

# ユーザインタフェース (第1回)

五十嵐 健夫

## 主な内容

Human Computer Interaction (HCI)  
User Interface (UI)

使いやすいソフトウェアをデザインするための方法論

最新のインタフェース研究の紹介

課題を通じたデザインと評価の実践

## Course Credit

Attendance

Assignment  
Programming and User Testing

## Schedule

6/7 Design and Evaluation

6/14 Selected Topics, 課題出題

7/21 User Centered Design by Nolwenn Maudet

6/28 Interactive Computer Graphics, 課題構想発表

7/5 Crowd Sourcing and Human Computation

7/9 課題レポート締切 (深夜)

7/12 課題成果発表

五十嵐 健夫



/ 講義情報 /

## Outline

- Background
- 「Design of Everyday Things」
- Design Process
  - Research, Analysis, Prototyping
- Evaluation Methods
  - Without Test Users
  - With Test Users

## Outline

- Background
- 「Design of Everyday Things」
- Design Process
  - Research, Analysis, Prototyping
- Evaluation Methods
  - Without Test Users
  - With Test Users

## HCI (Human Computer Interaction) とは何か

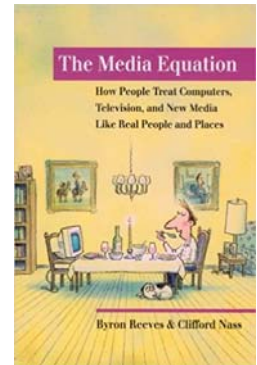
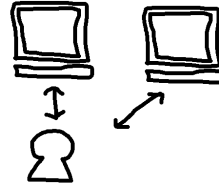
人間と計算機のかかわりに関する学問。  
コンピュータ科学の一分野だが、学際的である。  
計算機科学・認知心理学・デザイン/アート



人間にとって使いやすいインターフェースを開発する。  
計算機を使っている人間の行動について研究する。

## The Media Equation

人間は「情報メディア」に対して、  
あたかも人間に接するのと同じよう  
に接してしまう。



## Why is HCI Important?

- Life
  - 人間の命に関わる
  - 人間生活の質の向上に関わる
- Difficult
  - ソフトウェアの大部分を占める
  - よいものをデザインするのは簡単ではない
- Business
  - 生産性の向上・売上げの増加に直結する
  - ブランドイメージに結びつく

## ユーザビリティとは？

- × 使いやすさ (単なるオプション)
- 使用可能性 (根本的な要求性能の一つ)

### ユーザビリティの定義

「特定のコンテキストにおいて特定のユーザによってある製品が、特定の目的を達成するために用いられる際の、効果・効率・満足度の度合い」

|     |                 |
|-----|-----------------|
| 効果  | 目的の達成可能性        |
| 効率  | 無駄なく最短経路で達成できるか |
| 満足度 | 不愉快な思いをさせないか    |

## Outline

- Background
- 「Design of Everyday Things」
- Design Process
  - Research, Analysis, Prototyping
- Evaluation Methods
  - Without Test Users
  - With Test Users

D.Norman  
“design of everyday things”  
「誰のためのデザイン？」

インタフェースデザインの重要性を訴えた本



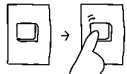
「失敗するのは、ユーザの責任でなくデザイナーの責任」

「デザインの工夫で、効率が上がり失敗がへる」

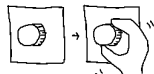
よいデザインをするためのいくつかの知識

よいデザイン、悪いデザインの例とその分析

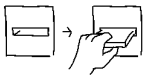
## 「アフォーダンス」 使い方を示唆する特徴



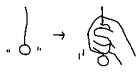
ボタン=押す



ノブ=回す



スロット=差し込む

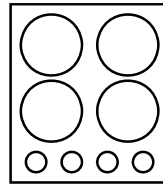


ひも=引く

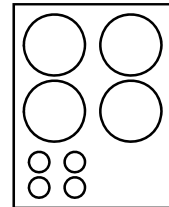
アフォーダンスをうまく使えば  
説明が不要になり誤りが減る。

## 「自然な対応づけ」

ガスのレンジの例



分かりにくい



分かりやすい

## 「自然な対応づけ」

自動車のシート調節の例



分かりにくい



分かりやすい

## 「自然な対応づけ」

ドアの例



開くのはどっち？



開くのは左

## 「可視性とフィードバック」

例) ボタンを恣意的な順に押す

フィードバックなし 要マニュアル

エラー多

ディスプレイあり マニュアル・記憶不要

何が起きているのが見える  
入力に対して適切なフィードバックを返す

## 「エラーの防止」



← チャンネル1  
← チャンネル2  
← チャンネル3  
← チャンネル4  
← 設定の消去!

無線装置の例

人は必ずエラーをする。  
エラーを起こりにくくする&被害を小さくする工夫が必要。

## 「概念モデル」

「物事がどう動作するのか、その原理に関する心の中のモデル」



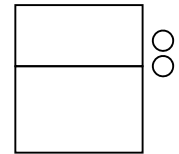
機械の動作の概念モデルをうまく構築できると操作が楽になる。

ただしい概念モデルを提供し、それにあった動作をするように設計するべき。

## 「概念モデル」

|                    |           |
|--------------------|-----------|
| Normal Settings    | C and 5   |
| Colder Fresh Food  | C and 6-7 |
| Coldest Fresh Food | B and 8-9 |
| Colder Freezer     | D and 7-8 |
| Warmer Fresh Food  | C and 4-1 |
| OFF (both)         | 0         |

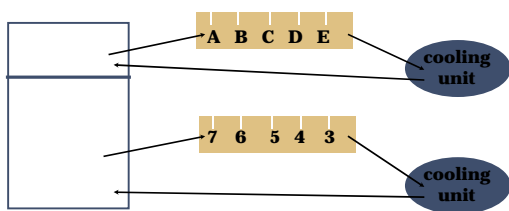
冷蔵庫のスイッチ



|           |           |
|-----------|-----------|
| A B C D E | 7 6 5 4 3 |
|-----------|-----------|

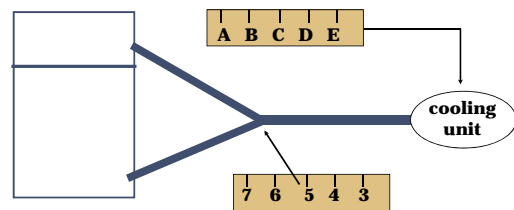
分かりにくいものの例。メンタルモデルを構築できない！

## 「普通に思い浮かべる概念モデル」



独立した制御

## 「この冷蔵庫の実際の動作モデル」



解決法

この様子がわかるように表示を工夫する or 直感的にわかるようにシステムを作り直す

D.Norman

## 「誰のためのデザイン？」

「使いにくいデザインができる理由」

D.Norman

## 「誰のためのデザイン？」

「使いにくいデザインができる理由」

デザインする人はエキスパートになってしまう

(普通のユーザの困難が理解できない)

正しくデザインされていない (テストしていない)

美的基準によって評価される (デザイン賞など)

機能の豊富さが賞賛される。

購入するときにあまり考慮されない。

悪いのはユーザと思いつむ。

...

## D.Norman 「誰のためのデザイン？」

「使いやすいデザインのための原則」

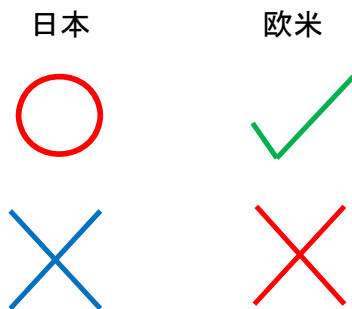
- 外界にある知識を利用する。(ものの置き場所)
- 作業の構造を単純化する。(人間の短期記憶)
- 対象を目に見えるようにする。(冷蔵庫)
- 自然な対応付けを行う。(コンロ)
- 自然の制約や人工的な制約を活用する。(レゴ)
- エラーに備えたデザインをする。(undo)
- 標準化する。(keyboard, 信号, カレンダー)

## B. Shneiderman “Designing the User Interface”

デザインにおいて留意すべき点

- Physical 人間の物理的特性・場所の特性
- Cognitive 認知的・知覚的特性 (e.g. Steven's Power Law)
- Individual 個人差
- Cultural 文化的・国際的な多様性 (e.g. OX、色の意味)
- Impaired 障害者・高齢者・子供 ユニバーサルデザイン  
(e.g. 先天赤緑色覚異常)

## 文化による記号の意味の違い

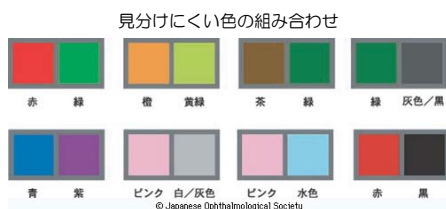


## 文化による色の意味の違い



## 先天赤緑色覚異常

日本人男性の4.50%, 白人男性では約8%



色以外の情報も組み合わせる(文字など)  
色彩だけでなく輝度に違いを持たせる  
色で区別するならば5程度が限界

## 色覚異常



<http://www.vischeck.com/>

## 知覚に関する基礎知識

- Preemptive Perception (意識的な探索を必要としない、自動的・即時的な知覚)を活用する。
- 人間の知覚の特性(歪み)をよく理解する。
- 「注意」の重要性を理解する。

## 知覚に関する基礎知識

- Preemptive Perception (意識的な探索を必要としない、自動的・即時的な知覚)を活用する。
- 人間の知覚の特性(歪み)をよく理解する。
- 「注意」の重要性を理解する。

## Preemptive Perception

```
1281768756138976546984506985604982826762
9809858458224509856458945098450980943585
9091030209905959595772564675050678904567
8845789809821677654876364908560912949686
```

How many 3's ?

[Stasko, Agrawala]

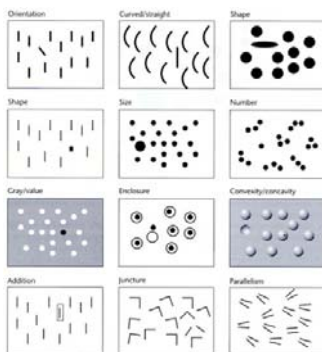
## Preemptive Perception

```
1281768756 3 976546984506985604982826762
98098584582245098564589450984509809 3 35
909 3 209905959595772564675050678904567
8845789809821677654876 3 4908560912949686
```

How many 3's ?

[Stasko, Agrawala]

## Preemptive Features

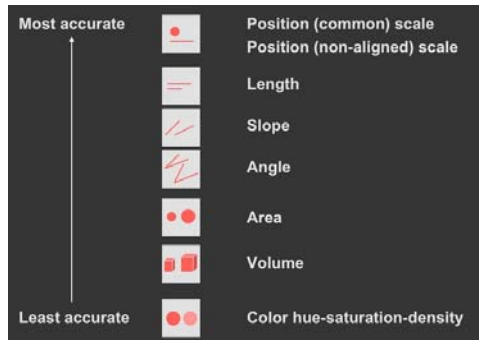


[Information Visualization, Ware 04]

## 知覚に関する基礎知識

- Preemptive Perception (意識的な探索を必要としない、自動的・即時的な知覚)を活用する。
- 人間の知覚の特性(歪み)をよく理解する。
- 「注意」の重要性を理解する。

## Relative Magnitude Estimation

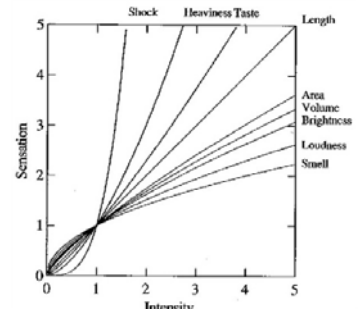


[Agrawala]

## Steven's Power Law

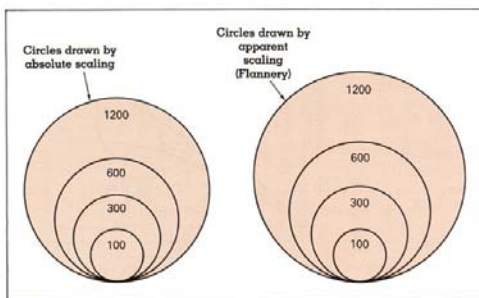
$$S=I^p$$

$P < 1$  underestimate  
 $P > 1$  overestimate



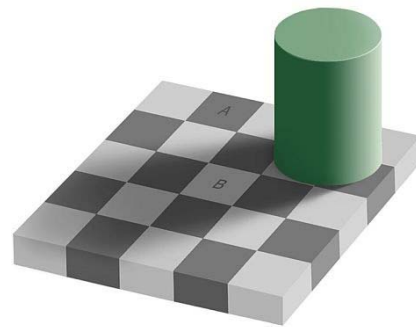
[Information Visualization, Ware 04]

## Apparent Magnitude Scaling

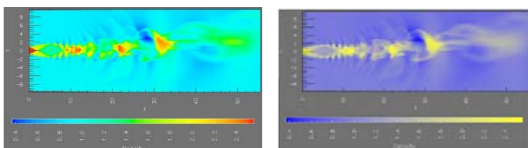


[Cartography: Thematic Map design, Dent 96]

## 色の知覚



## カラーマッピング



レインボーカラーマップ  
 低周波成分がわかりやすい

色調による1次元表現  
 高周波成分がわかりやすい

<http://www.research.ibm.com/people/l/lloyd/color/color.HTM>

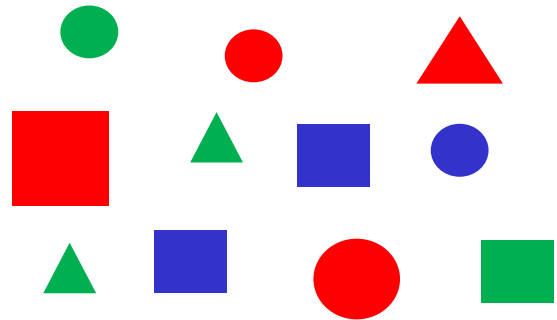
## Avoid Saturated Colors



## 知覚に関する基礎知識

- Preemptive Perception (意識的な探索を必要としない、自動的・即時的な知覚) を活用する。
- 人間の知覚の特性(歪み)をよく理解する。
- 「注意」の重要性を理解する。

## Count Number of Each Color



How Many Red?

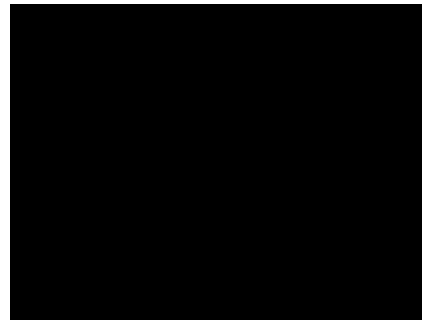
How Many Triangles?

Difference?



<http://sunburst.usd.edu/~schieber/coglab/ChangeBlindness.html>

Difference?



<http://sunburst.usd.edu/~schieber/coglab/ChangeBlindness.html>



## Difference?



<http://sunburst.usd.edu/~schieber/coglab/ChangeBlindness.html>

## Difference?



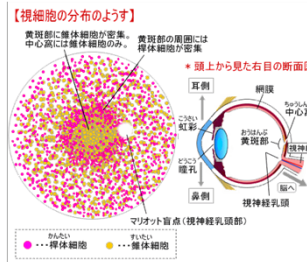
<http://sunburst.usd.edu/~schieber/coglab/ChangeBlindness.html>

## 知覚に関する基礎知識

- Preemptive Perception (意識的な探索を必要としない、自動的・即時的な知覚) を活用する。
- 人間の知覚の特性(歪み)をよく理解する。
- 「注意」の重要性を理解する。

## 中心視野と周辺視野

Curve Ball Illusion ([video](#))



## Outline

- Background
- 「Design of Everyday Things」
- Design Process
  - Research, Analysis, Prototyping
- Evaluation Methods
  - Without Test Users
  - With Test Users

## Design Process 1: Research (ユーザ調査)

デザインするにあたっては  
まずユーザについて知る必要がある。

- × 単なるインタビュー・質問  
(自分の失敗に気づいていない  
(原因や解決法を正しく把握できない)  
(要約してしまう。脚色してしまう)
- 行動を観察する  
エスノメソドロジー  
(コンテキスト調査法 (師匠と弟子))

## コンテキスト調査法

師匠（ユーザ）の体験を弟子（デザイナー）に継承する。  
「教えるつもりになる」 結論のみ→体験の詳細

仕事を見せながら説明する  
不明点があれば質問する  
内容を再確認する。

## Design Process 2: Analysis ユーザニーズ分析

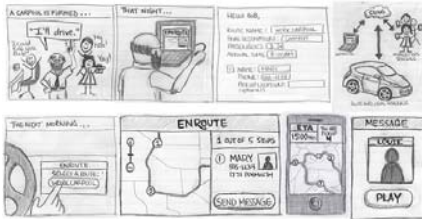
ユーザ調査の結果をもとに、必要な機能などを  
明らかにしていく。分解・分析・発想

ユーザニーズ分析のための道具  
(チーム内のコミュニケーションツール)

シナリオ・ストーリーボード

ペルソナ

## シナリオ・ストーリーボード



システムを使用するプロセスを詳細に物語として  
書き出したもの。

## ペルソナ

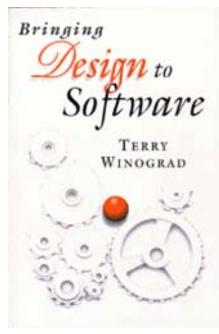
| Donny & Marie  |   | Personas overview  |  |
|----------------|---|--|--|
| Name           | Donny   | Glen   | Susan  |
| Role/Job title | Performer   | Video Engineer   | Stagehand  |
| Age/Gender     | 33, male  | 48, male<br>Wants to be able to edit the system up and down and fourth if again unless there is more content to add.                           | 30, female   |
| Plan Points    | is in the middle of performing and can't be distracted  | Would prefer to handle this off to a more junior person  | Susan changes jobs a lot and might not be around for too long  |
| Goals          | 1) Wants multimedia in his show but wants it to be interactive<br>2) Needs an easy interface that can manage large amounts of content quickly and on the fly. | 1) Adults relies on the system and tag them with possible search terms.<br>2) Would prefer to be able to hand this off to a more junior person | 1) Type in what the audience says from backstage on a computer<br>Results with thumbnails are presented to an iPad that Donny is holding.<br>2) This person can change frequently. |

仮想のユーザ。非常に具体的。(名前、職業、年齢、など)。

## Design Process 2: プロトタイプング

参考文献

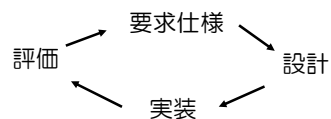
Bringing Design to Software  
Edited by Terry Winograd



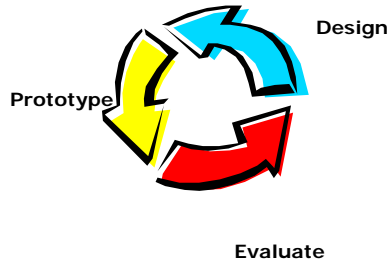
## 通常のソフトウェア開発

要求仕様 → 設計 → 実装 → 検証

## インタフェースデザイン



## Designing User Interfaces



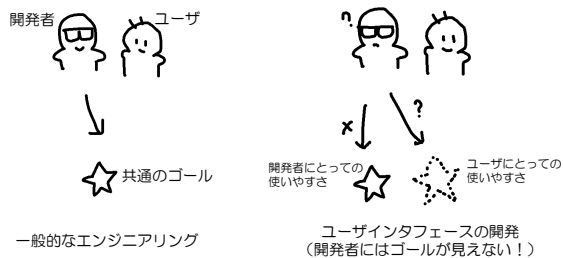
手早く様々な動作を試せる環境が重要

## プロトタイピングの重要性

- いろいろなデザインのパリエーションを試す。
  - 大きな探索空間の中からより良いものを選択できる。
- それぞれのデザインについて簡単にテストできる。
  - 完全な実装するより安くて早い
- ユーザテストを繰り返して、ユーザにフォーカスを当てたデザインができる。
  - 正しいゴールに向かうことができる。

## ユーザテストの重要性

- 開発者はすべて知ってしまう
  - = 自分だけでは、ユーザの気持ちは絶対に理解できない。



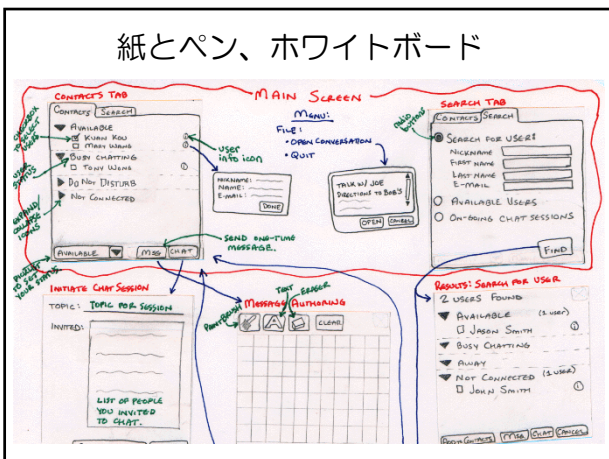
## ラピッドプロトタイピングツール

Low Fidelity

- 紙とペン、ホワイトボード
  - Wizard of Oz
  - PowerPoint
  - Flash
  - Visual Basic
  - Java
- High Fidelity

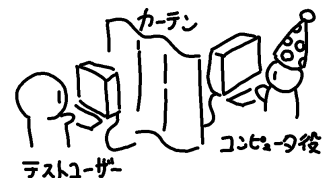


## 紙とペン、ホワイトボード



## Wizard of Oz

人間が裏に隠れてコンピュータの振りをする。  
(複雑なコードを書かずにテストを行う)



音声認識や文字認識、AIを利用した  
インタフェースデザインの検討に用いる

## Wizard of Oz



音声電話を利用した応答システムのデザインの例  
オペレータがマウスクリックで操作する。

## プロトタイピングツール

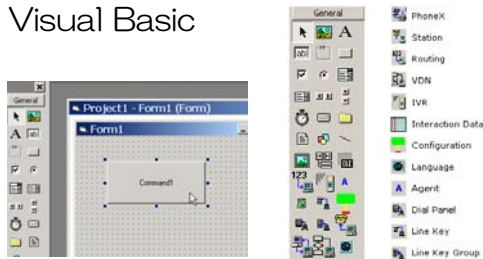
### Adobe Flash



時間軸に従った制御。デザイナーによく使われる。NonGUI  
簡単なスクリプトが書ける。

## インタフェースビルダー

### Visual Basic



GUI部品を並べていく。  
イベントに対する動作を記述する。

## Outline

- Background
- 「Design of Everyday Things」
- Design Process
  - Research, Analysis, Prototyping
- Evaluation Methods
  - Without Test Users
  - With Test Users

## ユーザインタフェースの評価基準

## ユーザインタフェースの評価基準

- 目的を達成できたか。成功率。
- 操作の速さ。操作時間。
- 操作の正確さ。エラー率。
- 操作の覚えやすさ。学習のしやすさ。
- 操作の思い出しやすさ。
- 主観的満足
  - 自分で操作している感覚。
  - 視覚的な美しさ。
  - 動作の軽快さ。反応の良さ。

## Evaluation Methods

- Without Test Users
  - Guidelines ガイドラインにそったチェック
  - Task Analysis 形式的タスク分析
- With Test Users
  - Subjective (インタビュー、アンケート、フォーカスグループ)
  - Log Analysis (プロトコル解析、ログ解析、時間計測)
  - Observation (対話、ハーフミラー、ビデオ)

## インタフェースの評価法



参考文献

Usability Engineering  
Jakob Nielsen

## Evaluation Methods

- Without Test Users
  - Guidelines ガイドラインにそったチェック
  - Task Analysis 形式的タスク分析
- With Test Users
  - Subjective (インタビュー、アンケート、フォーカスグループ)
  - Log Analysis (プロトコル解析、ログ解析、時間計測)
  - Observation (対話、ハーフミラー、ビデオ)

## ガイドラインに沿ったチェック

Heuristic Evaluation by Experts

- 1) Pre-evaluation training
  - give evaluators needed domain knowledge and information on the scenario
- 2) Evaluation
  - individuals evaluate and then aggregate results
- 3) Severity rating
  - determine how severe each problem is (priority)
    - can do this first individually and then as a group
- 4) Debriefing
  - discuss the outcome with design team

## ガイドラインに沿ったチェック

Check List

1. シンプルで自然な対話
2. ユーザの言葉で話す
3. 記憶負荷を最小限にする
4. 一貫性
5. フィードバック
6. 出口を明らかにする
7. ショートカット
8. 適切なエラーメッセージ
9. エラーを防ぐ
10. ヘルプとドキュメンテーション

## ガイドラインに沿ったチェック

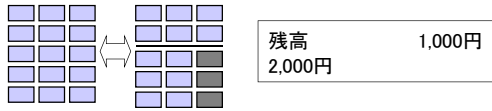
1) シンプルで自然な対話



## ガイドラインに沿ったチェック

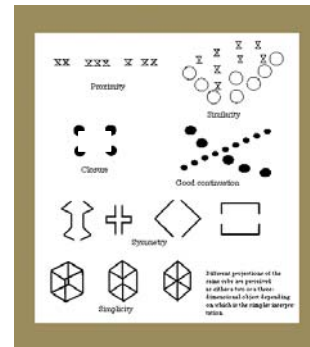
### 1) シンプルで自然な対話

グラフィックデザインの原則  
(ゲシュタルト理論)



少ないほど良い オブジェクト数、色数

## ゲシュタルト理論



## ガイドラインに沿ったチェック

### 3) 記憶負荷を最小限にする

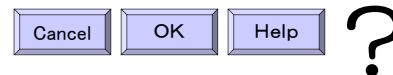
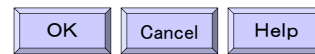
例や単位を画面に表示する



少数のルールで多くの操作ができるように。  
汎用コマンド (コピー、UNDO、etc)

## ガイドラインに沿ったチェック

### 4) 一貫性



## ガイドラインに沿ったチェック

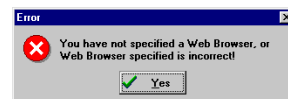
### 4) フィードバック



瞬時と感じる応答     ~0.1秒  
問題ない応答         ~1秒  
対話に集中できる     ~10秒  
要待ち時間表示       10秒~

## ガイドラインに沿ったチェック

### 8) 適切なエラーメッセージ



Computer:     Type user name  
Bissert        Bisseret  
Computer:     Error, type user name

ユーザの理解できる言葉で理由を説明する。  
解決法を提示すること。

## ガイドラインに沿ったチェック

### 9) エラーを防ぐ

モードの使用を避ける。

どのモードなのかを明示する。

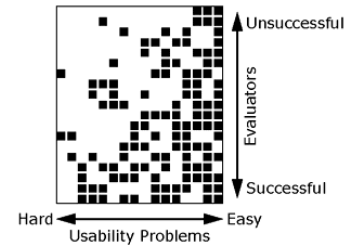
## Heuristic Evaluation by Experts

実際にユーザを使うテストよりも安くて早い。  
複数人で独立にチェックすること。(3~5人)

人によって違う問題を発見する。

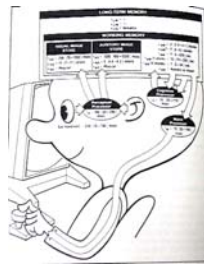
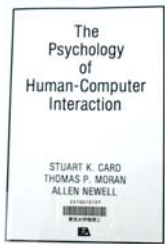
複数人を揃えることで問題を多く発見できる。

ポアソン分布になる



## 形式的タスク分析

Stuart K. Card "The Psychology of Human-Computer Interaction"



認知心理学の知見・手法をHCIに応用した

Stuart K. Card

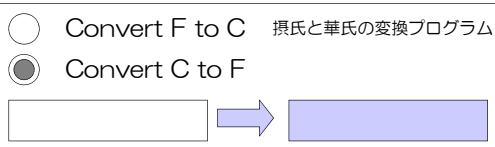
"The Psychology of Human-Computer Interaction"

## KLM model (Key-stroke Level Model)

- K: キー打鍵の時間(平均0.2秒。0.08~1.2秒)
- P: マウスのポインティングの時間(平均1.1秒。0.8~1.5秒)
- H: 手の移動時間(平均0.4秒)
- D: 長さlの線分をn本描画する時間(0.9 n + 0.16 l秒)
- M: 精神的準備時間(平均1.35秒)
- R: システムの応答時間(t)

例: プルダウンメニューから候補の一つを選択:  
MHPKMPK = 5.7秒

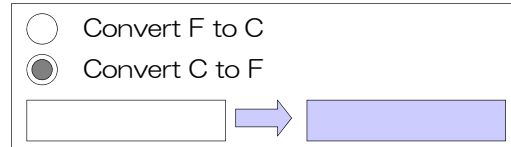
## KLM model による分析の例



- H 手をマウスへ
- P マウスを移動
- K マウスでクリック
- H 手をキーボードへ
- KKKK 4桁の数値入力
- K リターン

→ HPKHKKKKK

## KLM model による分析の例

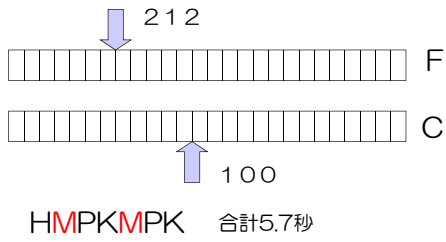


HPKHKKKKK  
→ HMPKHMKKKKMK 合計7.15秒

摂氏と華氏があらかじめ選択されていた場合、  
MKKKKKMK 合計3.7秒

平均5.4秒

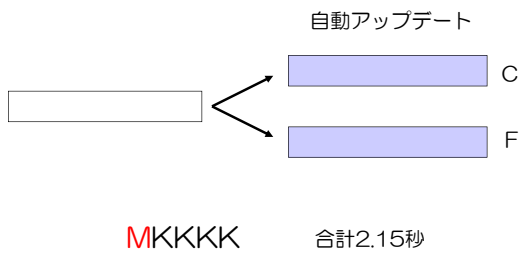
### KLM model による分析の例



### KLM model による分析の例

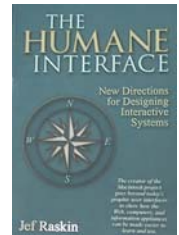


### KLM model による分析の例

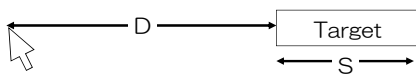


### KLM model による分析の例

正確な時間の予測はできないが  
複数のデザインの間での  
検討に役に立つ。



### Fit's law



$$\text{Time} = a + b \log_2 (D/S + 1)$$

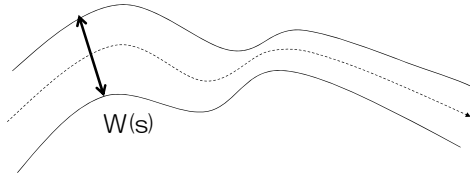
### Hick's law

n個の中から1個選ぶ

$$\text{Time} = a + b \log_2 (n + 1)$$



## Steering law

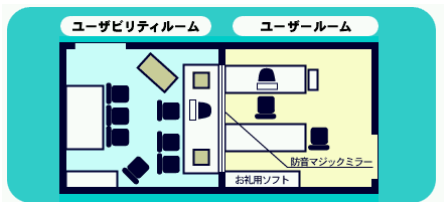


$$T = a + b \int_c \frac{ds}{W(s)}$$

## Evaluation Methods

- Without Test Users
  - Guidelines ガイドラインにそったチェック
  - Task Analysis 形式的タスク分析
- With Test Users
  - Subjective (インタビュー、アンケート、フォーカスグループ)
  - Log Analysis (プロトコル解析、ログ解析、時間計測)
  - Observation (対話、ハーフミラー、ビデオ)

## Usability Laboratory



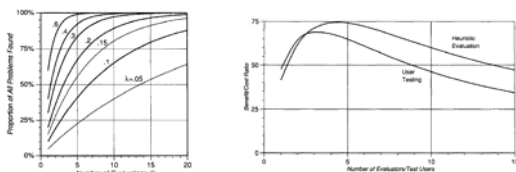
マイクロソフトのページより

## ユーザテストの仕方

1. 準備
2. 目的の説明  
「評価するのは製品であってユーザではない」  
「失敗するのは製品のせいである」
3. 「いつでもやめてよい」と知らせる
4. 部屋の装置について説明する
5. 「声を出しながら考えること」を教える
6. 「操作を助けることはしません」と伝える
7. ソフトウェアと作業内容を説明する
8. 質問がないか聞いてから始める
9. まとめ  
何を知りたかったのか説明する  
質問がないか聞く。感想を聞く。説明を求める。

## A Mathematical Model of the Finding of Usability Problems J. Nielsen and T. K. Landauer '93

問題点を見つけるのに必要なユーザテストの数を解析している。



小さいシステムなら3~5人くらいでOK

<http://www.useit.com/alertbox/20000319.html>

| Project Size | Cost     | Benefits    | Benefit/Cost Ratio |
|--------------|----------|-------------|--------------------|
| Small        | \$10,000 | \$37,900    | 3.8                |
| Medium-large | \$18,000 | \$673,000   | 34                 |
| Very large   | \$46,000 | \$8,200,000 | 178                |

Table 5 Cost-benefit analysis for using the optimal number of test users in user testing

## 注意すべき点

被験者の個人差が大きい。個人差で2倍の速さ。  
被験者の学習。一度使ったらもう使えない。  
要素の混乱。条件を平等に。順番のバランス。  
within group 一人が両方テスト  
between group 一人は片方だけ

倫理的問題。被験者はモルモット？  
ストレス・プライバシー

## Summary

- Background
- 「Design of Everyday Things」
- Design Process
  - Research, Analysis, Prototyping
- Evaluation Methods
  - Without Test Users
  - With Test Users