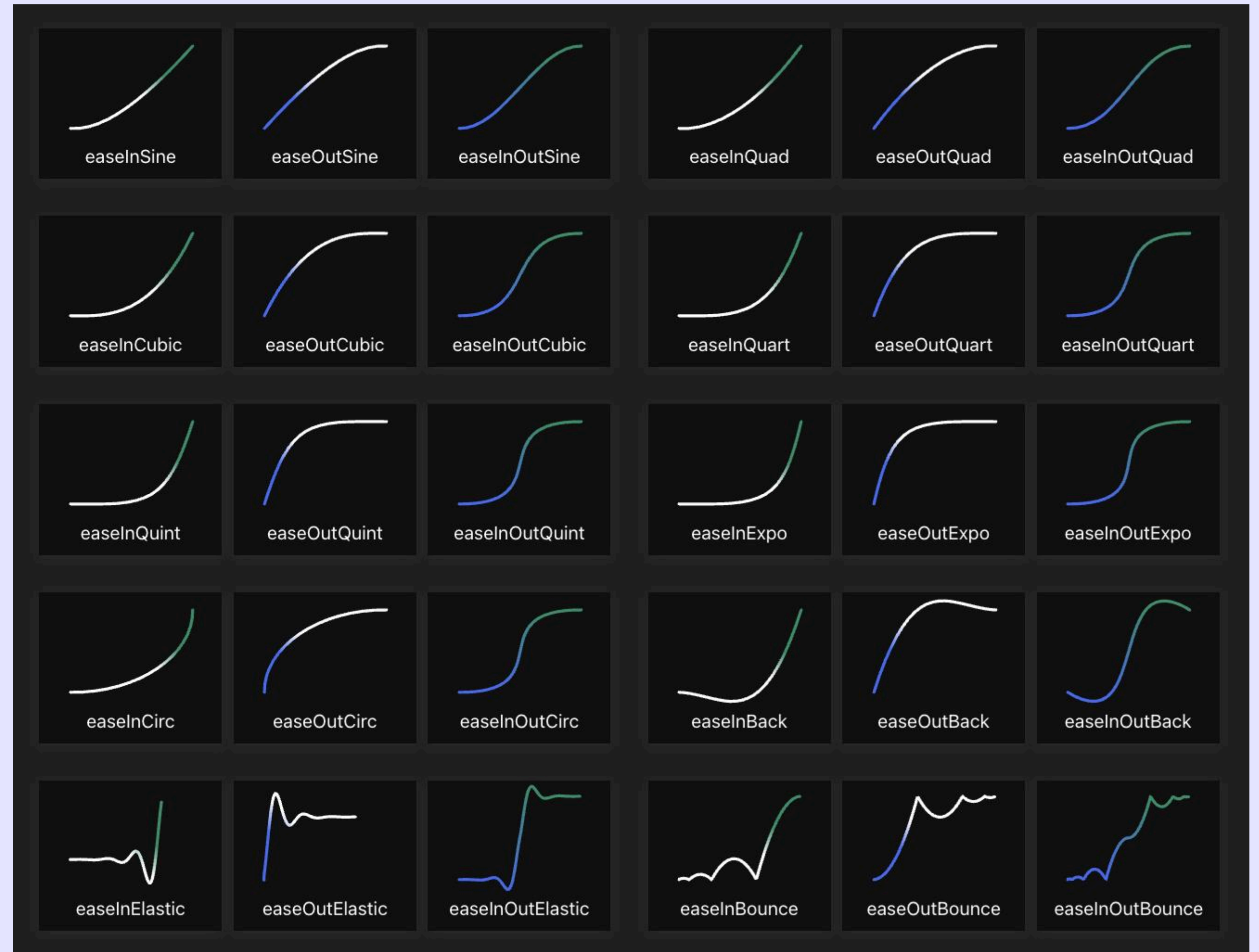
3. エンジニアとの共通言語を増やすための基礎知識

3-c. アニメーション

イージング

イージングとは、アニメーション進行具合 (速度の変化) を制御するための数学的な関数。イージング関数ともいう。

同じduration, delayのアニメーションでも、イージングが異なると与える印象が全く異なるアニメーションになる。



※参照: イージングチートシート (<https://easings.net/ja>)

3. エンジニアとの共通言語を増やすための基礎知識

3-c. アニメーション

**アニメーションの用語
(duration, easing, delay)と
イーザングの種類を覚えておくと
実装時の値の話直接的に
できるので話が早い。**

3. エンジニアとの共通言語を増やすための基礎知識

d. CSSの単位

3. エンジニアとの共通言語を増やすための基礎知識

3-d. CSSの単位

親要素や

ビューポート(≒ウィンドウ)の
サイズによって

相対的に決まる値の単位

%, vw, vh, cqw, cqh など

親要素や

ビューポート(≒ウィンドウ)の
サイズによって

変動しない絶対的な値の単位

px, em, rem[※] など

※em, remは要素のフォントサイズが基準となる単位

3. エンジニアとの共通言語を増やすための基礎知識

3-d. 3Dの基礎知識

単位自体をおぼえるというよりは
ウィンドウやコンテンツのサイズによって
相対的に変化するのか
それらによらない絶対的な値があり
それぞれどういう振る舞いをするかを
知っておくと良い。

3. エンジニアとの共通言語を増やすための基礎知識

e. 3Dの基礎知識 (おまけ)

3. エンジニアとの共通言語を増やすための基礎知識

3-e. 3Dの基礎知識

- **透視投影 (Perspective)、平行投影 (Orthographic)**
- **視野 (FOV / Field Of View)**
- **テクスチャ**
- **UVマップ (3Dの展開図)**
- **ライトの種類 (平行光源、点光源、環境光など)**
- **環境マップ (反射表現などを使用するときを使う)**
- **GLTF/GLB (ファイル形式)**

などなど、このあたりを知っていると会話しやすい(かも)

3. エンジニアとの共通言語を増やすための基礎知識

3-e. 3Dの基礎知識

**Webサイトの演出として
3DCGを使いたいときに
ある程度の知識があると
実装するエンジニアや3DCGの制作者とも
やり取りしやすくなる。**

3. エンジニアとの共通言語を増やすための基礎知識

まとめ

**OS・デバイス・ブラウザごとの
挙動の違いやCSSの仕様を理解し
用語なども知っておくと…
(加えて3Dの知識もあるとなお良い)**

3. エンジニアとの共通言語を増やすための基礎知識

まとめ

**エンジニアとの
やり取りのコストが減らせて
かつ実装フレンドリーな
設計ができるかも**

Index

1

まずは、プロジェクトを成功させるという心構えが必要



2

エンジニアがWebサイトを実装するときに気にする点



3

エンジニアとの共通言語を増やすための基礎知識



4

やり取りの際に意識したほうがよいこと

4. やり取りの際に意識したほうがよいこと

その1.

なるべく具体的に伝える

その2.

**あとで参照したいものは
参照しやすい場所に「ストック」する**

4. やり取りの際に意識したほうがよいこと

その1.

なるべく具体的に伝える

4. やり取りの際に意識したほうがよいこと

その1. なるべく具体的に伝える

**どこまで詳細に伝えて
どこまでエンジニアに任せるかは
エンジニアのスキルや関係性による。**

4. やり取りの際に意識したほうがよいこと

その1. なるべく具体的に伝える

- **テキストで伝える**
- **リファレンスのサイトで伝える**
- **絵コンテのように数コマで伝える**
- **映像で伝える**

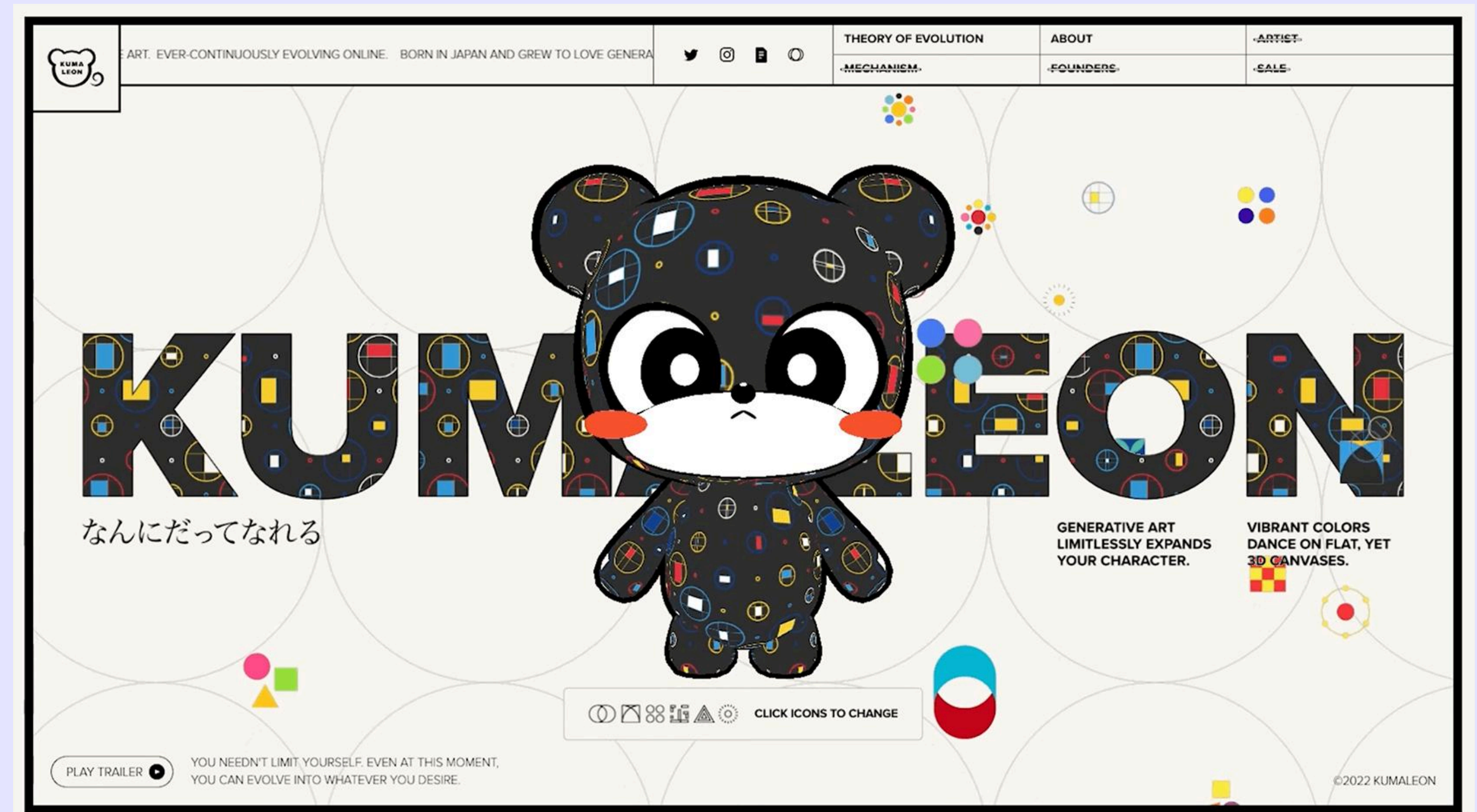
4. やり取りの際に意識したほうがよいこと

その1. なるべく具体的に伝える

- 動きのイメージを伝える手段

動きのイメージを
テキスト +
リファレンスのサイト +
絵コンテで伝える

(例) KUMALEON



KUMALEON (<https://kumaleon.com/>)

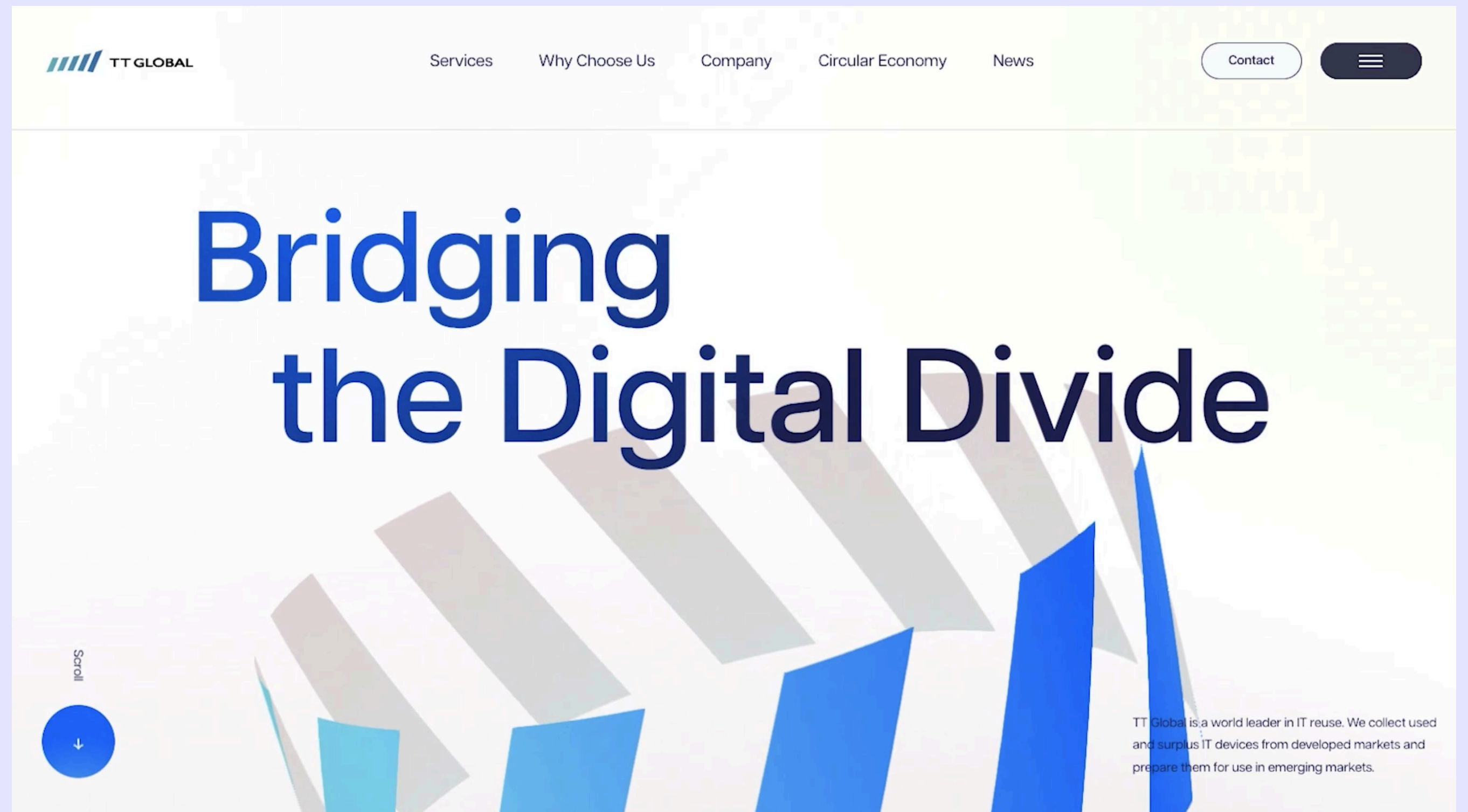
4. やり取りの際に意識したほうがよいこと

その1. なるべく具体的に伝える

- 動きのイメージを伝える手段

動きのイメージを
動画 (After Effectsで制作)
で伝える

(例) TT Global



TT Global (<https://ttglobal.com/>)

4. やり取りの際に意識したほうがよいこと

その1. なるべく具体的に伝える

**なるべく詳細を具体的に伝えると
デザイナーの頭の中のイメージと
実際に実装されたものの誤差が小さくなる。
ビジュアルで伝えるとより効果的。**

4. やり取りの際に意識したほうがよいこと

その2.

**あとで参照したいものは
参照しやすい場所に「ストック」する**

4. やり取りの際に意識したほうがよいこと

その2. あとで参照したいものは参照しやすい場所に「ストック」する

重要な決定事項



unshift 10:18

@designer

じゃあdurationは 0.4s, easingはeaseOutExpoにしておおきます。



pdm 10:24

@chnnel 明日の打ち合わせですが、11:00~12:00でお願いします！



unshift 10:30

@pdm 承知しました！

テキストチャットのような
フロー型の
コミュニケーションツールだと
重要な決定事項が流れてしまう。

検索はできるけど面倒。

4. やり取りの際に意識したほうがよいこと

その2. あとで参照したいものは参照しやすい場所に「ストック」する

**情報の種類によって
ストック先を使い分けるとよい。**

(修正などのタスクレベルであれば、タスク管理ツールに
UIごとのリファレンスURLであれば、でデザインデータに直接貼り付け
不変の仕様であれば、wikiやスプレッドシートなど)

**ストック先が複数あっても、
チーム内でルールが決まっていれば問題ない。**

4. やり取りの際に意識したほうがよいこと

その1. あとで参照したいものは参照しやすい場所に「ストック」する

**人間は忘れる生き物なので、
とにかくログを残す。
(自分で言ったことすらも忘れる。)**

4. やり取りの際に意識したほうがよいこと

その1. あとで参照したいものは参照しやすい場所に「ストック」する

**ログは参照しやすい場所に
ストックしておくこと
検索する時間を節約できる。**

4. やり取りの際に意識したほうがよいこと

まとめ

**しっかり再現してもらいたい箇所はなるべく具体的に伝え
決定事項は参照しやすい場所に
記載しておくことによって
やり取りの回数と
情報を探す手間を減らすことができる。**

Index

1

まずは、プロジェクトを成功させるという心構えが必要

済

2

エンジニアがWebサイトを実装するときに気にする点

済

3

エンジニアとの共通言語を増やすための基礎知識

済

4

やり取りの際に意識したほうがよいこと

済

このセッションのまとめ

エンジニアとデザイナーの
コミュニケーションが
プロジェクトの進行
制作物の質に
大きく影響を与えることを意識する。

このセッションのまとめ

実装時に必要な情報を把握し

実装に関する基礎知識を

身につけておくと

コミュニケーションのコストが減らせる。

このセッションのまとめ

こだわり抜きたい箇所は
なるべく具体的に伝え
決定したことは参照しやすい場所に
まとめておくとよい。

みんなハッピー

ご清聴ありがとうございました。

セッションの参加者の方に

メールでアンケートをお送りしますので

ぜひご協力ください！