

# エンジニアの 知的生産術

豊田工業大学 教養基礎セミナー

2019-05-27

サイボウズラボ 主幹研究員

西尾 泰和

# 何を話すか

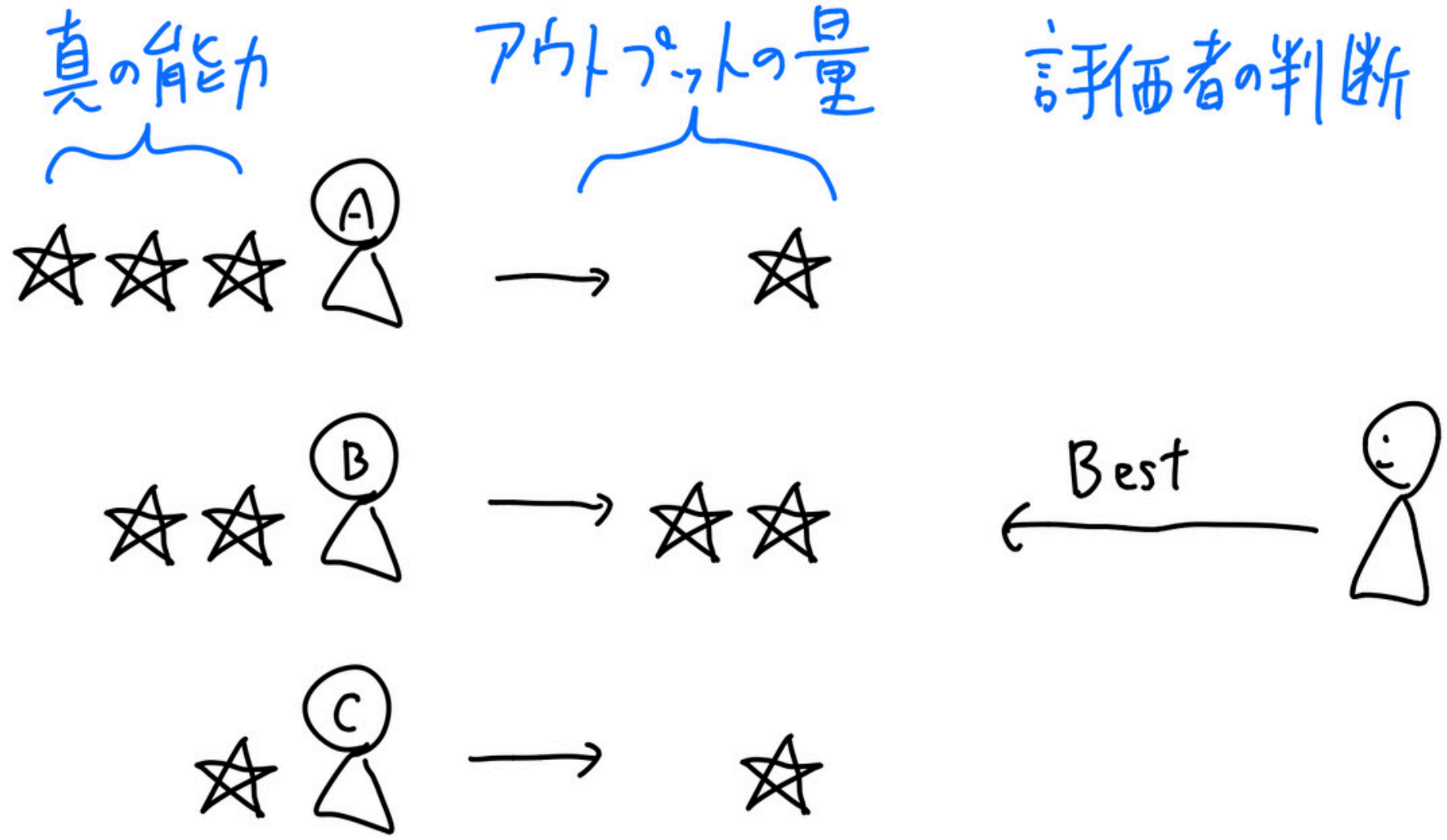
- みなさんはアウトプットの方法を学んでいると聞きました
- 私もアウトプットが大事だと考え、  
「エンジニアの知的生産術」にその手法をまとめました。
- この授業では、
  - まず、なぜアウトプット手法を学ぶとその後の人生で有利なのかを解説します。(why)
  - 次に「エンジニアの知的生産術」をベースにアウトプットの手法を概観します。(how)
  - 最後にみなさんの今後の人生に訪れるゲームのルールを紹介し、アウトプットスキルをどう使うか解説します。

世界は  
本質的に  
不平等

# 世界は本質的に不平等

- 人間の能力に限界があるため
- 人は他人を評価するときに、  
その人の内面を見ることができない
- その人のアウトプットしたものしか  
見ることができない

# 人の評価で起こること



# 不平等の メリット

# 未踏ジュニア

- 僕がPMとして関わっている、17歳以下の学生向けの育成&開発支援プロジェクト
- 今年127件の応募があり、12件ほど選んで、各プロジェクトに最大50万の資金援助をする
- 競争的資金は必ず不平等である。  
応募者全員にお金を配ったりしない。

# 平等に配るとどうなるか

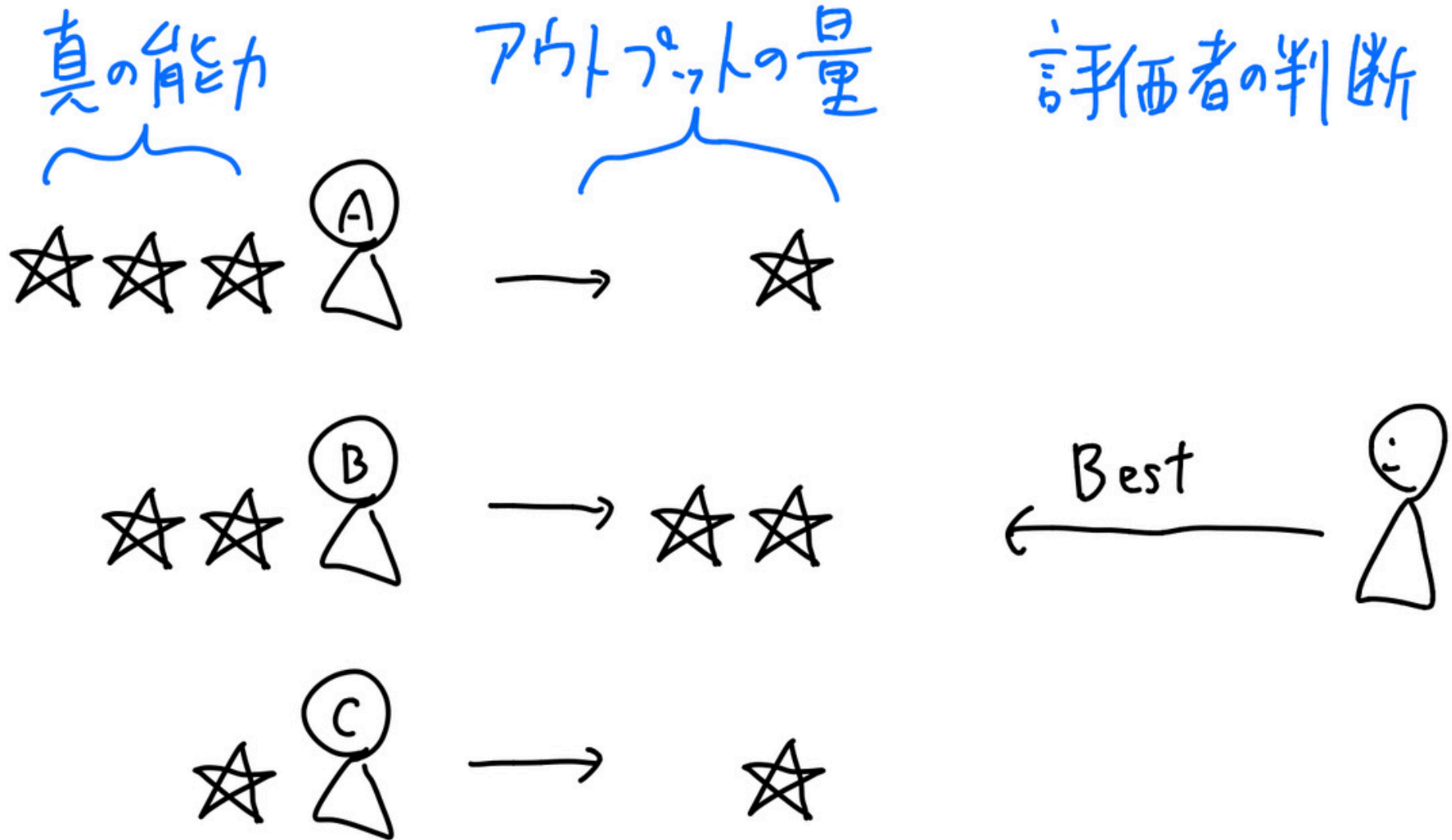
- 今年の応募者全員に均等にお金を配ると1人あたり5万円になる。
- 応募書類を出すだけで5万円もらえるなら有資格者はもっと応募するよね。
- 高校生は全国で300万人いる。  
全員に平等に配ると1人あたり2円。  
→何もできない



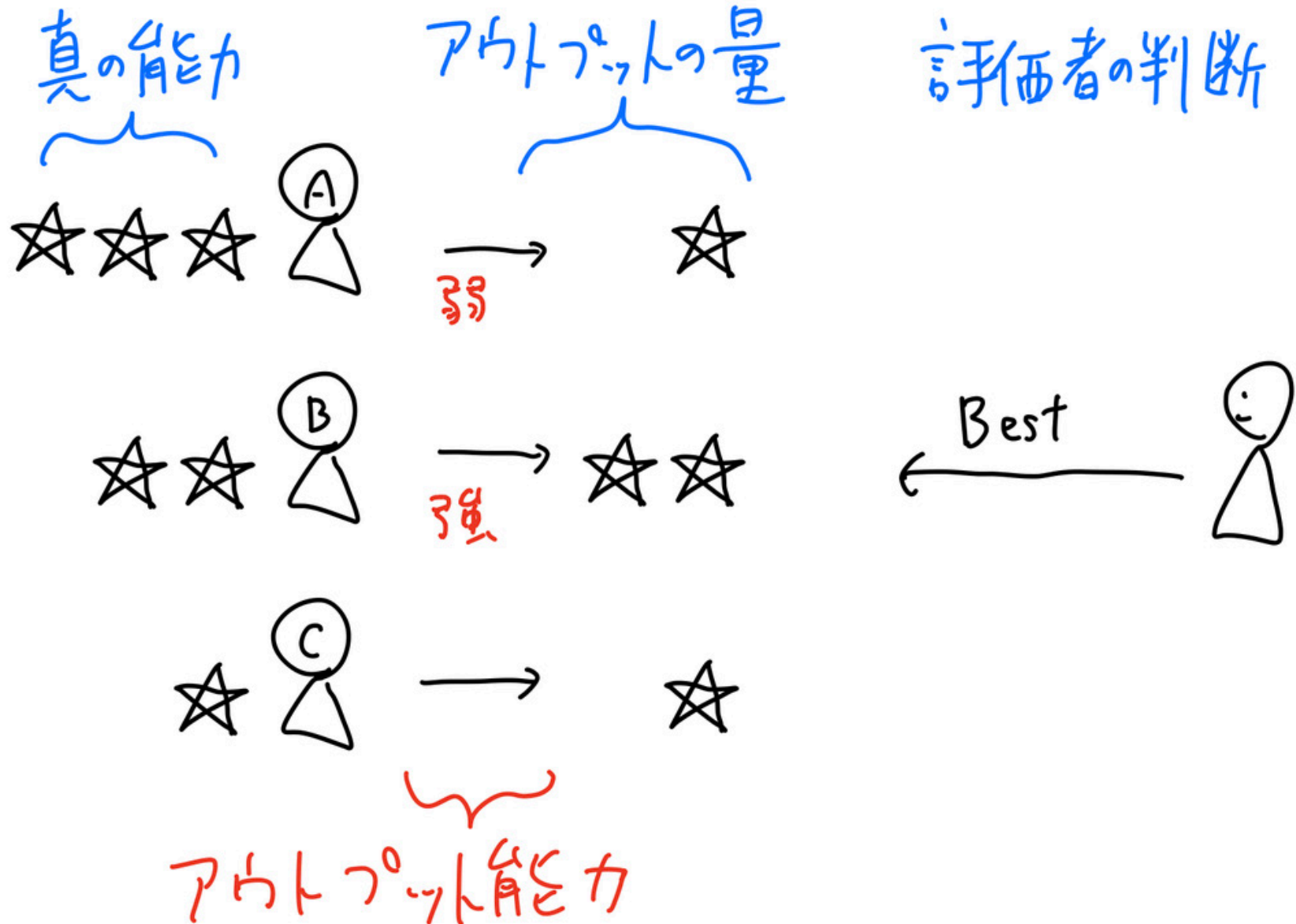
# 不平等にはメリットがある

- 資源を偏らせることにはメリットがある
- 平等に配ったのでは起こらない「良いこと」が資源を偏らせることで起きる
- そのため、世の中では「全員に**平等**に配る」ではなく「**選抜**した一部の人に配る」行為が広く行われている

# 人の評価で起こること



# アウトプット能力



# アウトプット能力

- 自分のやったこと、得たいもの、考えなどを  
受け手にわかりやすい形で表現する能力
- これが強いと競争で有利
- だから鍛える

# 自己紹介

=僕の過去のアウトプットを紹介

# コーディングを支える技術

WEB+DB PRESS plus

## コーディング を支える技術

成り立ちから学ぶプログラミング作法

Nishio Hirokazu  
西尾泰和  
[著]

「なぜ言語設計者は  
この文法を作ったのか」

「言語に共通の知識を身  
につける」

### なぜ言語設計者は この文法を作ったのか

演算子／繰り返し／関数／例外／スコープ  
動的型付け／型推論／配列／辞書  
スレッド／ロック／トランザショナルメモリ  
クラス／インターフェース／継承／委譲  
C3線形化／Mix-in／トレイト

言語に共通の知識を身につける

技術評論社



# コーディングを支える技術

WEB+DB PRESS plus

## コーディング を支える技術

成り立ちから学ぶプログラミング作法

Nishio Hirokazu  
西尾泰和  
[著]

2013年発売(32歳)

陳腐化しにくい知識に  
フォーカス

6年経った今でも  
売れ続けている

## なぜ言語設計者は この文法を作ったのか

演算子／繰り返し／関数／例外／スコープ  
動的型付け／型推論／配列／辞書  
スレッド／ロック／トランザクショナルメモリ  
クラス／インターフェース／継承／委譲  
C3線形化／Mix-in／トレイト

言語に共通の知識を身に付ける

技術評論社



この本は

どうやって

書かれたのか



# エンジニアの知的生産術

WEB+DB PRESS plus

## エンジニアの 知的生産術

効率的に学び、整理し、アウトプットする

Nishio Hirokazu  
西尾泰和  
[著]

自分の環境に合わせて適用し  
創造性を高める

知りたいところから学ぶ  
タスクを小さく区切る  
インプットとアウトプットの繰り返しが記憶を鍛える  
読書の技術  
思い付く限り書き出して全体像を把握する  
アイデアを言語化するアナロジー  
自分自身の経営戦略

10年後も役立つ考え方を身に付ける

技術評論社

本がどう書かれたのか、  
アウトプットのプロセス  
を解説した

発売10日前からAmazon  
予約でベストセラー入り

# 総合ランキング10位



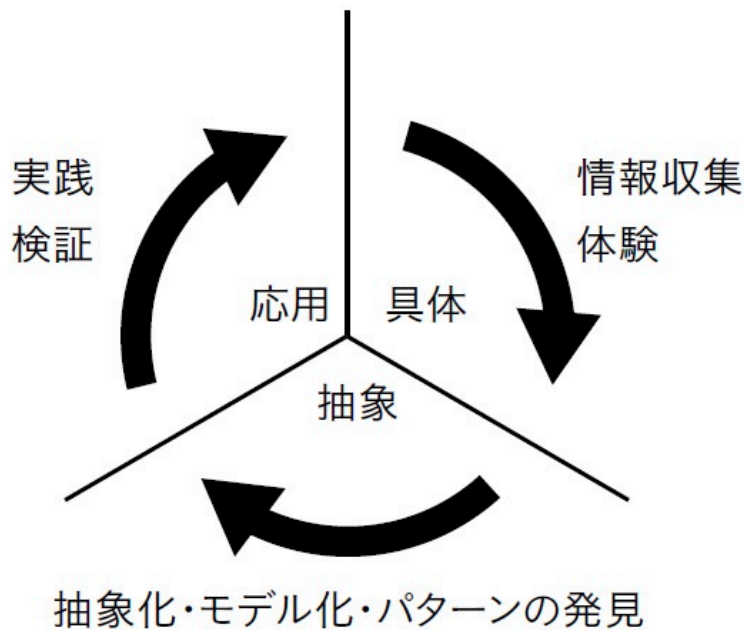
ジュンク堂池袋本店の  
総合ランキングで10位に

# エンジニアの知的生産術

- 今回この書籍をベースに講義をするが、90分で全部伝えるのは当然無理
- まずは「大雑把に全体像を把握」を目指す
- 参考図書としてPDFを共有したので夏休みにでも読むと今後の人生に良い効果

# 各章紹介(0, 1, 2)

0章(はじめに)、1章、2章は  
サイクルを回す学び方の説明



まず0章で大雑把に。  
次に1章で詳しく。

2章はサイクルを回す  
原動力"やる気"について。

学びは情報収集・モデル化・検証の繰り返し

# 各章紹介(3, 4)

3章は記憶について。

学んでも蓄積しなかったのでは意味がないから。

4章は読書の方法について。

情報のインプット効率を高めると有用だから。

# よくある質問

Q: なぜ「知的生産術」の本で、学びや  
インプットの方法について書いているのか？

A: 新しいものを思いついたとき、  
すべての要素が新しいのではない。  
大部分の要素は既存のもの。  
組み合わせ方が新しい場合が多い。  
だから「材料」はたくさん入れたほうがいい。

# 各章紹介(5, 6)

5章・6章がアウトプットに関する章

- 5章はインプットしたたくさんの材料をどう組み合わせしていくか
- 6章は心の中に生まれた新しい考えをどうやって言葉にしていくか

# 各章紹介(7)

「何を学ぶのが正しいか？」とよく質問される

人と同じものを学んでも差別化にならないので  
その「質問の仕方」や「答えを求める考え方」  
が変わる必要がある

7章は学ぶこと・アウトプットすることを  
どうやって自分の人生の利益につなげるかの  
**「自分経営戦略」**の話



# アウトプット の具体的方法

# 5章の超圧縮解説

- **思ったことを全部書き出す:**  
消えないようにするために
- **ボトムアップで組み合わせる:**  
新しい構造を見つけるために
- **グループに表札をつけて圧縮:**  
脳の負担を減らすために

# 思ったことを全部書き出す

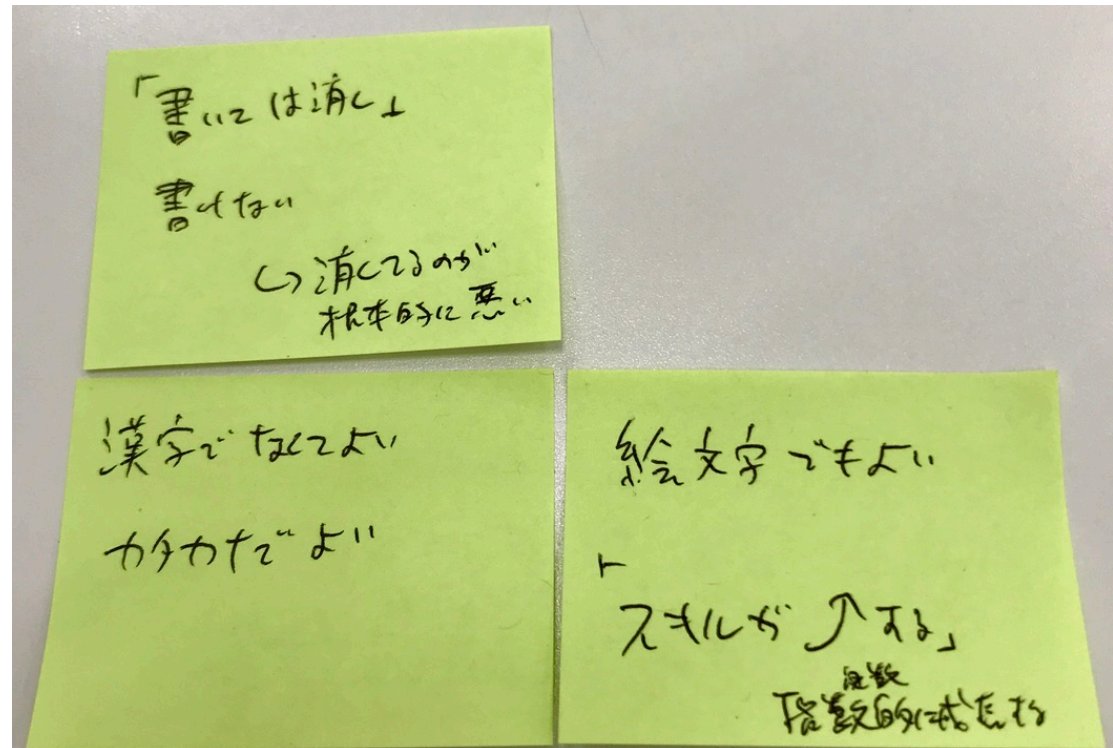
- 一番重要な原則
- 頭だけで考えていると、どんどん消えてしまう
- まず書いて、消えなくする

# 「書いては消ししている」

- 「書いては消し書いては消ししてなかなか進まない」 ←よくあるバッドパターン
- 文章案1を書いて、消して、文章案2を書いたとき、文章案1はすでに頭から消えている
  - 比較したり、よいとこどりをしたりできない
- 「消す」のがよくない、書いてから考えるべき

# 雑に書く

- 最初から完成度100%でなくてよい
- 漢字でなくて良い、カタカナで良い
- うまく言葉にできなければ絵文字でも良い



# タスクを小さくする

- 長文を書くタスクは「大きく」感じてやる気を失いがち
- 「文章に使うかもしれない断片を書いてみる」
- これは小さいタスク
  - 書けば1歩ゴールに近づく
  - いきなり長文を書くより心のハードルが低い

# 付箋を使う？

- 書き出しやすい方法で書き出せばいい
  - たとえばスマホのフリック入力でチャットに
- この講義の続きは「付箋に書いた」という想定
- 現時点で僕自身はそうやっているから

Excelや箇条書きエディタ(アウトライナー)を使って似たことをやる人もいる

書いたものを  
広げて一覧する



# 書いたものを広げる

- 書いたものは「脳の外」に出ている
- 組み合わせを考えるためにはもう一度「脳の中」に戻す必要がある
- 一覧できるように並べれば視線を動かすだけで脳の中に入れなおすことができる

# 眺めて集める

- 付箋を眺めて、関係ありそうなものを近くに移動する
- 「話がつながりそう」みたいに
- やっている最中に思いついたことは、どんどん付箋に書く。書かなければ消えてしまうから。

# トップダウンかボトムアップか

- あらかじめ構造が決まっている文章を作る際に「断片を構造に合わせて分類」するアプローチ
- 断片がすんなり分類できるならよいが、しばしばそうではない
- 「伝えたいこと」(ストーリー, 構造)がまだ自分の中で明確になってない場合
  - →まずそれを発見する必要がある
  - →ボトムアップでくっつけていく

# グループに表札をつけて圧縮

- 「エンジニアの知的生産術」では、  
付箋が100枚ある状態を前提として、  
「圧縮することで脳の負担を減らす」話をした
- 枚数が少ないうちは無理にやらなくても良い
- このスキルを身につけると、  
普通の人が情報洪水で溺れるような状況でも、  
溺れずに文章を作っていくことができる

超圧縮で  
解説した

大事なこと

スキルは  
実践なしに  
身につかない

# 例えばキャッチボール

- 解説書を読むだけでできるようになるか？
- 解説動画を見たらできるようになるか？
- 先生が教室でスライドを使って説明するのを眺めているとできるようになるか？



なるわけない

# キャッチボール

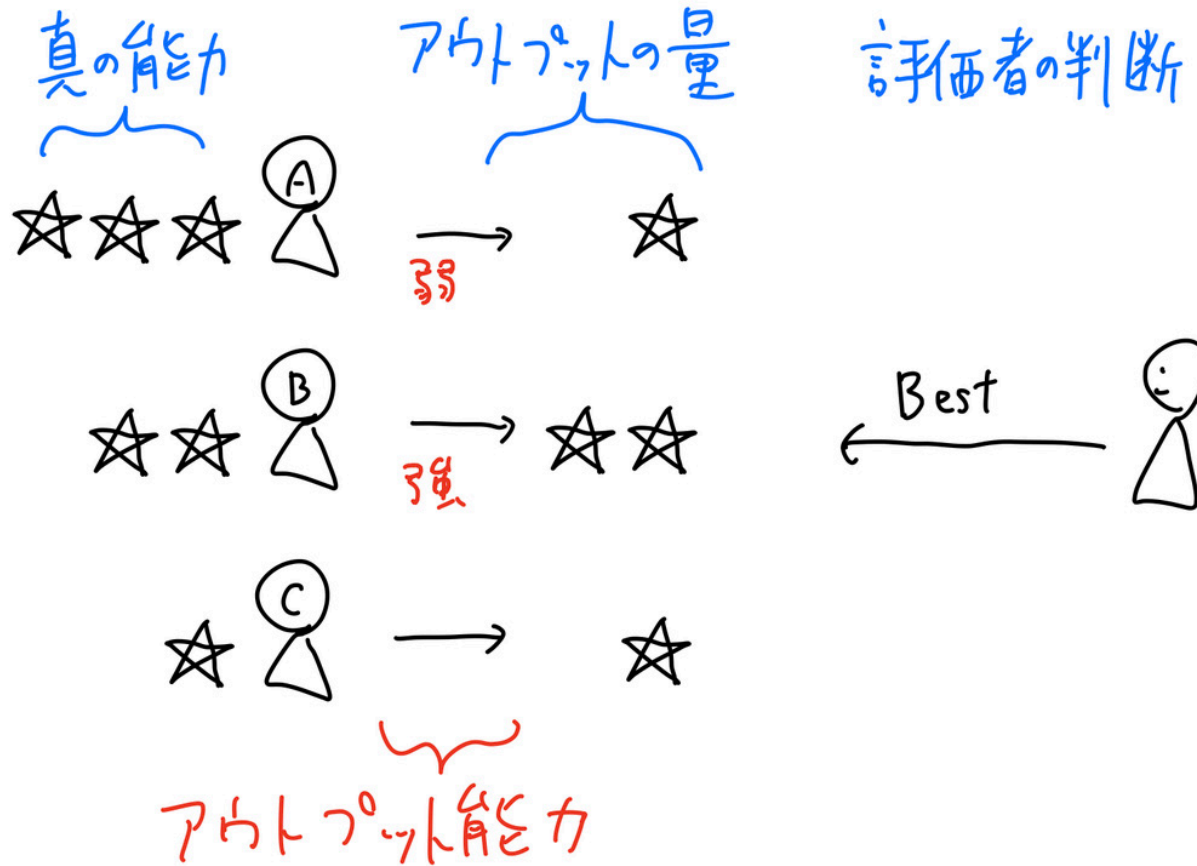
- 実際にボールを投げてみる
  - → 狙ったところに飛ばない
- 飛んできたボールを受けてみる
  - → 受け損なって落としてしまう
- この「実践」と「失敗」を避けていてはキャッチボールができるようにはならない

知的生産の  
スキルも同様

# 知的生産

- 授業を聞くだけでできるようにはならない
- 実際に文章を書いて、人に伝えてみる
  - →思ったように伝わらない
- これをやらなければ、できるようにならない

# 数年後にはこうなる

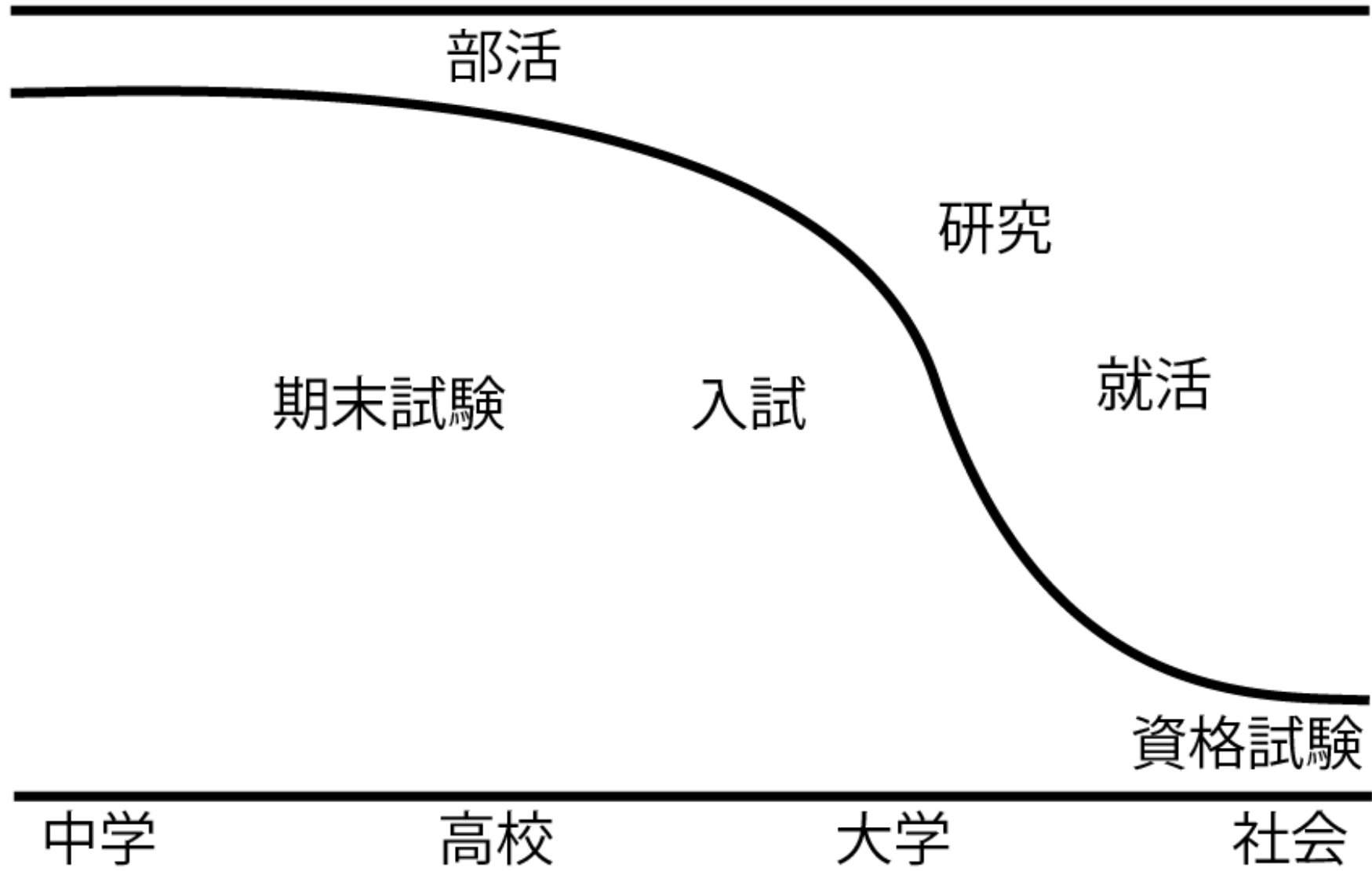


アウトプット能力の強弱が  
あなたの得る待遇に影響する

# ゲームの ルールの 変化

みなさんは  
今、過渡期

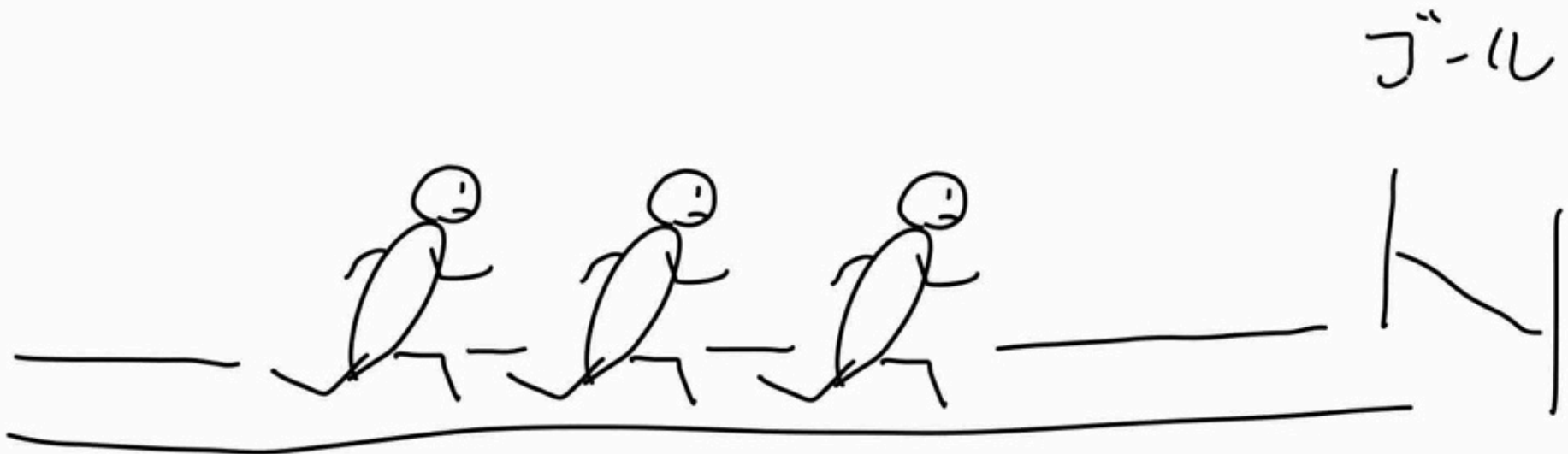
# 大学あたりで切り替わるものがある





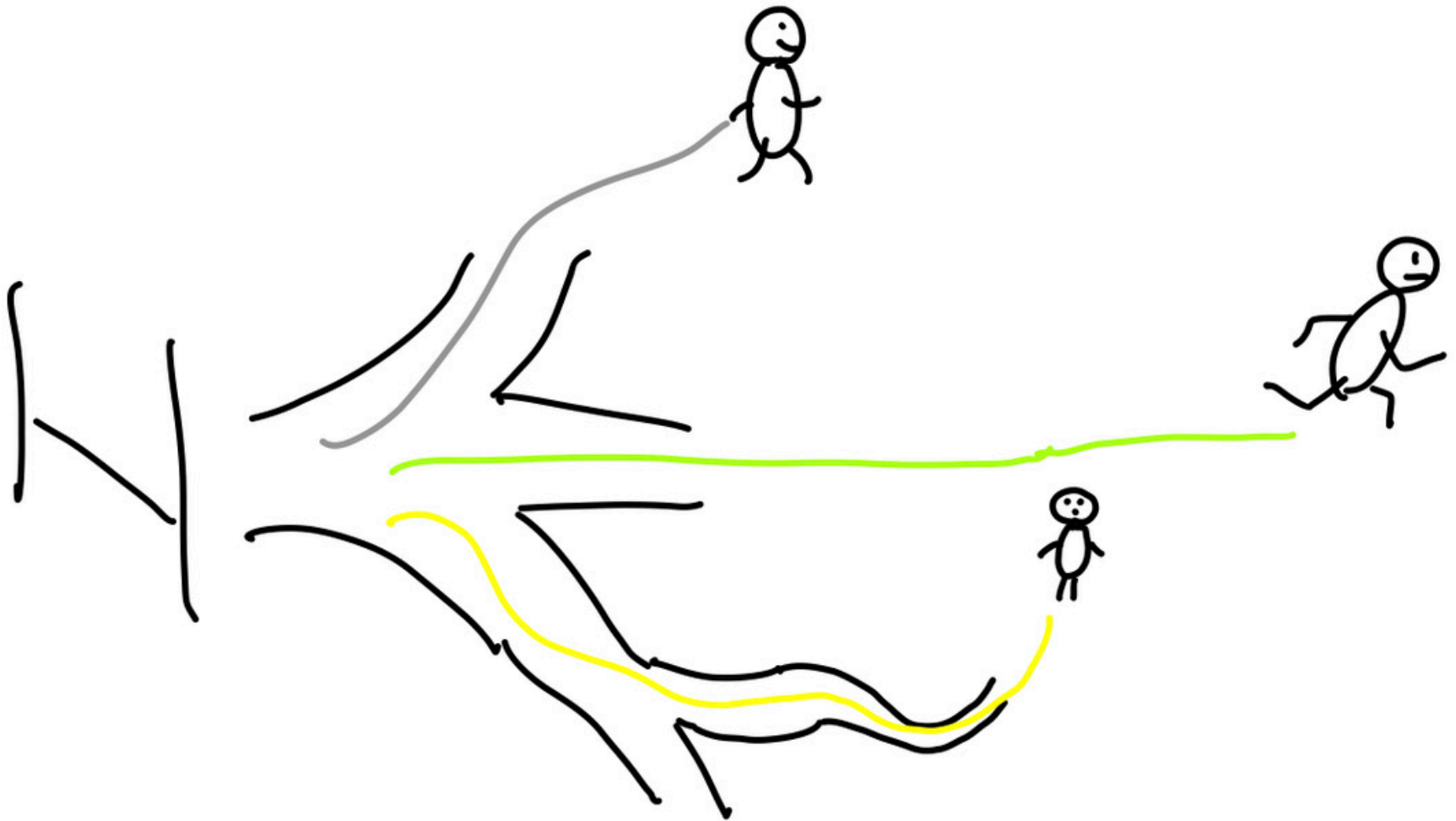
# 一本道のレース

- 走るコースとゴールが決まっている
- 「良い走り」をした人が勝つ



# 探索のゲーム

明確なコースやゴールがない

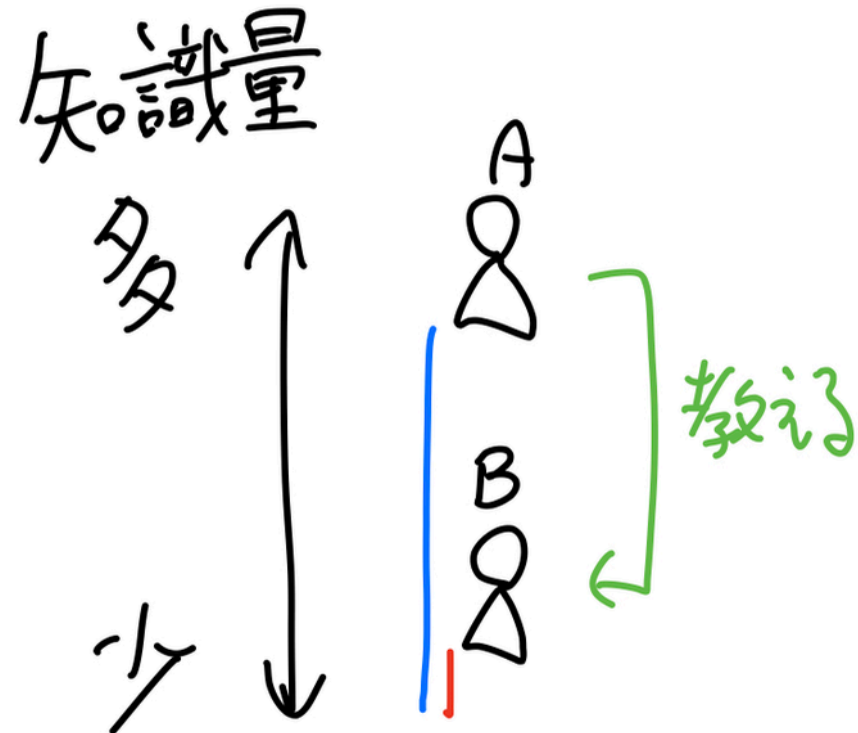
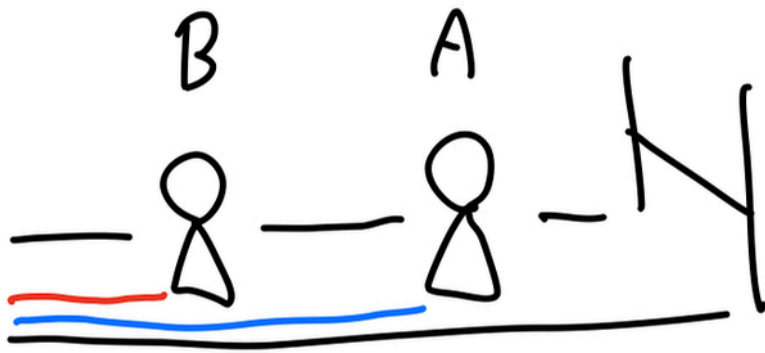


# 探索のゲーム

- まっすぐ走るのが正解とは言えない
- 宝探し：最初に見つけた人が宝を取る
  - 他人の後をついていっても得るものがない
- 他人と違う道を行くほうが有利
- まれに大きな金脈が発見される
  - 人が殺到して奪い合う→一時的にレース

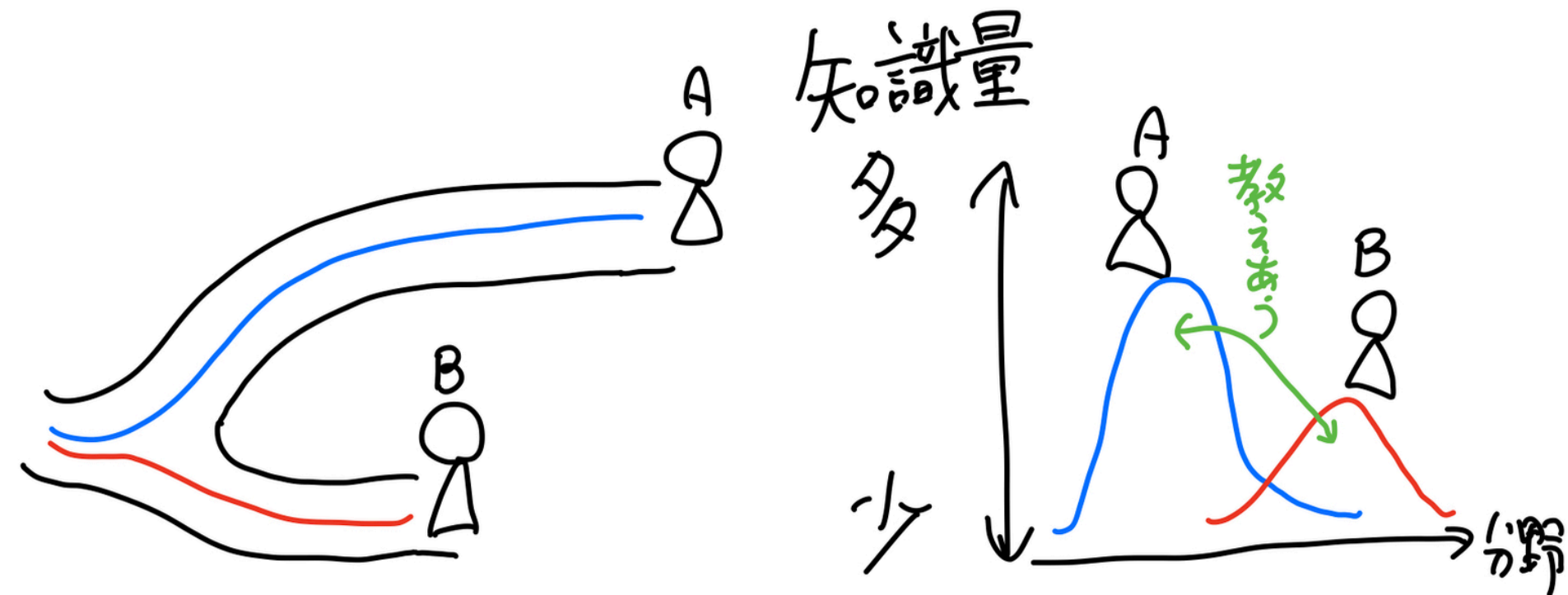
# 一本道のレース

- 一本道のレースのイメージを持っていると「自分に教えられることはない」と考えがち



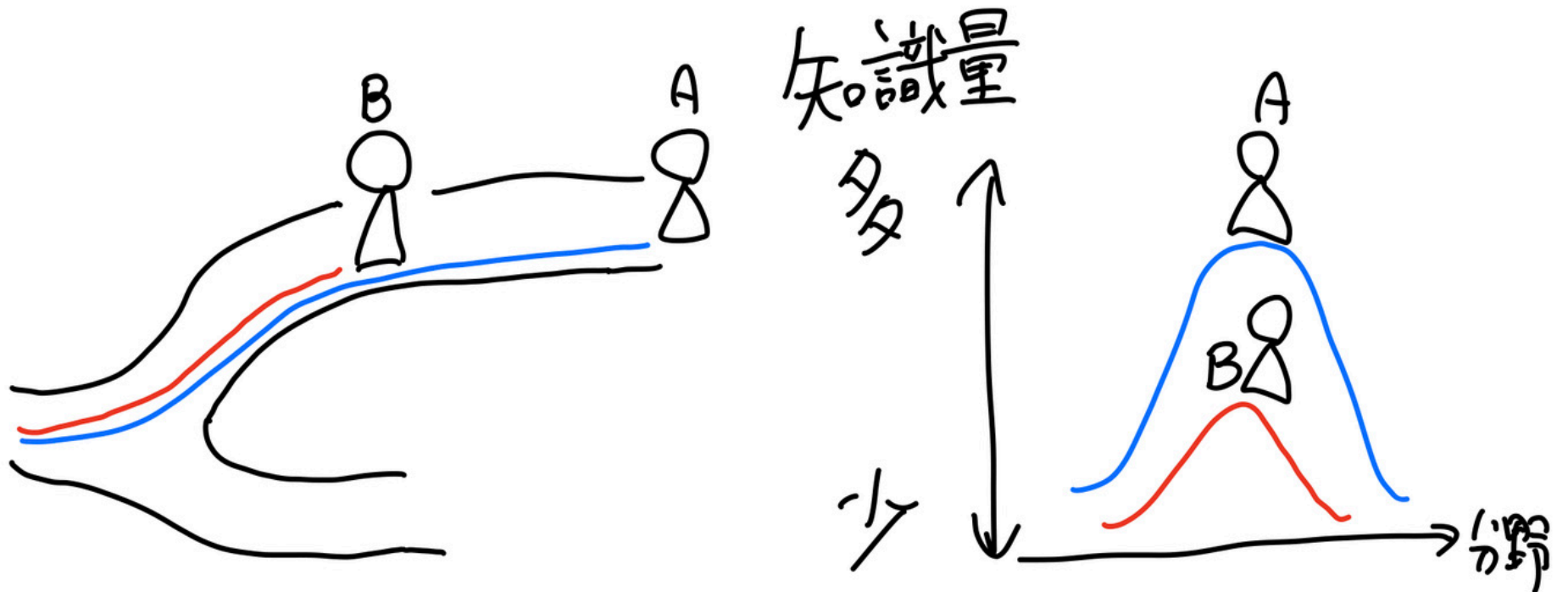
# 探索のゲーム

- 道が枝分かれしている→知識量の多いAさんも自分が通らなかった道のことは知らない
- お互いに教え合うことメリットが生まれる



# 同じ道を通ると

- 同じ道を通ると、一本道のレースと同じになる
- 教え合うことができない



# 探索のゲーム

- 「正解の道」はない
- どの道を通っても良い
- 同じ道を通った人とは知識の交換ができない
- 他の道を通った人とは、知識を交換してお互いに学び合うことができる
- 知識を他人に与える=**アウトプット**

# Give & TakeはGiveから始まる<sup>56</sup>

- どんどんアウトプットして人に与える
- 情報は情報を発信する人のところに集まる
- もしかするとうまく伝わらないかもしれない
- 間違った情報を書いて批判されるかもしれない
- でもキャッチボールと同じ。投げ損ねることを恐れて投げないと、投げられるようにならない。

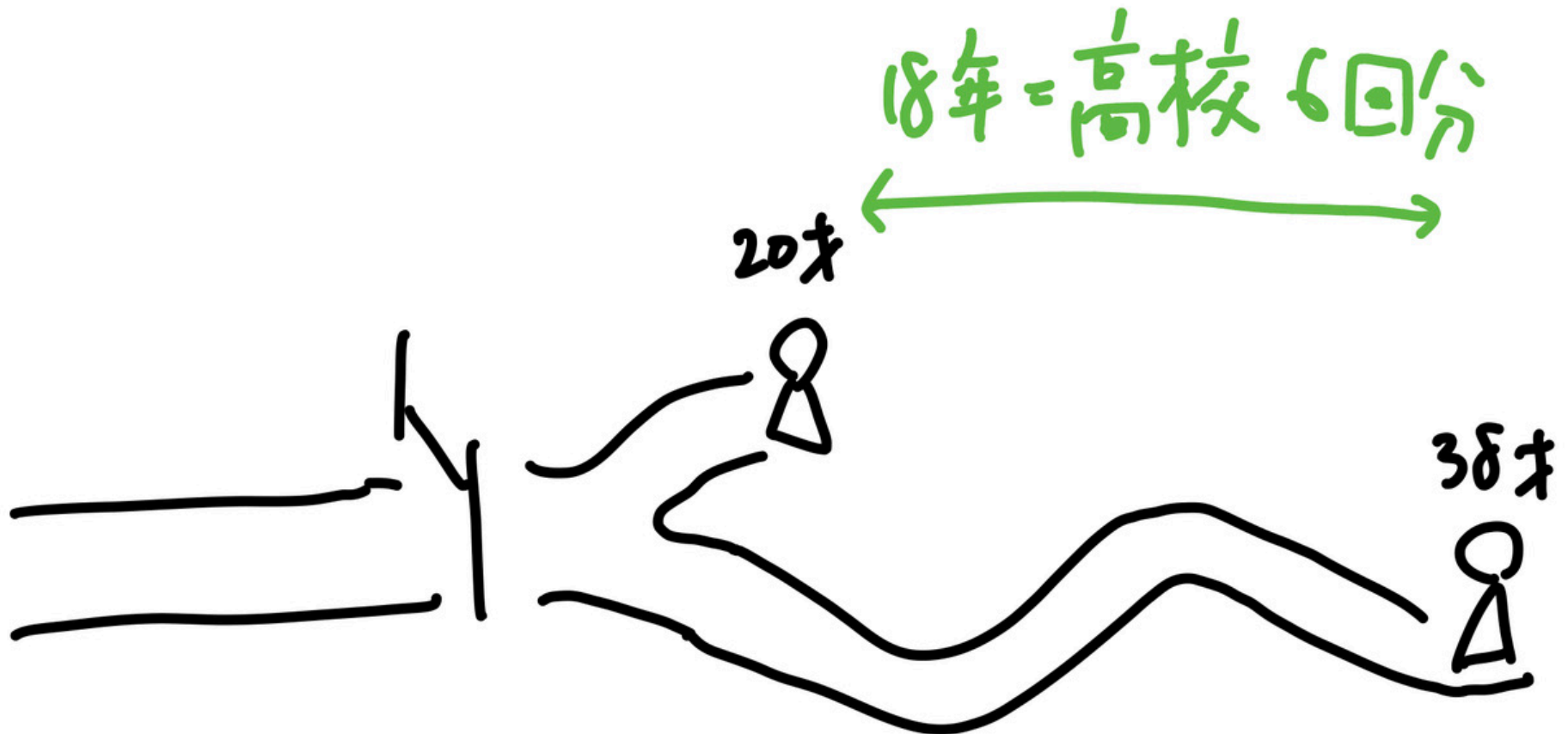


# 何をアウトプット？

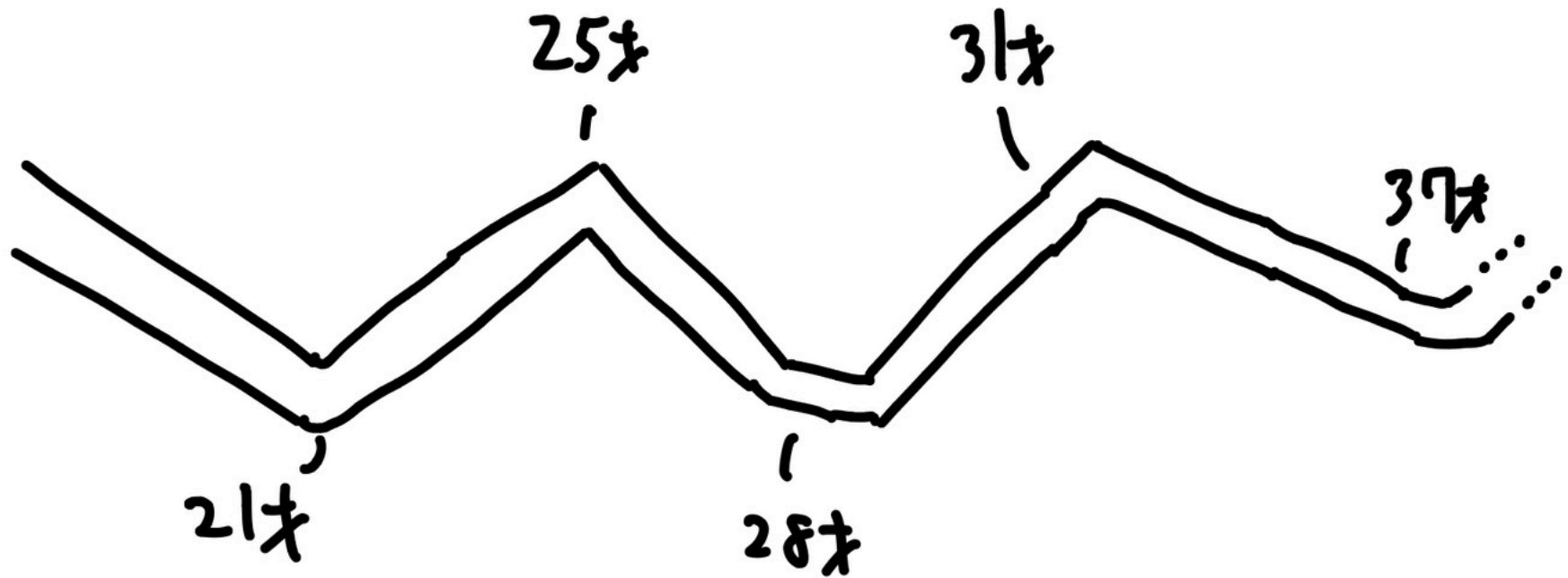
- Q: 何をアウトプットしたらいいかわからない
- A: あなたが自分で経験したことをメインにする
  - あなたと全く同じ経験をした人はいない
  - その経験について、あなたは世界一詳しい

# 人生は長い

- あなたが僕の年になるまでに高校6回分ある



# ざっくり個人史



21:大学を変える 25:就職 28:再び大学に

31:「コーディングを支える技術」出版

→プログラミング言語より「学び方」に関心が移る

37:「エンジニアの知的生産術」出版

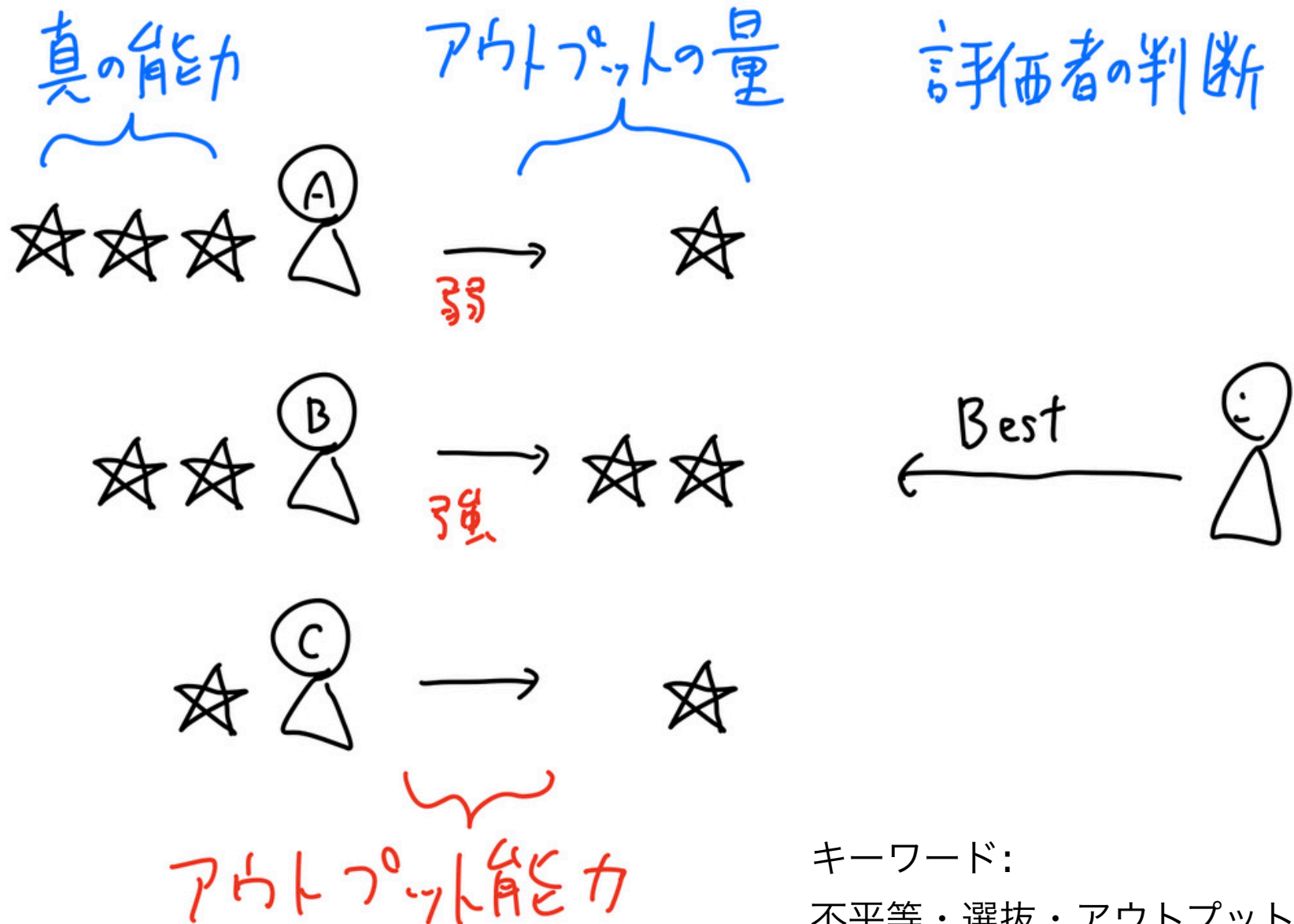
# 3～4年

- 3～4年で「今までと違う道」に進んでる
- それを5回やったのが今
- みなさんも最近「高校」から「大学」に進んだ
- 3～4年でまた「今までと違う道」に進む
- 今後あなたがどの道を選んで進むのか、  
それを決めるのはあなた自身

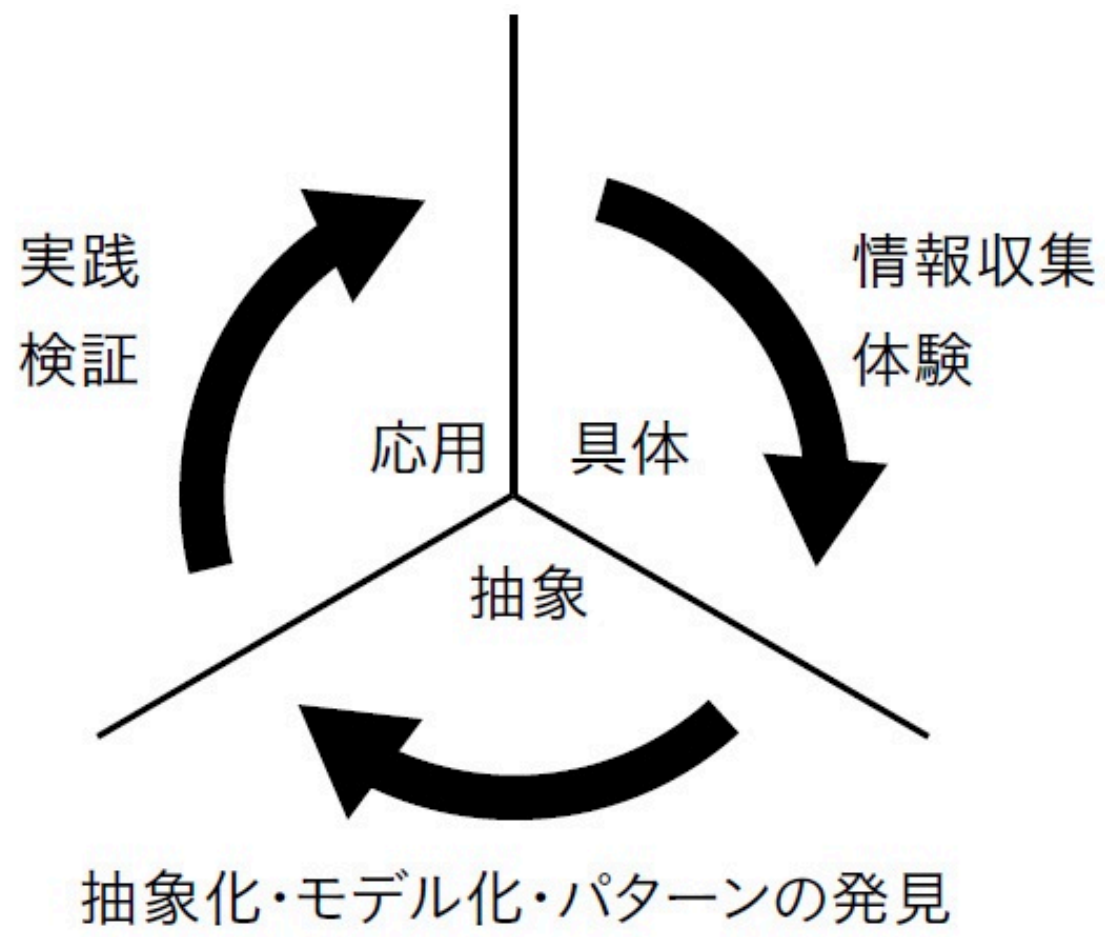
# まとめ

- なぜアウトプット(why)
- どうアウトプット(how)
- 何をアウトプット(what)

# なぜアウトプット



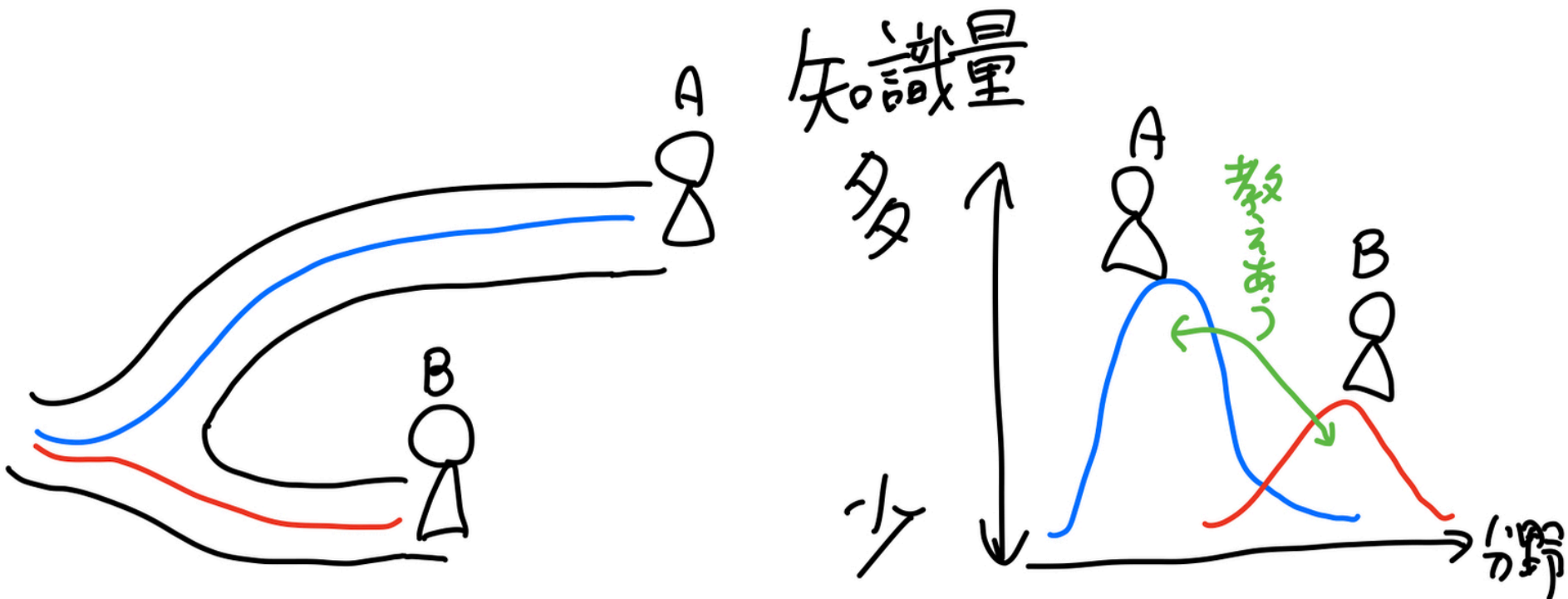
# どうアウトプット



キーワード:  
消えなくする  
一覧する  
構造を見つける  
キャッチボール  
実践と失敗

学びは情報収集・モデル化・検証の繰り返し

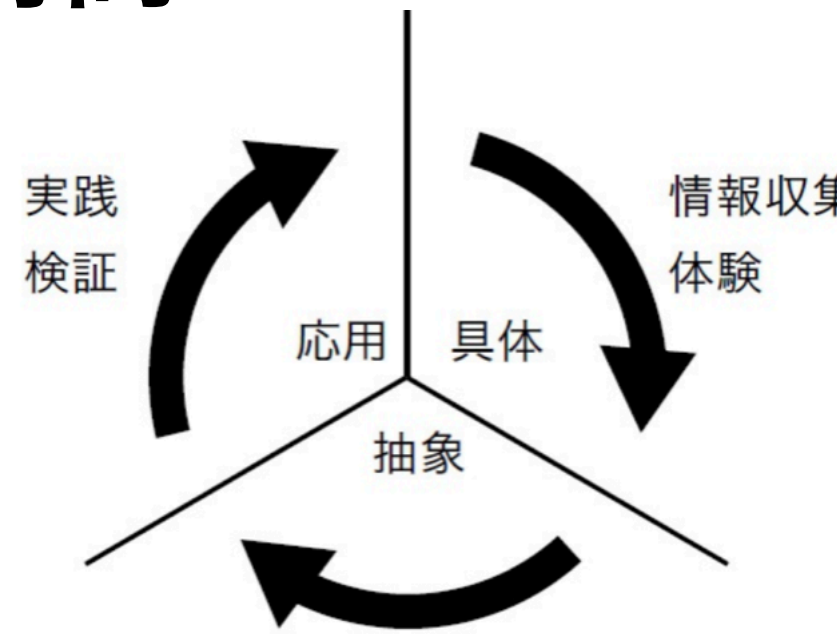
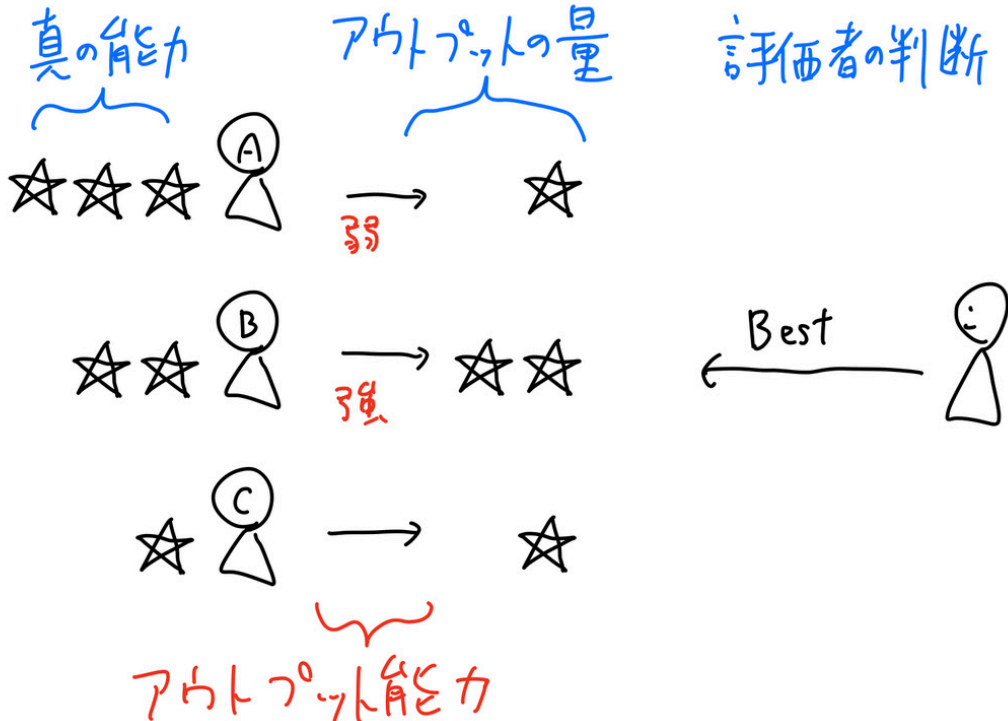
# 何をアウトプット



キーワード：探索のゲーム・人と違う道・教えあい・自分の経験をアウトプット



# 質疑の時間

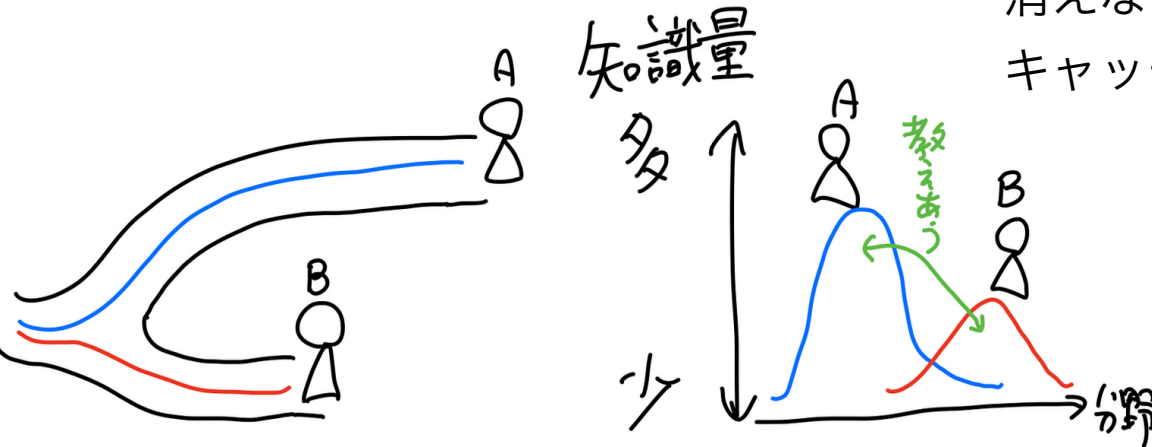


抽象化・モデル化・パターンの発見

学びは情報収集・モデル化・検証の繰り返し

消えなくする・一覧する・構造を見つける  
キャッチボール・実践と失敗

不平等・選抜・アウトプット能力



探索のゲーム・人と違う道  
教えあい  
自分の経験をアウトプット

# 質疑=キャッチボール

- ボールがすぐに投げ返されてこなかった
- この時にすぐに次のボールを投げるべきか？
- 落ち着いて投げ返すための時間が必要では？  
投げ返されてくるのをじっくり待つべきでは？

# ボールを投げ返す時間

- 他人の質疑応答を聞きながら、質問したいことをまとめるのは難易度が高い
- 質問したいことを考える時間を明示的に用意した方が良いのではないか

# よくある質疑時間

- 「誰か質問のある人～」
- 誰かが手をあげる
- 当てる
- 他の人は聞くのに精一杯で  
次の質問をまとめる時間がない

# 今回の質疑応答

- まず**1分間シンキングタイム**を開始する
- 紙に書いて考え、質問が決まったら手をあげる
  - すぐに会話を始めない
  - 番号を振るので質問者は自分の番号をメモ
- シンキングタイムが終了したら  
番号の順に質問を聞いて答えていく
- 全部終わったらまたシンキングタイムに入る