

# UIデザイナーを取り巻く様々な設計

Intrinsic meaning of UI Design

Question みなさん、設計って何をどこまでやっていますか？

デザインは“設計”という意味だけど、  
UXやビジュアル・インタラクションの設計だけ？  
UIデザイナーが関わる設計はそこだけなのではないでしょうか？

というわけで、今回は…

---

UIデザインにおける

ヒトやコンピュータの情報処理・設計について

UIデザイナー視点でまとめてみました。

---

# Profile

Tsuyoshi Higuchi

@tyshgc

- UI Designer
- 趣味でサービスつくったり、ハッカソンに参加したり
- 最近熱いのはDDDと情報処理心理学

毎週日曜朝に

色々な企業・学生の  
エンジニアと

渋谷で趣味開発

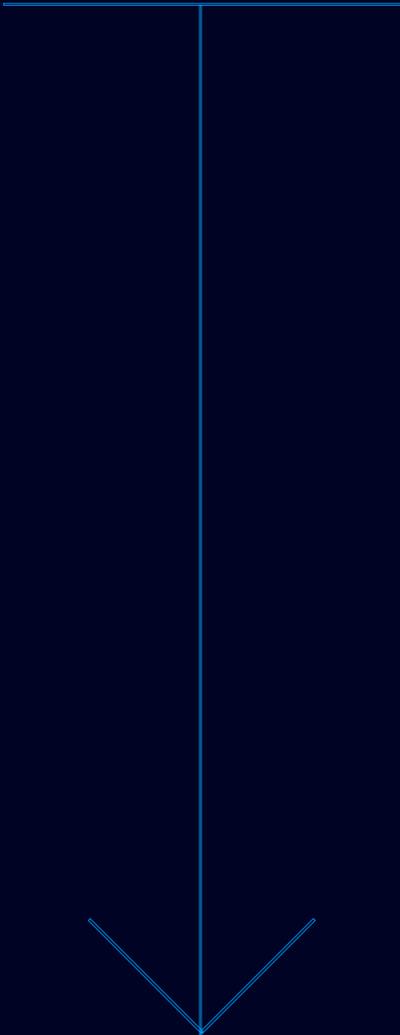
やってます。

デザイナーが少ないので  
興味ある方はぜひ！

devMorning



# Agenda



1. 情報処理のデザイン
2. モデルのデザイン
3. 開発コミュニケーションのデザイン
4. まとめ

# 1. 情報処理のデザイン

# よくみるUXフロー

認知

興味

行動

比較

意思  
決定

利用

UI

UI

UI

UI

UI

この流れの中にUIが介在する

さらに細かくユーザフローを見る

# 記事リーダーアプリで記事を選ぶまでのユーザフロー

Mainflow

起動したらリストがあると思っている

リストセルはタップできるという思い込み

タップすると画面が切り替わっている

アプリを起動する

リスト面を見る

記事をタップする

Subflow

画像などで内容を軽く認知

タイトルなどで内容を軽く認知

記事セルに指でふれる

画面が記事詳細へ切り替わるのを期待

ユーザフローは単一軸ではなく、  
各タスクにあるサブフローを経て次のタスクへ進む

ユーザフローを踏まえて、UIデザインってそもそもなに？

UIデザイン

= ヒトとコンピュータを繋ぐ層のデザイン

Human

User Interface Design

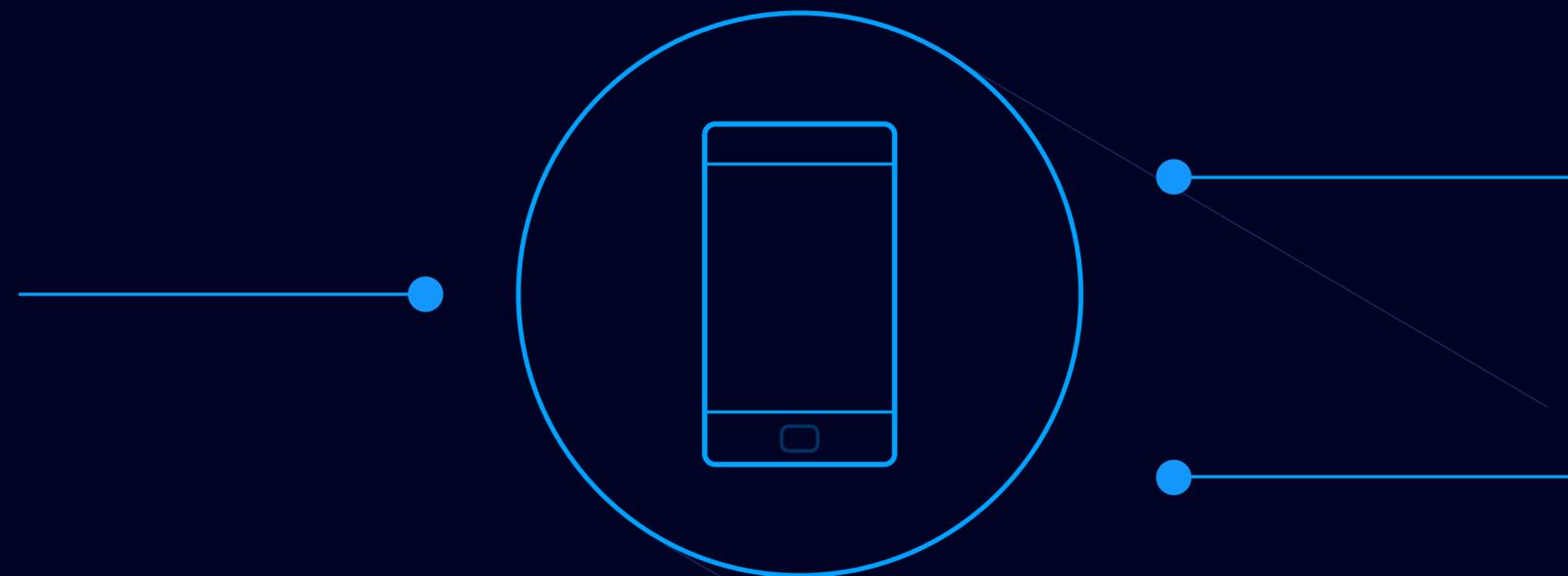
Computer



UIをデザインするということは、  
ユーザの流暢性《認知や処理のしやすさ》  
を向上させること



ヒトとコンピュータ  
どちらの仕組みも知る必要があるのでは？



非エンジニアの考える  
コンピュータの情報処理の仕組みは？



ちょっと、コンピュータの処理をブラックボックスにしすぎ...

もうちょっと詳しく見てみる…

例えばWebフロントエンドのアーキテクチャ

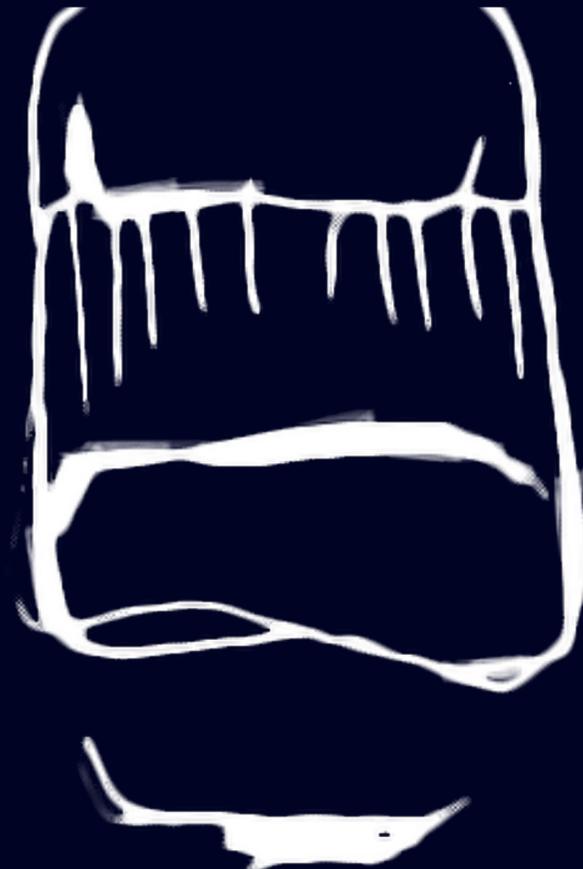
# Fluxのアーキテクチャだとこんな感じ



FluxはFacebookが提案した一方向型のアーキテクチャ



…見えるぞ！私にも流れが見えるッ



WebAPI取得時に待機アニメーションで  
遅延を感じさせない工夫できるかも？

とか

Storeに持つデータはどんなものがあるのか見えた。  
足りないフラグを追加してもらわねば…

とか

コンポーネントを意識してデザインし、実装コストを抑えれば  
別の大切な部分にエンジニアリングを注力できるかも？

とか

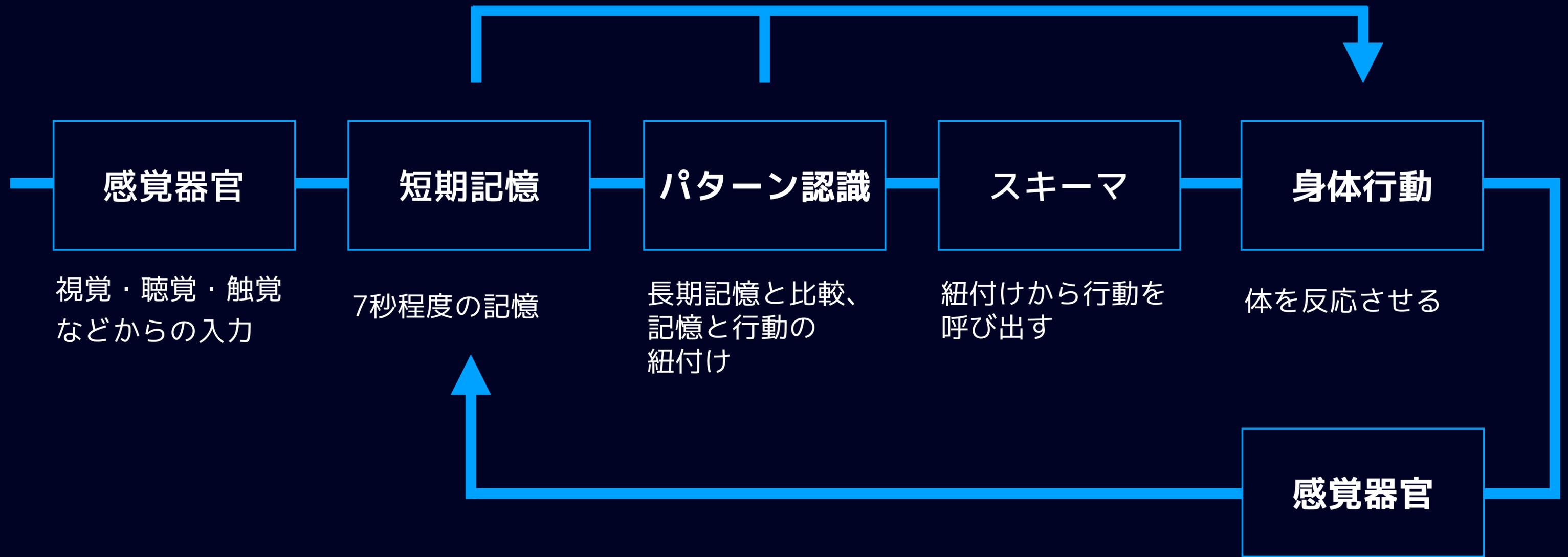


ヒトってどうやって動いているのだろう？



脳、有能！  
でも、そんな単純なはずは…

# 人間情報処理のおおよそなフロー



もう少し実際は複雑ではあるが、ヒトの情報処理もデータ駆動

人間が認識するまでの最適化が考えられるから、  
例えば、ディテールはアフォーダンスを踏まえて設計できるよね？

とか

カラーリングはパターン認識。そこから得られる意味情報（雰囲気）  
を無意識下で視覚野になげれば、ブランドのサポートになるかも？

とか

繰り返し学習を行う・体験をすることで長期記憶に残ってほしい  
ラベリングの意味付けとかできるかもしれないね？

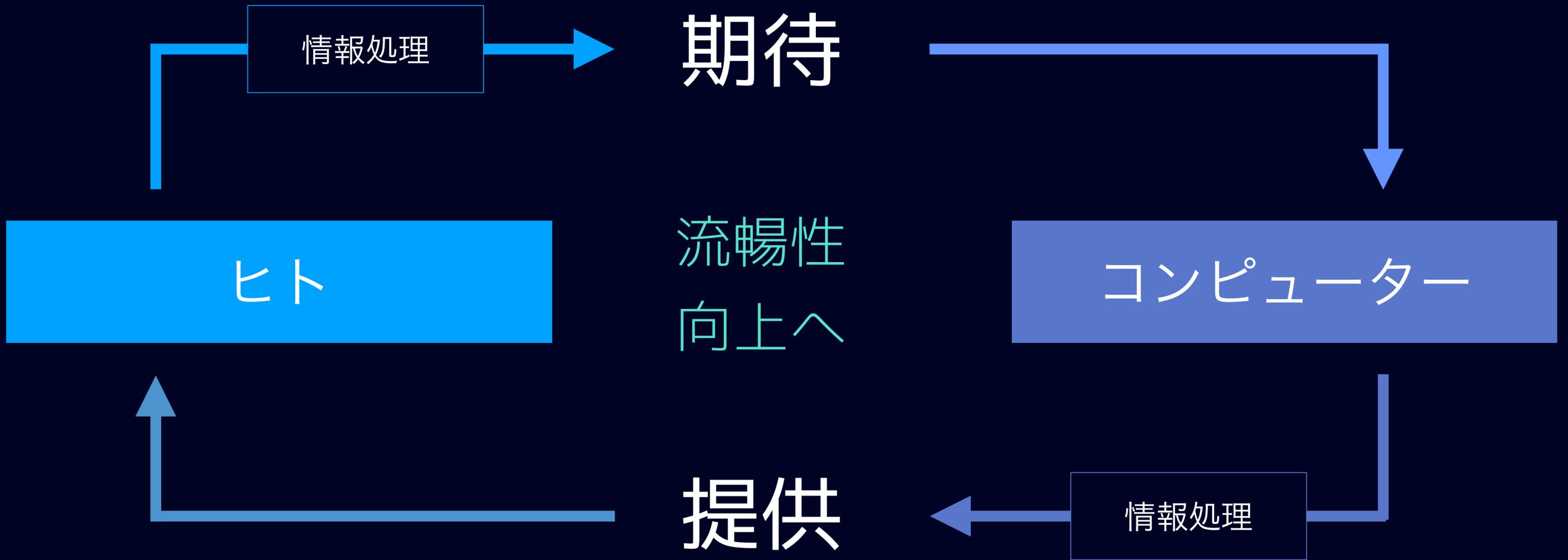
とか

UIデザイン = システムとヒトを繋ぐ層のデザイン

どちらにも細かい情報処理のフローがあることがわかった！

Human User Interface Design Computer

ヒューマンコンピュータインタラクション



価値を提供するために両サイドの仕組みを知り  
最適化されたUser Interfaceを設計する

UIデザイン

=

二者の職域にまたがる設計

Designer

User Interface Design

Engineer



いろいろな人が  
職域を超えて協業するには…  
モノやコトの本質を抽出して  
共有することが近道なのでは？

## 2. モデルのデザイン

サービスは、“利用者のメンタルモデル”を基にした  
“ドメインモデル”の抽象化

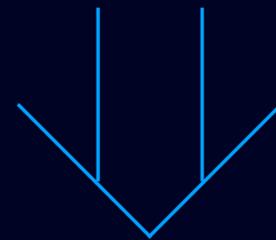
## メンタルモデル

物事の動作がどうやって起こるのかに対する  
`心や意識の概念モデル`

## ドメインモデル

サービスにあるべき  
`ビジネスの実体と業務の振る舞いの概念モデル`

サービスデザイン = 何がサービスのコアなのかの抽出が大切



- サービスの本質に基づいて機能や画面が設計されるべきでは？
- 画面（GUI）駆動開発はボトムアップなので、プロダクトの規模によってはアンチパターンになるかもしれない...

ユーザーの満足度を上げて使い続けてもらおうということは…

サービスに関わる

デザイナー、エンジニア、ビジネスパーソンが協業して、

どこに本質的課題があるか探していく長い旅

利用者のメンタルモデル から得られた

ドメインモデル の抽象化で

より本質的な課題の解決を論理的に行う

どうやってモデルを抽出するのか？

コミュニケーションから  
抽出できるらしい

### 3. 開発コミュニケーションのデザイン

エンジニアのみなさんの多くは…

ドメインモデルをコードの構造に落とし込み、  
共有するアーキテクチャをすでに実践しています

# DDD = Domain Driven Design

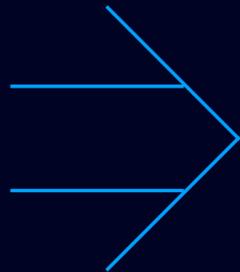
ドメイン駆動設計

---

ドメイン駆動設計を大雑把にいうと

ドメインモデルを設計の基盤とし、実際のコードへ落とし込み  
ドメインエキスパートやプロジェクトメンバと認識を共有する

ドメインモデルとして抽象化された  
サービスの関心事を  
コードに書いて、コードを読んで皆が理解する



```
type User struct {  
    FirstName string  
    LastName string  
    Age int  
    Hobbys []string  
}
```

また、ドメインモデルの抽出を行う際に、  
プロジェクトメンバーやドメインエキスパートなどと  
共通で使用する言葉の定義をする

例：メッセージャーサービスの場合

- メッセージャー is 自転車に乗って配送業務を行う人
- 顧客 is 配送元や配送先の企業
- 配送 is 配送元の荷物を配送先へ届ける行為



定義

共通に認識できる言語をきめる = ユビキタス言語

**共通言語をデザインできれば  
モデルの抽出化は行いやすくなり  
本質的なモデルの抽出が行えれば  
UIデザインも本質的になる**

# ドメイン駆動設計の参考リンク・書籍

ドメインモデルとユビキタス

ユビキタス言語とドメインモデル、そしてモデル探索のうずまき

<http://j5ik2o.me/blog/2013/05/07/domain-model/>

SNSチームでのドメイン駆動設計の実践

<http://labs.gree.jp/blog/2013/12/9330/>



**エリック・エヴァンスのドメイン駆動設計  
(IT Architects'Archive ソフトウェア開発の実践)**

エリック・エヴァンス (著), 今関 剛 (監修), 和智 右桂 (翻訳), 牧野 祐子 (翻訳)

## 4. まとめ

設計 = デザインは同義

ある目的を達成するための計画をたてて、  
無駄を削ぎ本質を明確にする作業を繰り返すこと

目的 = サービスをユーザに提供し利益を生むこと…

そのためには、ドメインへの理解とユーザへの理解  
それらをコミュニケーションでデザインしていく

人間中心のUIデザインを行うために  
モノやコト・ヒトの本質の理解をすすめるため  
設計に携わっていくことが近道になる  
UIデザイナーという職域にだけ閉じこもるのは…  
サービスにもヒトにも保守的な関わり方

コードをかけなくても理解ができれば  
ドメイン駆動設計のような本質視点の開発を  
エンジニアやドメインエキスパートと  
一緒におこなっていき、  
よりよいサービス提供ができるかもしれません。

少なくとも、  
エンジニアとデザイナーのコミュニケーションの促進や  
より現実に沿ったプロトタイピングなどができます。

UIデザイナーも  
コードや設計と友達になろう！