

# IT技術者のための イノベーション・デザイン

## 一課題の調査分析編その3一

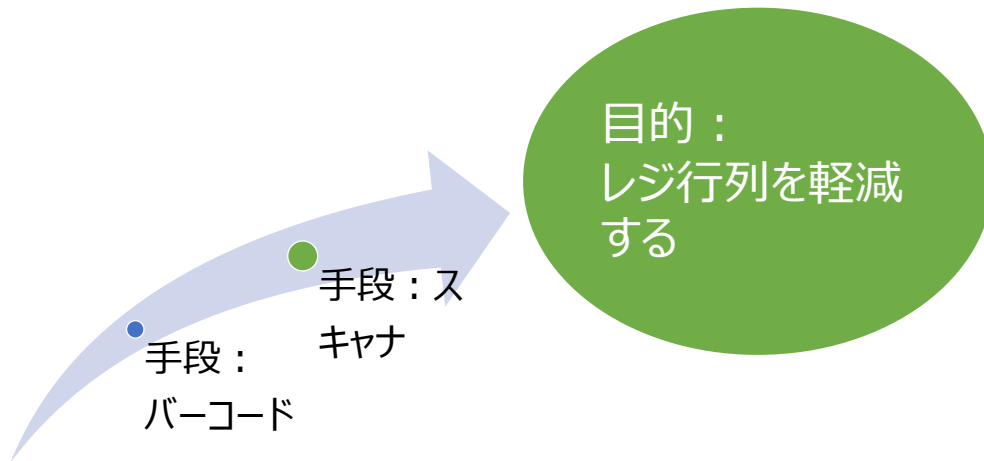
[https://satoyoshiharu.github.io/innovation\\_design/](https://satoyoshiharu.github.io/innovation_design/)

# 振り返り

2

まずこれまで見てきたことを振り返ります。

## 目的(WHAT) ≠ 実現手段(HOW)

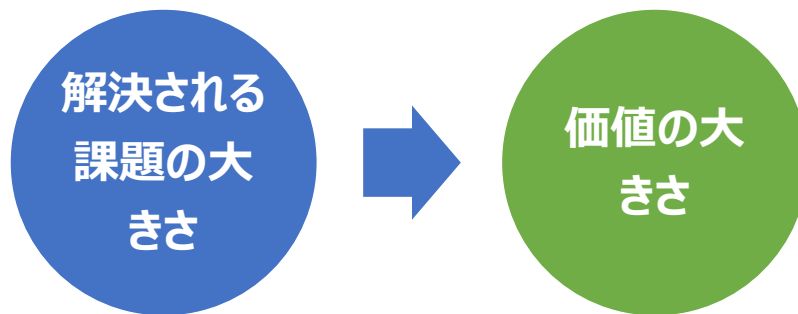


3



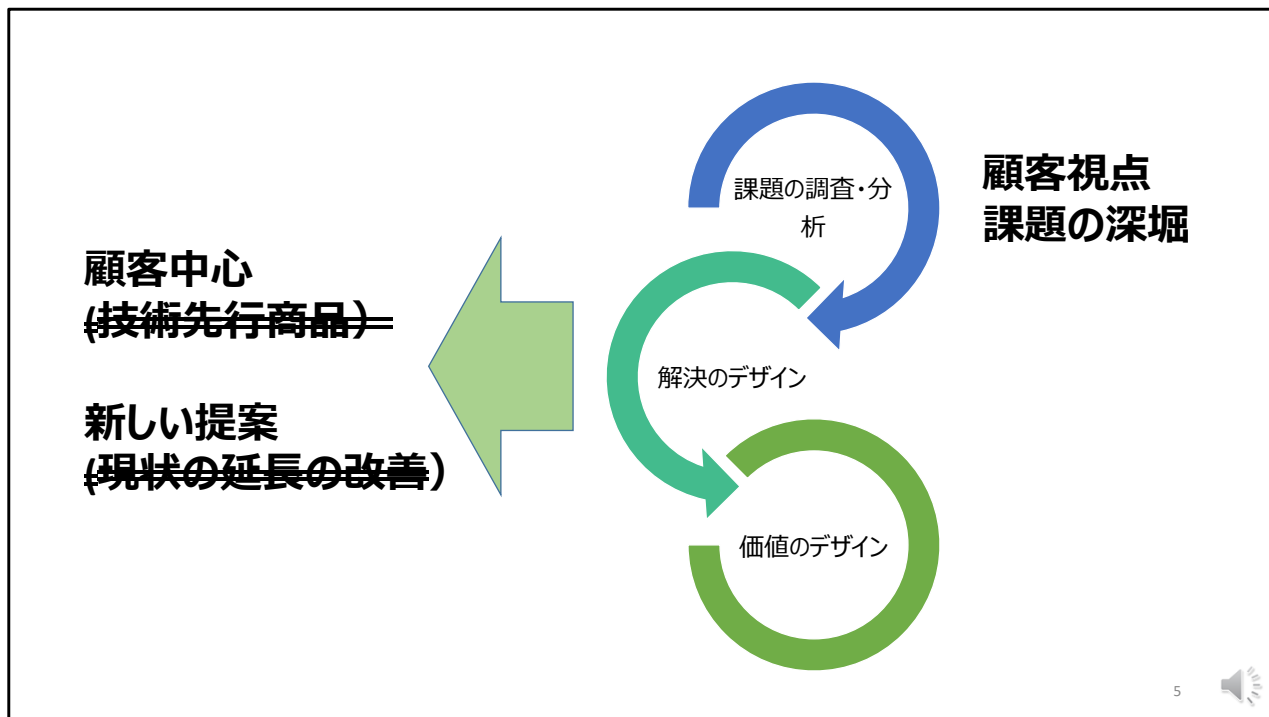
振り返りとして、最初に手段と目的は違うということを思い出してください。バーコードという技術は手段、スキャナーという技術は手段です。そして、レジ行列を軽減することが目的です。我々は、いま、顧客の課題の調査分析をしています。その対象である課題というのは、目的の部分です。誰が何に困ってどうしたかったか？小売り業界という顧客が、レジの非効率に困り、レジ行列を軽減したかった。そのような課題の定義を、自分たちでやろうとしています。

価値の大きさは、顧客のどんな課題を解決するか（WHAT）で決まる。



4

イノベーションは、＝技術革新ではありません。技術というHOWは、手段であって、お客様はそこには関心がありません。  
その価値は、HOWではなく、どういう課題を解決するかというWHATで決まります。



3つのフェーズ、課題の調査分析、解決のデザイン、価値のデザインを、やることを明確に分けて、この順番でやることで、顧客視点を一貫して維持し、新しい提案を生み出すことができます。課題の調査・分析フェーズで、100%顧客目線を持つことでその一貫してそれを維持し、技術先行商品の罠を避けます。また、最初に課題を深堀することで、現状の延長の改善に陥らずに新しい提案となるようにします。

## 顧客の言うこと != 顧客の本当の目的・課題

ヘンリー・フォード曰く、「顧客に欲しいものを聞いたら、『もっと速い馬が欲しい』と答えるだろう。」

スチーブ・ジョブズ曰く、「人は欲しいものがわからない。これだろう？と言われて初めてそれが欲しいとわかる。」「人が本当に欲しいものを見つけるのが、あなたの仕事だ」

6

新しい解決策を提案するとき、人々から話を聞いてはだめです。人々が言語化できるのは、すでにある解決策の延長にあることだけだからです。人々が言語化できないことを発見しなければなりません。人々が本当に欲しいものを見つけるのは、リーダーの仕事です。

## 定性的調査

- 発見
- 問いを明確にする



## 定量的調査

- 証明
- どのくらい？

7



調査は、定性から定量へ進めます。まずは、課題を発見し明確にしたうえで、どのくらいという問います。問いが明確になっておらず、あいまいな段階で、どのくらいかを調べても、導かれる結論はボケたものになります。

## 定性調査方法その1：行動観察法

課題が発生する現場で、実際の行動を、観察することで、無意識的な行動の背景にある（言語化されていない）目的・課題を探りだす。



8

定性調査方法の一つに行動観察法があります。この手法は、実際の行動を、課題が発生する現場で、観察することで、無意識的な行動の背景にある（言語化されていない）目的・課題を探りだすものです。



## 定性調査方法 2 : 半構造化インタビュー

事前に最低聞きたいことを準備しておくが、対話の中で、流れに即して進行させる。表面に出た言葉をきっかけに深掘し、本音や深層心理を探り出す。



9

半構造化インタビューは、通常のインタビューと比べて、最低限聞きたい質問を2, 3準備するだけで、ほかは流れに即して対話を進めます。流れの中で、未知のことを掘り出していきます。

## 今日のゴール

- 各人の行動観察のメモをもとに、課題仮説のアイデアをチームで、KJ法によってマインドマップにします。

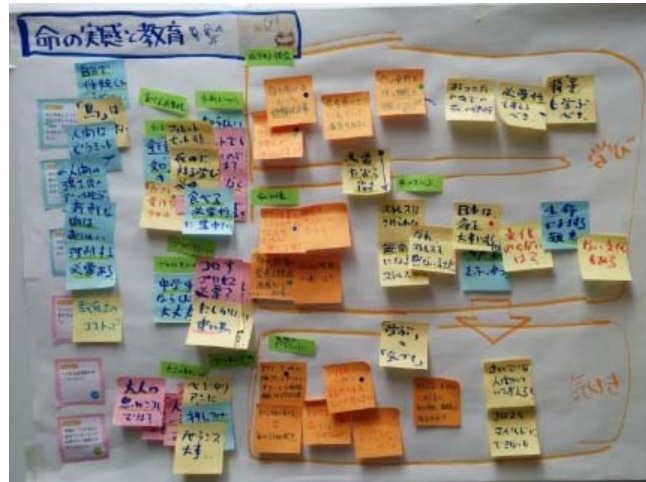
## チーム作業



11

なぜチームでアイデアをまとめるか？ チームのアウトプットは、個々のアウトプット以上のものを生むからです。

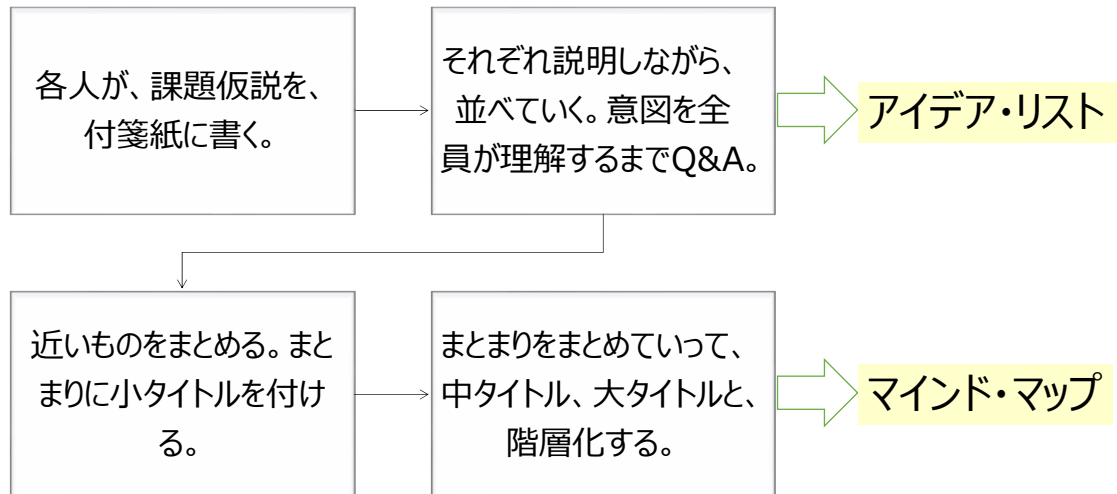
## KJ法ブレインストーミング（川喜田二郎）



12

KJ法は、たいていの方がすでに経験されていると思います。Affinity Diagram と呼ばれています。文化人類学者の川喜田二郎が1967年に作ったもので、個人の考えをまとめる際や、共同でブレインストーミングをすることに、利用されています。

## KJ法ブレインストーミングの手順



13

この授業では、この図にある手順で、KJ法を利用します。

まず、各人が、課題仮説を、付箋紙に書きます。チームで、それぞれ付箋紙に書いたことを説明しながら、並べていきます。意図を全員が理解するまでQ&Aします。全員のアイデアを並べたところで、アイデアリストができます。

次に、内容が意味的に近いものをまとめます。まとまりが落ち着いたら、まとまりに小タイトルを付けます。さらに、中タイトル、大タイトルと、階層化していきます。この段階で、できたマップを、マインドマップと呼ぶことにします。

## K J 法の注意点

- 他人のアイデアを批判しない。
- 他の人のアイデアに触発されて、追加OK。（右脳の連想による発想）
- 階層化の過程でも、追加OK。（視覚的空間認知を介した発想）

14

KJ法は、アイデアを自由に出し合うことがゴールなので、他人のアイデアを批判してはいけません。

他の人のアイデアに触発されて違うアイデアが出たら、随時、説明しながら、追加します（右脳の連想による発想）。

階層化の過程でも、新しいアイデアが出れば、随時、追加します。全体的に眺めると、見逃しや、新しい視点などに気づきます（視覚による空間認知の利用）。階層化を進める過程で、例えば、こういう括りならば、これこれもあるんじゃないか？ また、これがあるなら、あれもあるんじゃないか？ という具合に発想が広がります。

## 調査分析 理解度確認クイズ 4

- 以下から、正しいものと誤りとを選びなさい。
  - KJ法は、ブレイン・ストーミングなどに使う。
  - KJ法では、最初アイデアを出し尽くし、その後、類似のアイデアに分類しまとめていき、階層化する。
  - KJ法では、他人のアイデアに対し、批評を加えてもよい。
  - KJ法では、アイデア出し、階層化のいずれのステップでも、新規アイデアを追加してよい。

# 調査の演習 3

## 課題仮説のマインドマップ

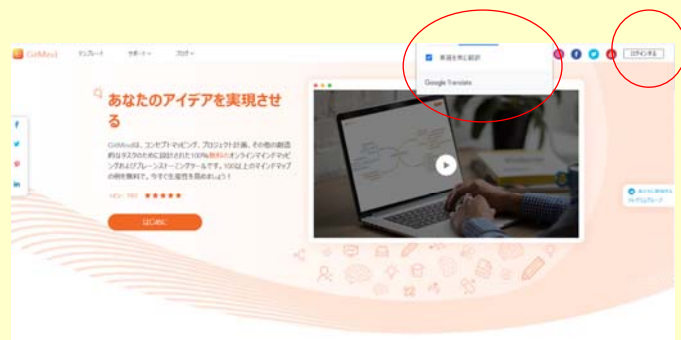


## マインドマップのツール比較

	利点	欠点
FreeMind	<ul style="list-style-type: none"> <li>• オープンソフトの老舗・標準</li> <li>• 多機能</li> <li>• 表示がコンパクトなのでたくさんの情報を整理できる</li> </ul>	<ul style="list-style-type: none"> <li>• クライアントソフトで、クラウドで使えない</li> <li>• プレゼンの構成とか、一人でじっくり検討したいときに便利</li> </ul>
MindMup	<ul style="list-style-type: none"> <li>• クラウド</li> <li>• スマホで編集できる</li> <li>• レイアウト自動</li> <li>• Export/import使える</li> </ul>	<ul style="list-style-type: none"> <li>• 英語UI</li> <li>• テンプレートPoor</li> <li>• 共同編集すこし面倒</li> <li>• 普段の一人使いに便利</li> </ul>
GitMind	<ul style="list-style-type: none"> <li>• クラウド</li> <li>• 日本語UIあり</li> <li>• きれいなデザインテンプレートがたくさん</li> <li>• 共同編集の手順が明確でわかりやすい</li> </ul>	<ul style="list-style-type: none"> <li>• 中華製</li> <li>• スマホではブラウザだけで編集できない</li> <li>• レイアウトは手動、すぐ汚くなる</li> <li>• XMind以外の他フォーマット Import/Exportない</li> <li>• チーム作業が楽</li> </ul>

# GitMindに慣れる

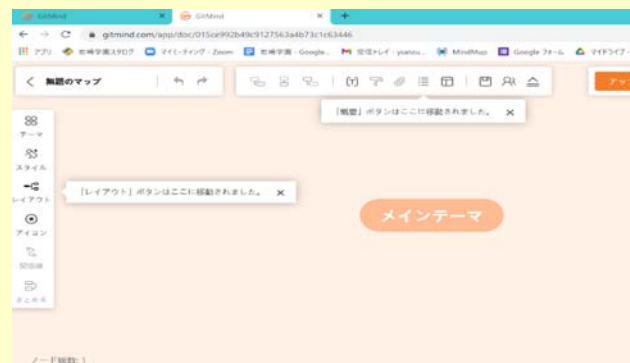
- Chromeで、<https://gitmind.com>にアクセスします。日本語を選択。
- すでにChromeでloginしていなければ、Googleアカウントで入ります。



18

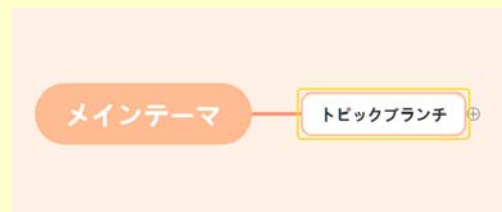
## GitMindに慣れる

- 「今すぐ無料作成」>「新しいブランクマップ」でMind Mapを作ります。
- 左上の「無題のマップ」を適当な名前に変更。



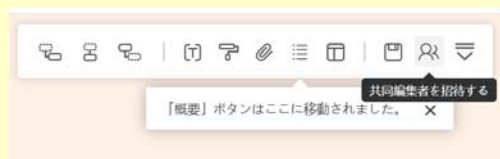
## GitMindに慣れる

- ルートノードをクリックして、テキストを編集してみる。
- 子ノードをぶら下げてみる。
- 孫ノードを2つぶら下げてみる。
- 丸マイナス記号をクリックして畳み込みしてみる。
- レイアウトで配置を変えてみる。



## GitMindに慣れる

- 上のツールバーのCooperation（共同編集者を招待する）をクリック



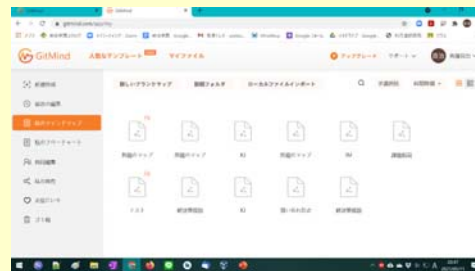
## GitMindに慣れる

- invite collaborators で、チームの他の人に、link&コードを教えるかメールを送る。
- 相互に編集者の役割を交換して、共同編集を試す。（一度に一人しか編集できない）
- 右上のShareで結果のURLを取得できることを確認する。チームの成果提出時は、このlinkを張ってもらいます。



## GitMindに慣れる

- 左上のタイトルの”<“で、自分のgitmapのfolder viewに戻れることを確認。一度作成したら、クラウドのファイルがここにあるので、再編集するには、ログイン後、ここに戻ればよい。



## K J 法で課題仮説展開

### 1. 準備

- ① 調査の演習 1（その 1 スライド）を振り返ってください。行動観察をしました。各自、メモしたアイデアを、チームに説明するための時間をとってください。
- ② リーダーは、課題仮説というタイトルで、GitMindのファイルの一つ作成して、チームメンバーを共同編集者として招待してください。

### 2. ブレークアウトルームに別れ、以下の作業をやってください。

- ① チームメンバーが、一人ずつ順番に、自分の課題アイデアから一つずつアイデアを説明し、ルートに追加していきます。それを終えたら、全員の課題仮説のアイデアリストとなります。
    - ① **ここでは、たくさん課題アイデアを出すことがゴールです。**
    - ② **行動観察で見つけたものを含め、自分が普段感じる不便なことでも結構です。なんの制約もなしに、課題と思いつくものを挙げてください。**
    - ③ **3回目の授業フォルダーに、課題アイデアリストのサンプルを置いてあるので、それ見て共感するものをピックアップしてもOK。**

ただし、これは思いつくためにシーン別にグルーピングしていますが、課題の中身による階層化にはなっていないので、次の階層化の作業時には、まとめ方は参考にしないでください。

  - ④ **とにかくいろんな課題アイデアを繰り広げてください。**
- ② チームで議論しながら、似たものをグループ化してタイトルをつけることを繰り返し、階層化し、マインド・マップとします。

### 3. スタログのチーム名フォルダーに、GitMindファイルのLinkを提出してください。

### 4. チーム発表会やります。



## まとめレポート

- 以下を文章にして提出してください。
  - チーム内で、自分のアイデアや意見を出せましたか？ 意見を出すことにためらったとしたら、そのときにどう思いましたか？