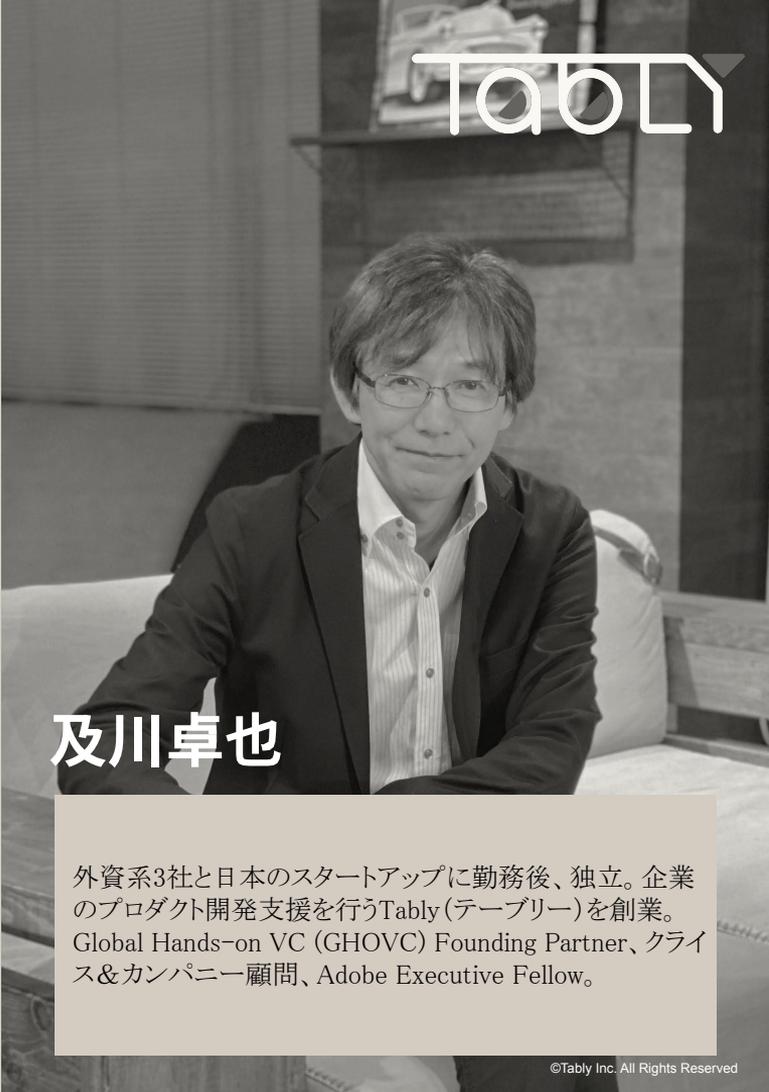


生成AI時代における 人間の情熱とプロダクト志向

Tably(テーブル)株式会社
及川卓也

Self Introduction

自己紹介

A black and white portrait of Takashi Ikawa, a man with glasses, wearing a dark blazer over a light-colored shirt, sitting on a sofa. The background is a simple room with a shelf and a lamp.

及川卓也

外資系3社と日本のスタートアップに勤務後、独立。企業
の製品開発支援を行うTably(テーブル)を創業。
Global Hands-on VC (GHOVC) Founding Partner、クライ
ス&カンパニー顧問、Adobe Executive Fellow。

HISTORY

digital

プリセールス
エンジニア

開発エンジニア

アーキテクト 兼
プロジェクトリード

Microsoft

エンジニアリング
リード

開発責任者

Google

プロダクト
マネージャー

エンジニアリング
マネージャー

Increments⁺
Qiita

プロダクト
マネージャー

TABLY

独立へ



「卓」が物を引き上げるように

人や事業、物事を社会に役立つ高さまで
テクノロジーで引き上げる

<http://tably.rocks>

Passion

- **Windows**の可能性
- **Web**の可能性
- ブラウザの再発明
- 災害復興での**IT**活用
- ...

digital

プリセールス
エンジニア

開発エンジニア

アーキテクト 兼
プロジェクトリード

Microsoft

エンジニアリング
リード

開発責任者

Google

プロダクト
マネージャー

エンジニアリング
マネージャー

Increments⁺
Qiita

プロダクト
マネージャー

TABLY

独立へ



開発しているプロダクトが

誰(個人・組織)

を

どんな状態

にするか説明できますか？

本日のお話



そのプロダクト開発に

人

を

引き込めますか？



急激な進化を遂げる



をどう取り入れるか

AIについて過去の認識

及川卓也 Oikawa Takuya

あらゆる
ビジネスを
一変させる
最強戦略

ソフトウェア

デジタル化 AI活用 サブスク導入

マイクロソフト
やグーグルで
学んだ方法論を
日本企業で
応用する

成功に向けて
IT開発の
主導権を手に入れる

日経BP

2019年発売

悩み1：今後エンジニアの価値が下がるのではないかと？

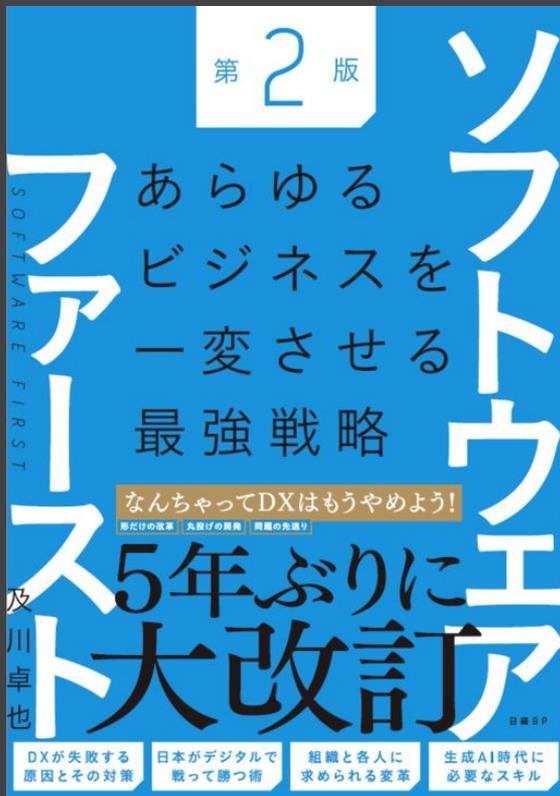
今日は人が担当している実装の仕事も、やがて陳腐化し、機械によって置き換えられるのではないかと？という話をよく聞きます。例えば「AIがこのまま進化すれば、ソフトウェアのプログラムはAIが自動生成するようになるだろう」「であれば自社にエンジニアを抱える必要がないのではないかと？」という懸念です。

A：今は人間にしかできない複雑な要件定義が求められる。

AIによってプログラムが自動生成されるという未来はいつかは来るでしょう。筆者が社会人になったばかりの80年代にもそんな話がありました。ただ、それから30年近く経つのに、そんな未来はまだ来ていません。いつ来るかも分からない未来に期待して、今何もせずには、デジタル化の流れに乗り遅れて会社が潰れてしまうでしょう。

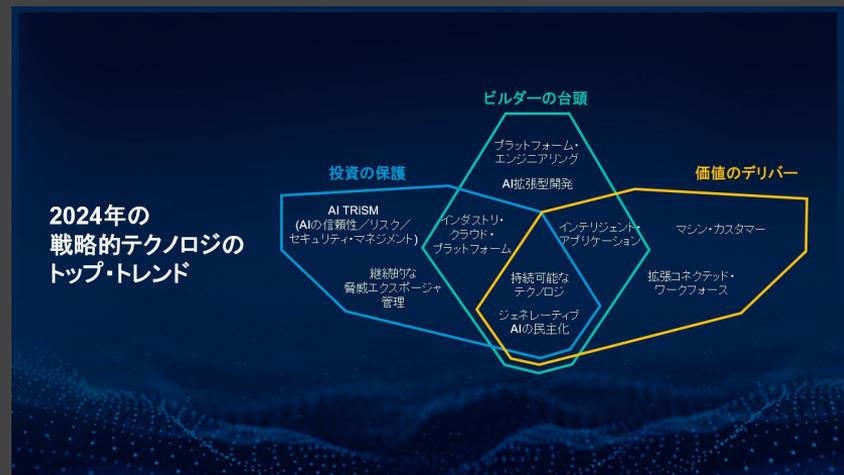
予想は完全に外れた

AIについての現在の状況



ソフトウェアファースト 第2版（2024年9月発売）では生成AIを含むAI活用について大幅に追記。

ガートナーの戦略的テクノロジートレンドには、「AI拡張型開発」が言及される。



一緒に考えていきましょう

生成AIの活用が不可欠となる中で、
人が情熱を持って開発を進めるには？

Agenda

- プロダクト開発における生成AIの活用
- 情熱駆動開発
- 情熱 × AI

Agenda

- プロダクト開発における生成AIの活用
- 情熱駆動開発
- 情熱 × AI

プロダクト開発における生成AI活用

- プロダクトマネジメント
- プロジェクトマネジメント
- UXリサーチ&デザイン
- UIデザイン
- ソフトウェア開発
- 品質確認（ソフトウェアテスト）

急速に活用が進む

労働市場へのインパクト

WORKING PAPER

GPTs are GPTs: An Early Look at the Labor Market Impact Potential of Large Language Models

Tyna Eloundou¹, Sam Manning^{1,2}, Pamela Mishkin^{1*}, and Daniel Rock³

¹OpenAI

²OpenResearch

³University of Pennsylvania

March 27, 2023

Abstract

We investigate the potential implications of large language models (LLMs), such as Generative Pre-trained Transformers (GPTs), on the U.S. labor market, focusing on the increased capabilities arising from LLM-powered software compared to LLMs on their own. Using a new rubric, we assess occupations based on their alignment with LLM capabilities, integrating both human expertise and GPT-4 classifications. Our findings reveal that around 80% of the U.S. workforce could have at least 10% of their work tasks affected by the introduction of LLMs, while approximately 19% of workers may see at least 50% of their tasks impacted. We do not make predictions about the development or adoption timeline of such LLMs. The projected effects span all wage levels, with higher-income jobs potentially facing greater exposure to LLM capabilities and LLM-powered software. Significantly, these impacts are not restricted to industries with higher recent productivity growth. Our analysis suggests that, with access to an LLM, about 45% of all worker tasks in the US could be completed significantly faster at the same level of quality. When incorporating software and tooling built on top of LLMs, this share increases to between 47 and 56% of all tasks. This finding implies that LLM-powered software will have a substantial effect on scaling the economic impacts of the underlying models. We conclude that LLMs such as GPTs exhibit traits of general purpose technologies, indicating that they could have considerable economic, social, and policy implications.

1 Introduction

As shown in Figure 1, recent years, months, and weeks have seen remarkable progress in the field of generative AI and large language models (LLMs). While the public often associates LLMs with various iterations of the Generative Pre-trained Transformer (GPT), LLMs can be trained using a range of architectures, and are not limited to transformer-based models (Devlin et al., 2019). LLMs can process and produce various forms of sequential data, including assembly language, protein sequences and chess games, extending beyond natural language applications alone. In this paper, we use LLMs and GPTs somewhat interchangeably, and specify in our rubric that these should be considered similar to the GPT-family of models available via ChatGPT or the OpenAI Playground (which at the time of labeling included models in the GPT-3.5 family but not in the GPT-4 family). We examine LLMs with text- and code-generating abilities, use the term “generative AI” to additionally include modalities such as images or audio, and use “LLM-powered software” to cover tools built on top of LLMs or that combine LLMs with other generative AI models.

*Corresponding author (pamela@openai.com). Authors contributed equally and are listed alphabetically.

GPTs are GPTs: An Early Look at the Labor Market Impact Potential of Large Language Models (2023 by OpenAI)

労働市場への影響の早期考察 ～ 大規模言語モデルの可能性

- 賃金の高い仕事は、インパクトを受ける
- プログラミングやライティングは、リスクが高い
- 職業として参入障壁が高いものほど、リスクが高い
- 科学やクリエイティブな役割は、リスクが低い

プロダクト開発における生成AI活用

- プロダクトマネジメント
- プロジェクトマネジメント
- UXリサーチ&デザイン
- UIデザイン
- ソフトウェア開発
- 品質確認（ソフトウェアテスト）



生成AIの登場により
大きなインパクトを受ける

プロダクト開発における生成AI活用

- プロダクトマネジメント
- プロジェクトマネジメント
- UXリサーチ&デザイン
- UIデザイン
- **ソフトウェア開発**
- 品質確認（ソフトウェアテスト）

ソフトウェア開発（プログラミング）においては、活用しないという選択肢はない

生成AIの活用も含めて、ソフトウェア開発手法・能力となっている

プログラミングを例に、生成AIを再考

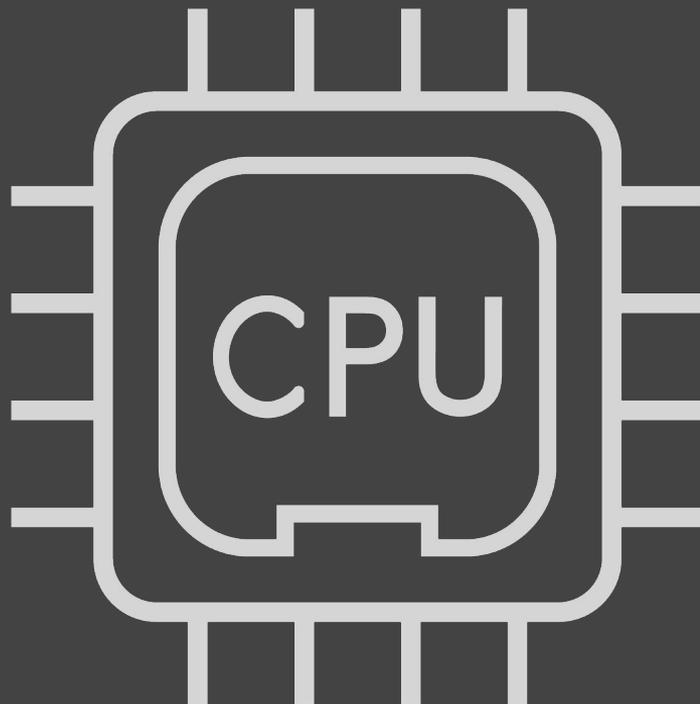
```
if (groupinfo->bblocks[0] != group_info->small_block) {
    int i;
    if (groupinfo->bblocks[0] != group_info->small_block) {
        for (i = 0; i < group_info->nblocks; i++)
            int i;
        freepage((unsigned long)groupinfo->bblocks[i]);
        for (i = 0; i < group_info->nblocks; i++)
            freepage((unsigned long)groupinfo->bblocks[i]);
        }
        kFree(groupinfo);
    }
    kFree(groupinfo);
}
EXPORTSYMBOL(groupsfree);

int groups_touser(gid_t user *grouplist,
/* export the groupinfo to a user-space array */
const struct group_info *group_info)
static int groups_touser(gid_t user *grouplist,
{
    const struct group_info *group_info)
    int i;
    unsigned int count = groupinfo->ngroups;
    int i;
    unsigned int count = groupinfo->ngroups;
    for (i = 0; i < group_info->nblocks; i++) {
        unsigned int cpcount = min(NGROUPSPERBLOCK, count);
        for (i = 0; i < group_info->nblocks; i++) {
            unsigned int len = cpcount * sizeof(*grouplist);
            int cpcount = min(NGROUPSPERBLOCK, count);
            cpcount * sizeof(*grouplist);
            }blocks[i], len))
```

プログラミングの進化

1+1=2

```
0000: B8 01 00 00 00  
0005: 83 C0 01  
0008: 89 EB  
000A: C3
```



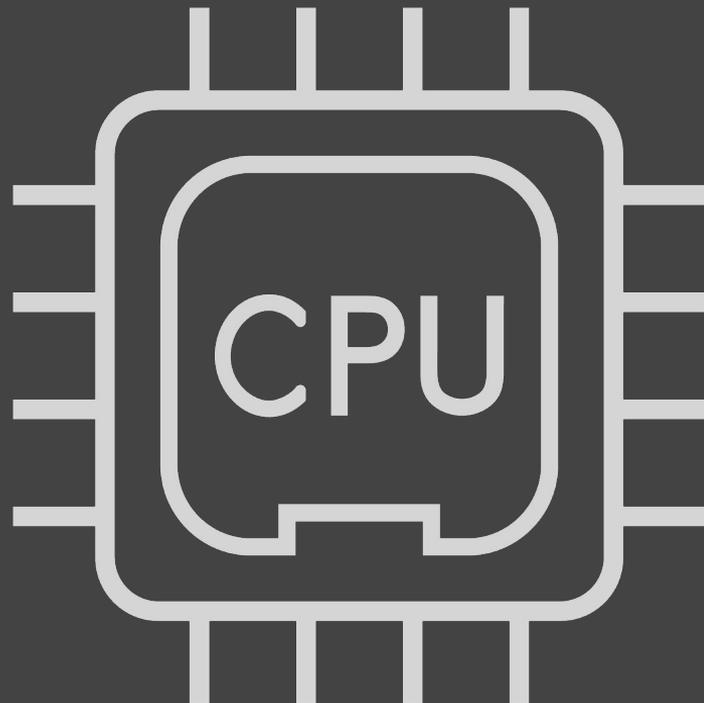
プログラミングの進化

1+1=2

```
global _start  
  
section .text  
_start:  
  mov eax, 1  
  add eax, 1  
  mov ebx, eax  
  ret
```



```
0000: B8 01 00 00 00  
0005: 83 C0 01  
0008: 89 EB  
000A: C3
```



```
section .data
```

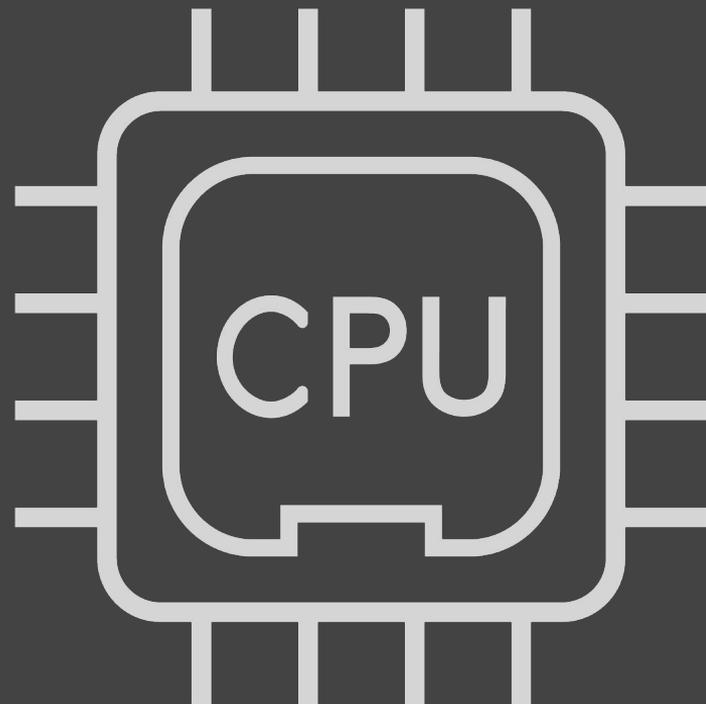
プログラミングの進化

1+1=2

```
int main() {  
    int a = 1;  
    int b = 1;  
    int c = a + b;  
    return 0;  
}
```



```
0000: B8 01 00 00 00  
0005: 83 C0 01  
0008: 89 EB  
000A: C3
```



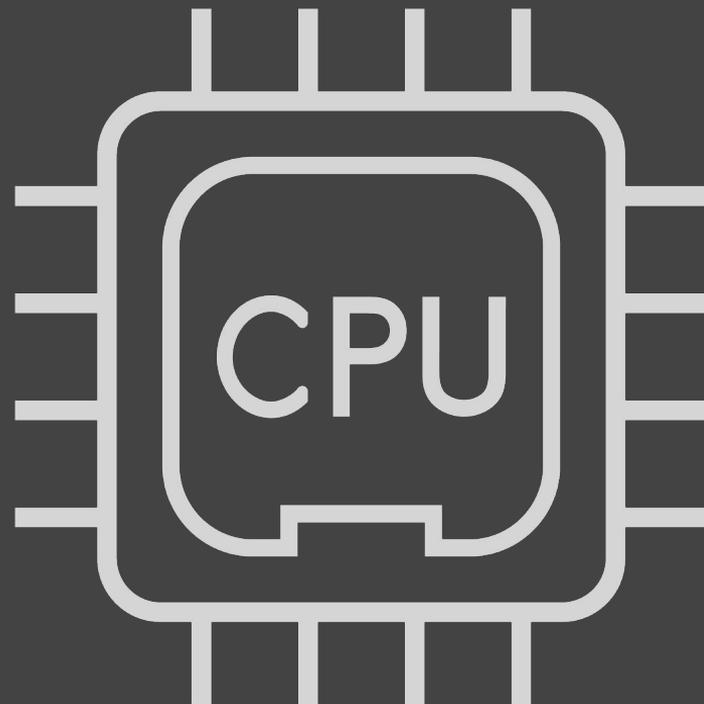
プログラミングの進化

1+1=2

```
int main() {  
    int a = 1;  
    int b = 1;  
    int c = a + b;  
    return 0;  
}
```



```
0000: B8 01 00 00 00  
0005: 83 C0 01  
0008: 89 EB  
000A: C3
```



どんどん人間に易しくなっている

プログラミングの進化

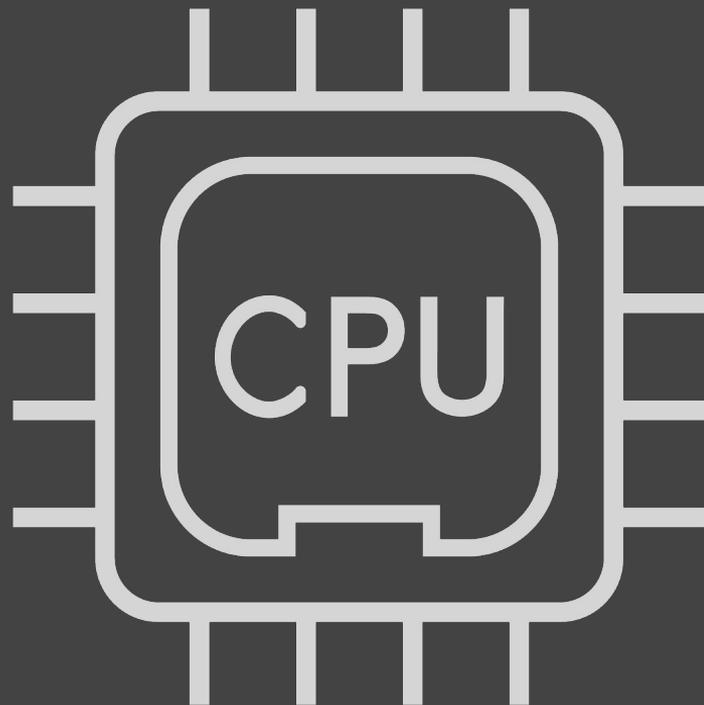
1+1=2

1+1を求め
るプログラ
ムを作って



```
int main() {  
  int a = 1;  
  int b = 1;  
  int c = a + b;  
  return 0;  
}
```

```
0000: B8 01 00 00 00  
0005: 83 C0 01  
0008: 89 EB  
000A: C3
```



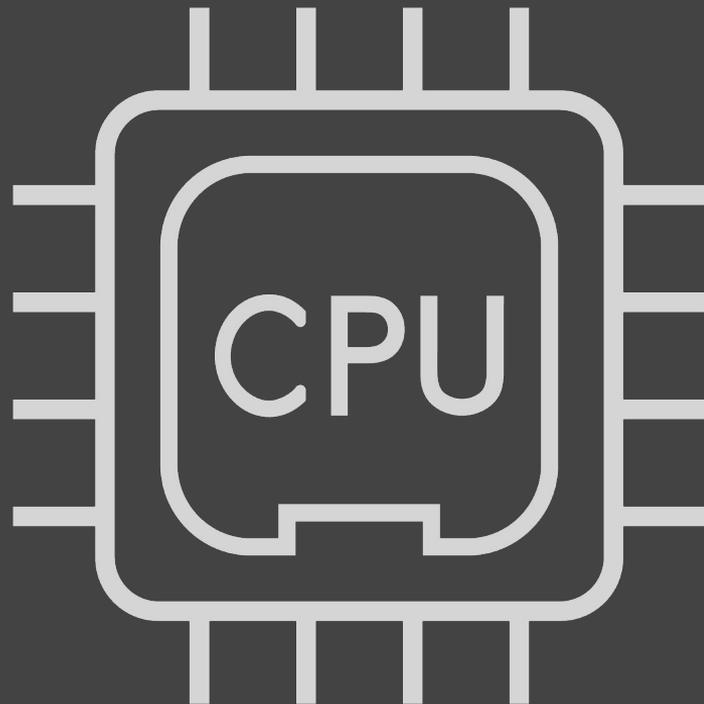
プログラミングの進化

1+1を求め
るプログラ
ムを作って



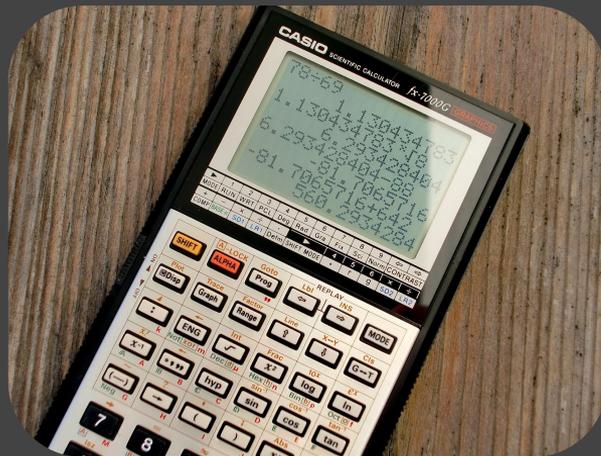
```
int main() {  
    int a = 1;  
    int b = 1;  
    int c = a + b;  
    return 0;  
}
```

```
0000: B8 01 00 00 00  
0005: 83 C0 01  
0008: 89 EB  
000A: C3
```



コンピューターに指示を出すことに変わりはない。具体的な指示を出そうとすると、今のプログラミングと本質的には同じことをすることになる。

算数・数学をなぜ学ぶのかをヒントにする



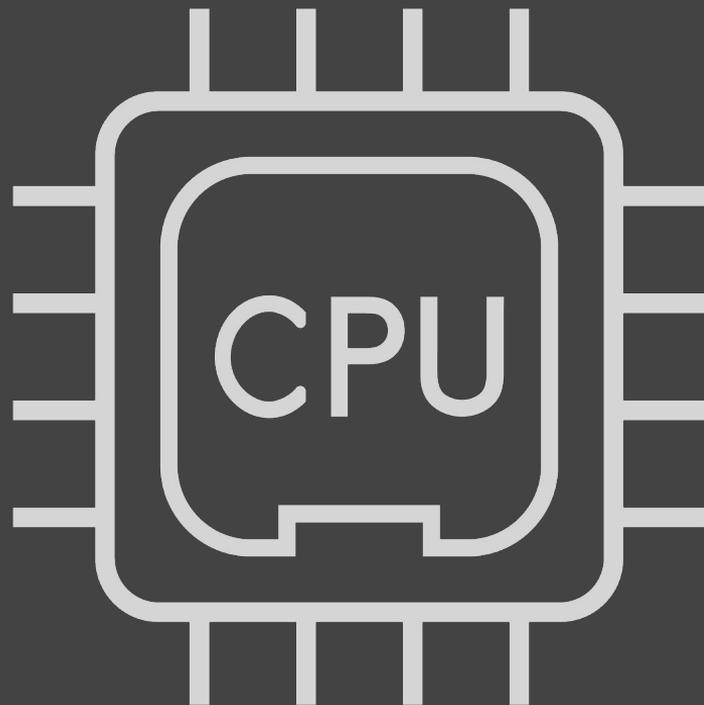
手や算盤、計算尺で計算していたものが電卓になり、そして表計算ソフトになったが、算数・数学は学び続けている

1+1=2

```
int main() {  
    int a = 1;  
    int b = 1;  
    int c = a + b;  
    return 0;  
}
```



```
0000: B8 01 00 00 00  
0005: 83 C0 01  
0008: 89 EB  
000A: C3
```



必要ならばバイナリにパッチ

二極化が進むソフトウェア開発人材の今後

コンピューターに直接指示を出す
(AIが支援)

AIに任せられない・任すべきではない仕事をする人材

- 教科書に書かれていること以上の判断が必要となる領域
- 倫理的に人間が作業すべき領域
- AIを積極的に活用するが、あくまでも支援されるのみ

AIに依頼する
(AIが主導)

AIに任せられる領域を担当する人材

- 従来のプログラミング作業は不要
- コンピューターへの指示はより抽象度の高い(機械からは遠い)ものとなる
- 従来のソフトウェア開発者の仕事の内容が変わるか、失業
- 従来はソフトウェア開発者でなかった人が参入

二極化が進むソフトウェア開発人材の今後

コンピューターに直接指示を出す
(AIが支援)

シニア
エンジニア

AIに依頼する
(AIが主導)

新卒
エンジニア

二極化が進むソフトウェア開発人材の今後

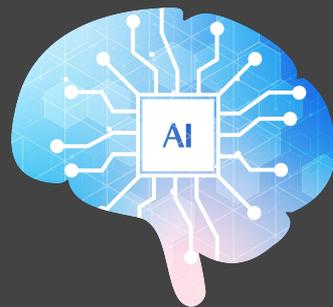
コンピューターに直接指示を出す
(AIが支援)

AIに依頼する
(AIが主導)

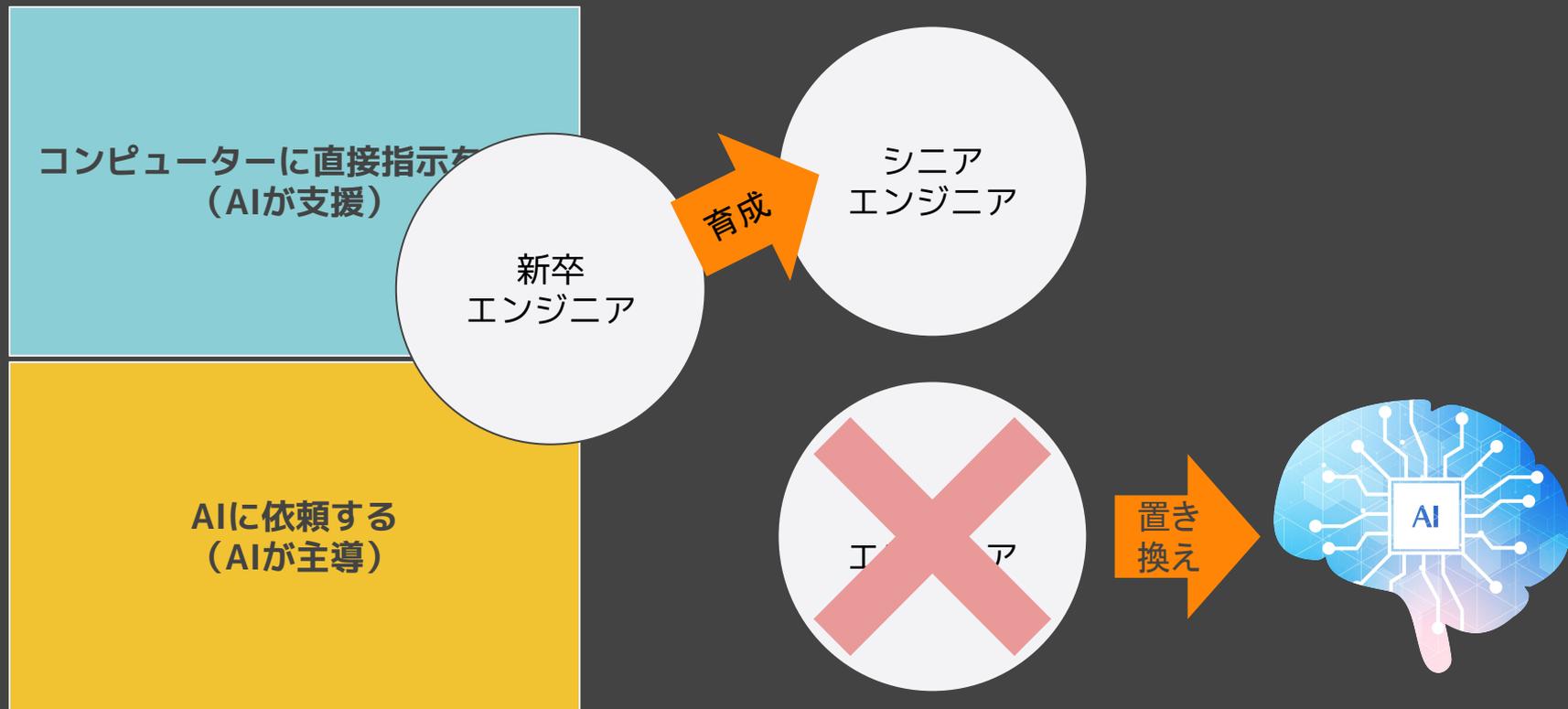
シニア
エンジニア

~~ジュニア~~

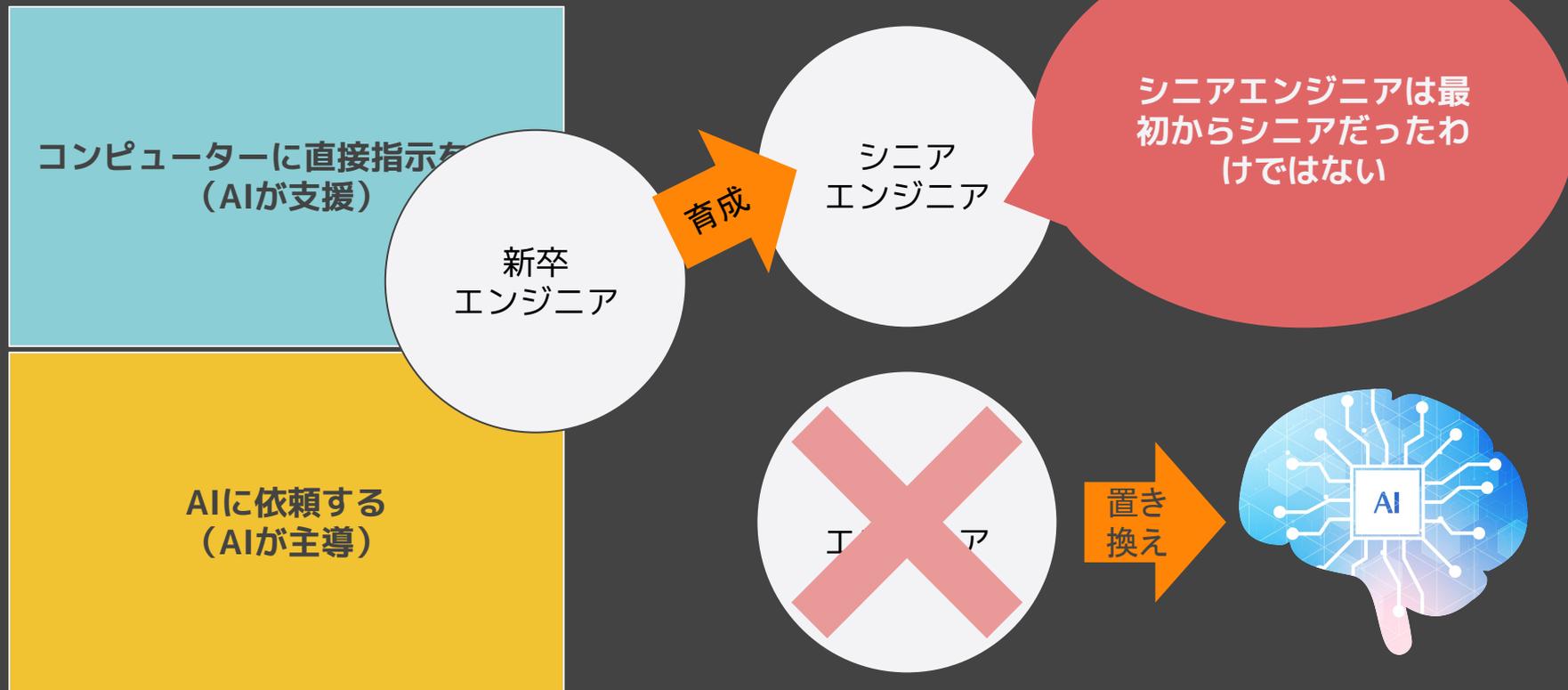
置き
換え



二極化が進むソフトウェア開発人材の今後



シニアエンジニアには一日にしてならず



プロダクト開発における生成AI活用

- プロダクトマネジメント
- プロジェクトマネジメント
- UXリサーチ&デザイン
- UIデザイン
- ソフトウェア開発
- 品質確認（ソフトウェアテスト）



- ペルソナ作成
- ユーザーインタビュー
 - ロールプレイ
 - ログ分析
- PRD作成



canvas



*Innovation
belongs here*

プロダクト開発における生成AI活用

- プロダクトマネジメント
- プロジェクトマネジメント
- UXリサーチ&デザイン
- UIデザイン
- ソフトウェア開発
- **品質確認（ソフトウェアテスト）**



Autify Genesis (β)

生成AIによる
テストケース・テストシナリオ
自動生成

プロダクトの仕様書や
機能の設計書

Generate ↓

テストケース

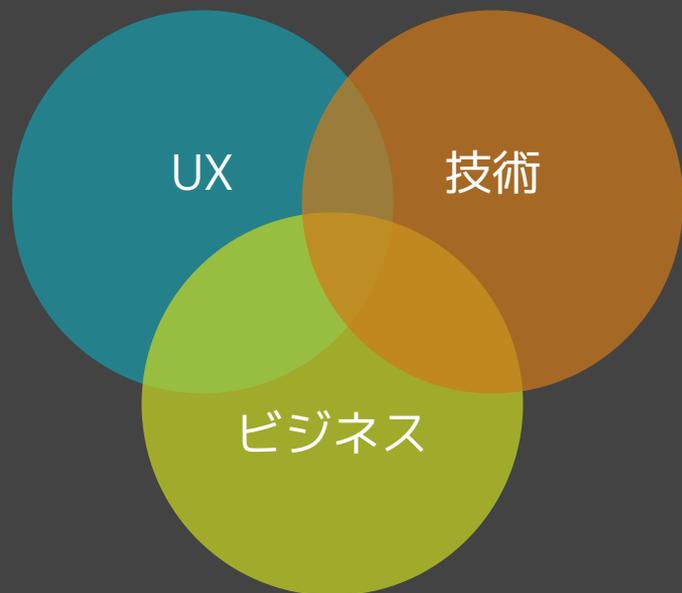


テストシナリオ

一方、生成AI（だけ）では
できないこと



プロダクトの意思決定



プロダクトを成功に結びつけるには、ときに**トレードオフ**も生じるプロダクトの意思決定を行なう必要がある。

プロダクトの意思決定には「絶対の正解」はない。



マイケル・ポーター

戦略とは何をしないかを定めることだ

トレードオフとは、企業がある戦略を選択する際に他の選択肢を放棄すること。企業が明確な戦略を持つためには、何をしないかを決定することが不可欠

同じ寿司店でも…



高級寿司店



回転寿司チェーン

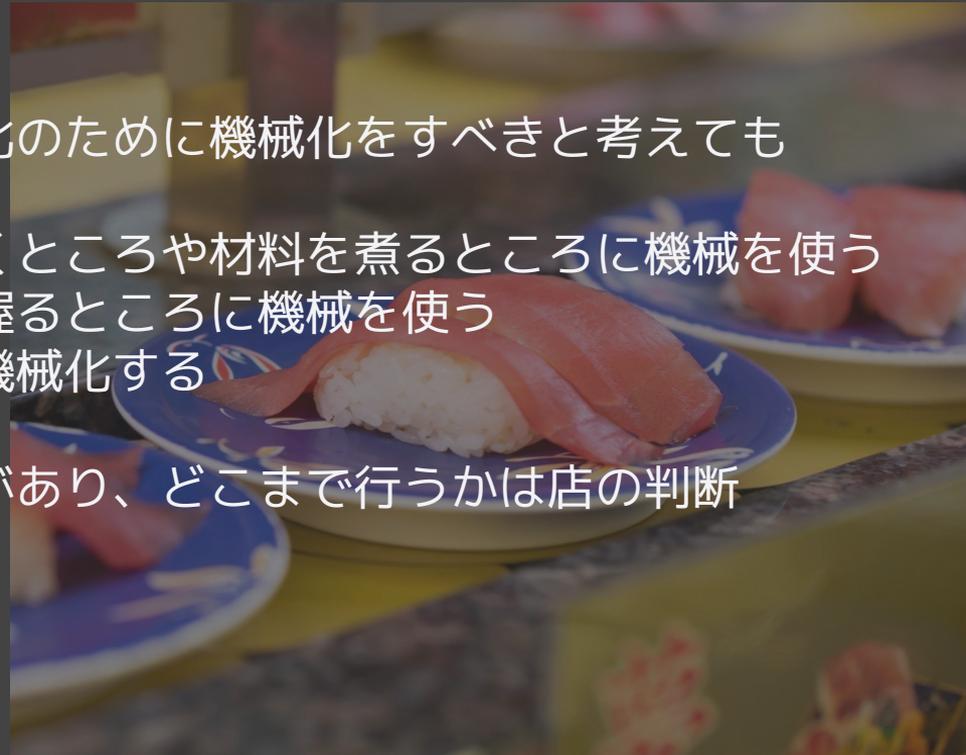
同じ寿司店でも…



効率化・省人化のために機械化をすべきと考えても

- 白米を炊くところや材料を煮るところに機械を使う
- シャリを握るところに機械を使う
- すべてを機械化する

などの選択肢があり、どこまで行うかは店の判断



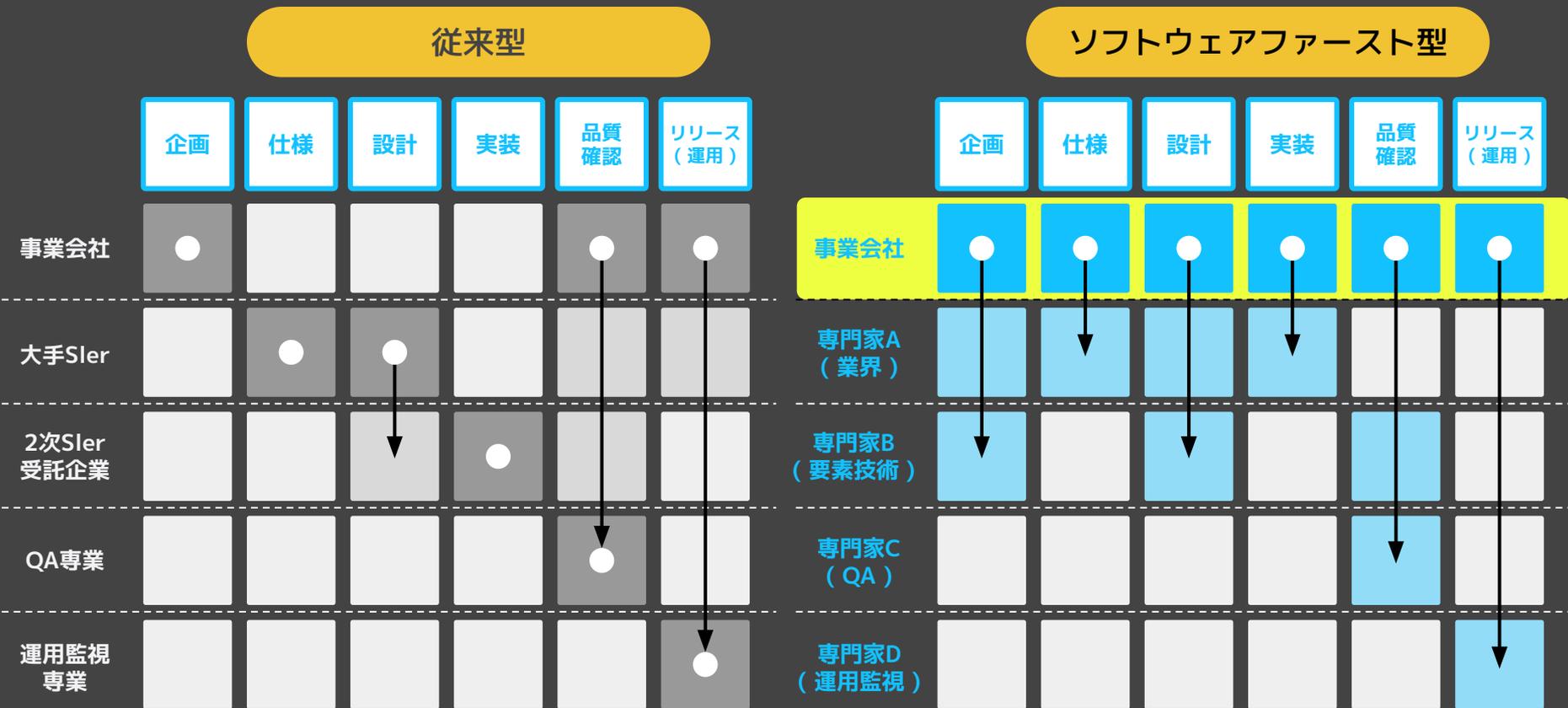
高級寿司店

回転寿司チェーン

避けるべきは 丸投げ



開発工程と担当領域



必要なのは「手の内化」

- コア部分を外部委託せずに、**ノウハウを自らの「手の内」に収める。**
- コア要素に**ブラックボックスとなる部分を残さない。**
- △ 内製化を単に進める。

必要なのは「手の内化」

- コア部分を外部委託せずに、**ノウハウを自らの「手の内」に収める。**
- コア要素に**ブラックボックスとなる部分を残さない。**
- △ 内製化を単に進める。
- × **AIにすべて任せる。**

Agenda

- プロダクト開発における生成AIの活用
- **情熱駆動開発**
- 情熱 × AI

情熱とは

人間の行動のエネルギー源

動機と情熱

2つの動機付け



外発的動機付け

義務、賞罰、強制などによる動機付け

例) 昇進や昇給に必要なので資格を習得



内発的動機付け

好奇心や関心などによる動機付け

例) ある国や文化を知りたいので、外国語を学ぶ

動機と情熱

2つの動機付け

情熱



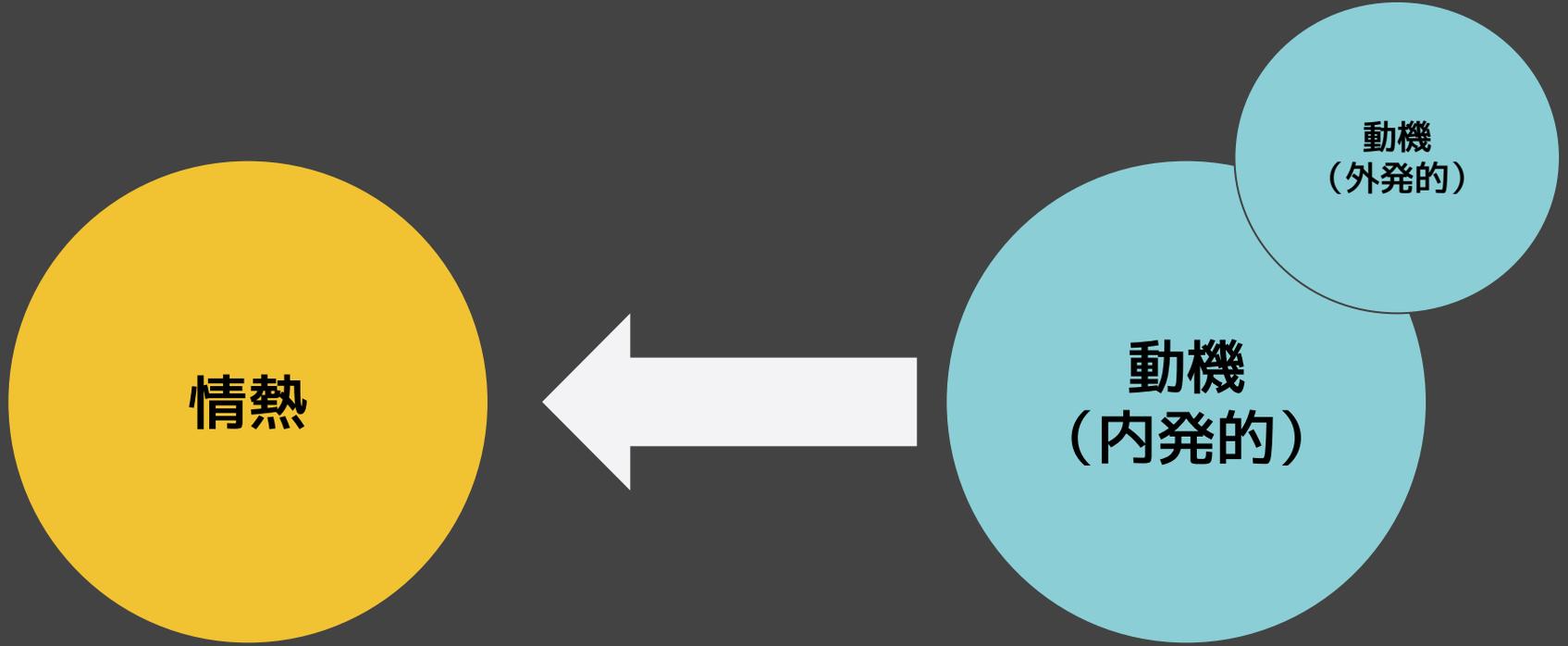
内発的動機付け

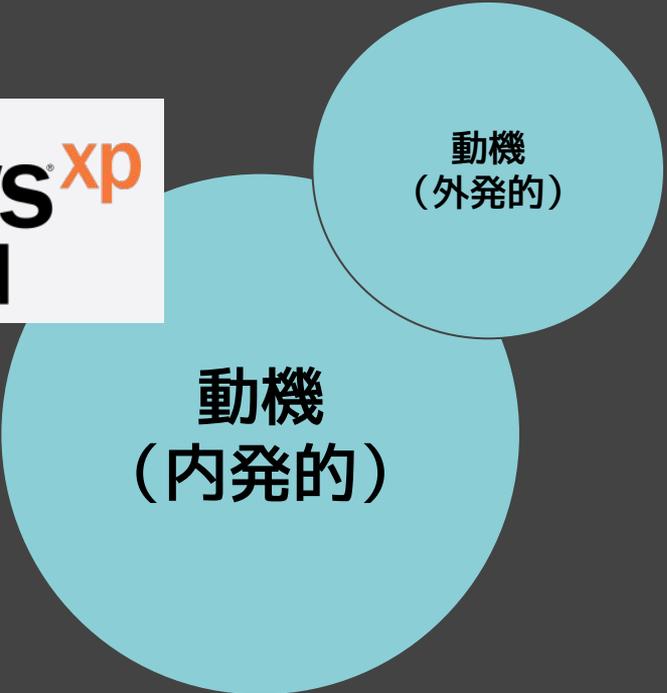
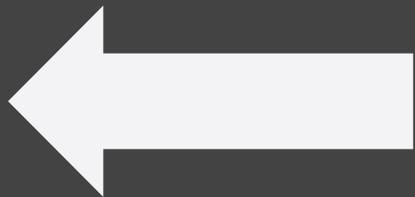
好奇心や関心などによる動機付け

例) ある国や文化を知りたいので、外国語を学ぶ

情熱

動機
(内発的)







情熱駆動開発
のすゝめ

Passion Driven Development



いま、日本に必要なのは「情熱」

- ✓ 人間の情熱やエネルギー、目的意識を開発プロセスの中心に据えよう
- ✓ 技術的な正確さや効率、ビジネスとしての成功だけでなく、「なぜ選択するのか」「なぜ実現するのか」「なぜ作るのか」「何を作るのか」「誰のために作るのか」といった人間らしい動機やビジョンに焦点を当てよう
- ✓ プロダクトの成功には幾多の苦難があるが、苦難を乗り越えられるのは情熱

情熱のベクトル

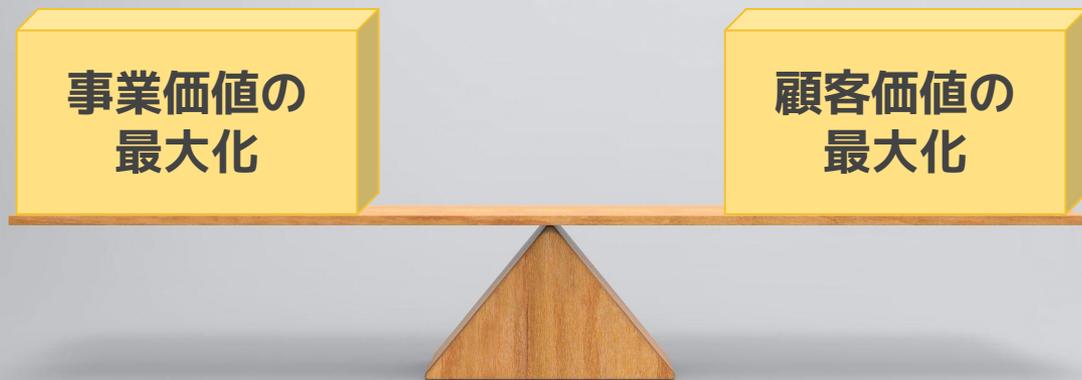
情熱とは本質に向き合うこと

様々な情熱のベクトル

- 技術
- 顧客
- 未来
- ビジネス



プロダクトの成功



プロダクトの成功に
は、この3つが等しく
重要

特に、顧客価値とビ
ジョン（未来）への情
熱を強く持つことが必
要

情熱駆動開発の例



Googleニュースは、
Googleの20%プロジェクトとして始まった、情熱駆
動開発の一例



「僕はやりたいんだ」

US進出のためのUXリサーチを支援。リサーチの結果、そのままの計画では失敗しそうなことがわかり、それを報告したときの、創業者CEOのひとつ

情熱駆動開発のための組織

- ✓ 情熱駆動開発のためには、**情熱と情熱のぶつかり合い**が不可欠
- ✓ 情熱が情熱を呼ぶための**組織文化**を醸成する
- ✓ コミュニケーションハブ&カタリストとしての**マネージャー**の役割が重要

Agenda

- プロダクト開発における生成AIの活用
- 情熱駆動開発
- 情熱 × AI

情熱駆動開発の課題

(悪い) プロダクトアウトになってしまう / 独りよがりですべてを進めてしまう

 顧客不在

 チームメンバーを置き去り

 ハラスメントリスク

 ビジネス視点なし

偉そうなことを言っている自分ですが…

初心者向け プログラミング学習サービス



type ひとつ上を目指す人の
キャリア転職サイト【タイプ】 ヘルプ

応募管理 ★ 検討中リスト スカウト AIの求人提案 転職ノウハウ

転職・求人情報サイトのtype > エンジニアtype > 働き方 > 課題より先に「作りたい」があった／及川卓也と『Jasmine Tea』の挑戦【前編】

エンジニアtype 働き方 スキル 転職 ITニュース 特集 連載

Et

“ 悪い意味の
プロダクトアウトから
始めてしまっ...” ”

及川卓也 Jasmine Teaの挑戦 #1

課題より先に「作りたい」があった／及川卓也と『Jasmine Tea』の挑戦【前編】

2024.04.08 #働き方
及川卓也 PdM プロダクト

* 現在、顧客・市場から学び、継続して改善中



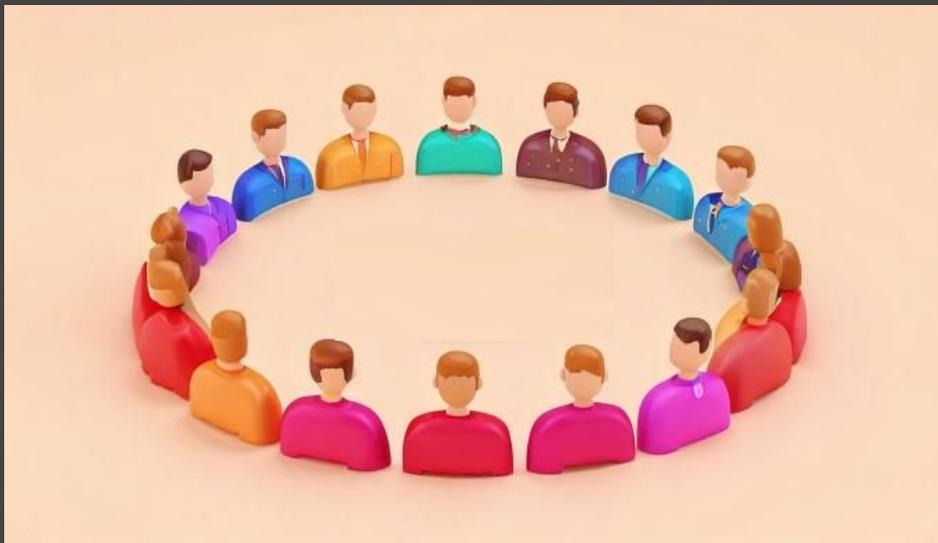
「僕はやりたいんだ」のMr.X

結果、やはり
うまくいかなかった

冷静と情熱のあいだ

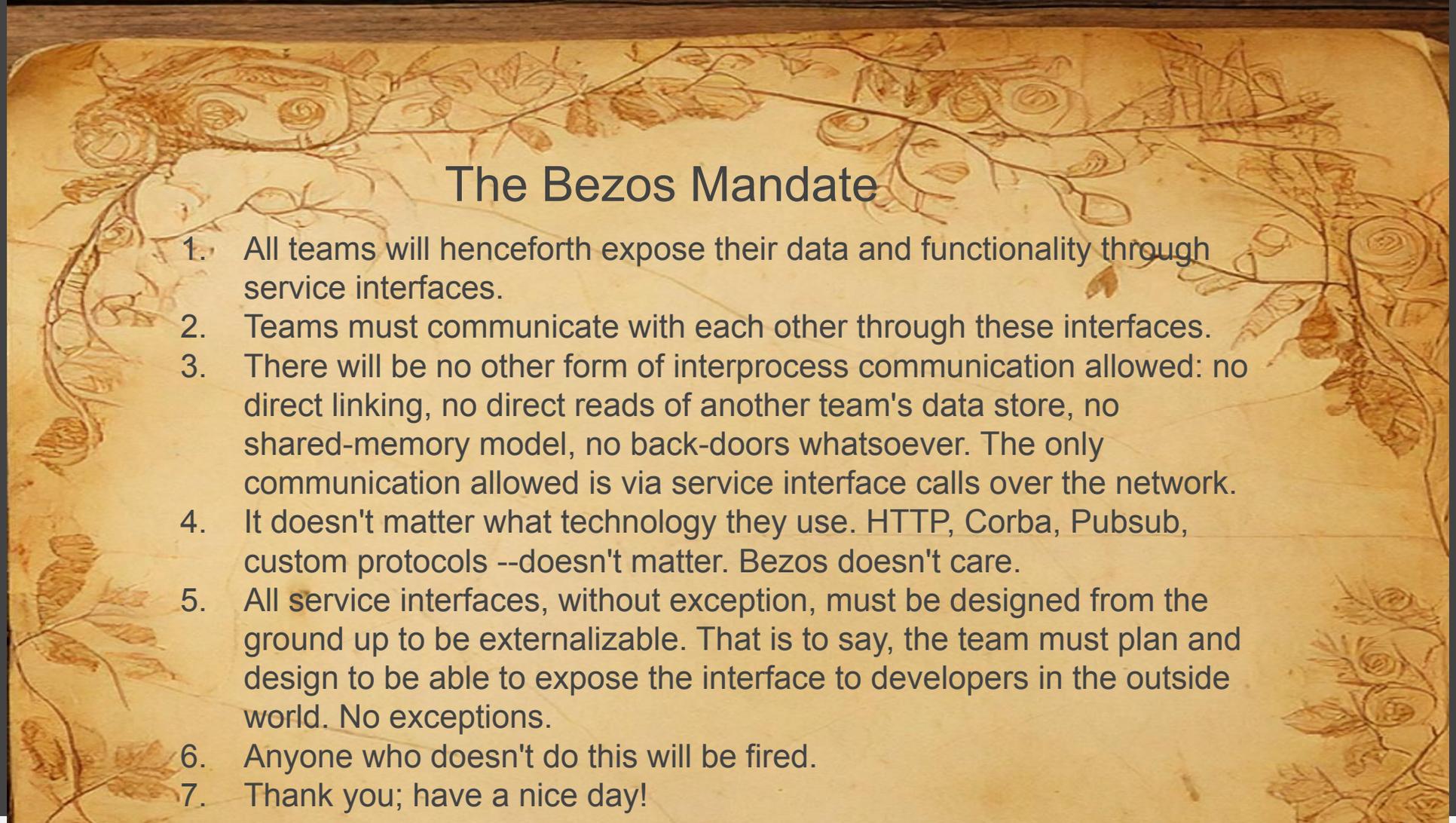
- 情熱の暴走を阻止する
 - 個人の暴走を阻止する
 - チームの暴走を阻止する

人間のコミュニケーション



基本的に人間のコミュニケーションは面倒くさい

- 意思を伝えられない
 - 伝えにくい
 - 間違っって伝わる
- 人間関係への影響
 - 関係性が壊れる
 - 評価が悪くなる



The Bezos Mandate

1. All teams will henceforth expose their data and functionality through service interfaces.
2. Teams must communicate with each other through these interfaces.
3. There will be no other form of interprocess communication allowed: no direct linking, no direct reads of another team's data store, no shared-memory model, no back-doors whatsoever. The only communication allowed is via service interface calls over the network.
4. It doesn't matter what technology they use. HTTP, Corba, Pubsub, custom protocols --doesn't matter. Bezos doesn't care.
5. All service interfaces, without exception, must be designed from the ground up to be externalizable. That is to say, the team must plan and design to be able to expose the interface to developers in the outside world. No exceptions.
6. Anyone who doesn't do this will be fired.
7. Thank you; have a nice day!

The Bezos Mandate

1. すべてのチームは今後、サービス・インターフェイスを通じてデータや機能を公開することになる。
2. チーム同士は、これらのインターフェイスを通じてコミュニケーションを取らなければならない。

ground up to be externalizable. That is to say, the team must plan and design to be able to expose the interface to developers in the outside world. No exceptions.

6. Anyone who doesn't do this will be fired.
7. Thank you; have a nice day!

失敗談

人間は機械ではない

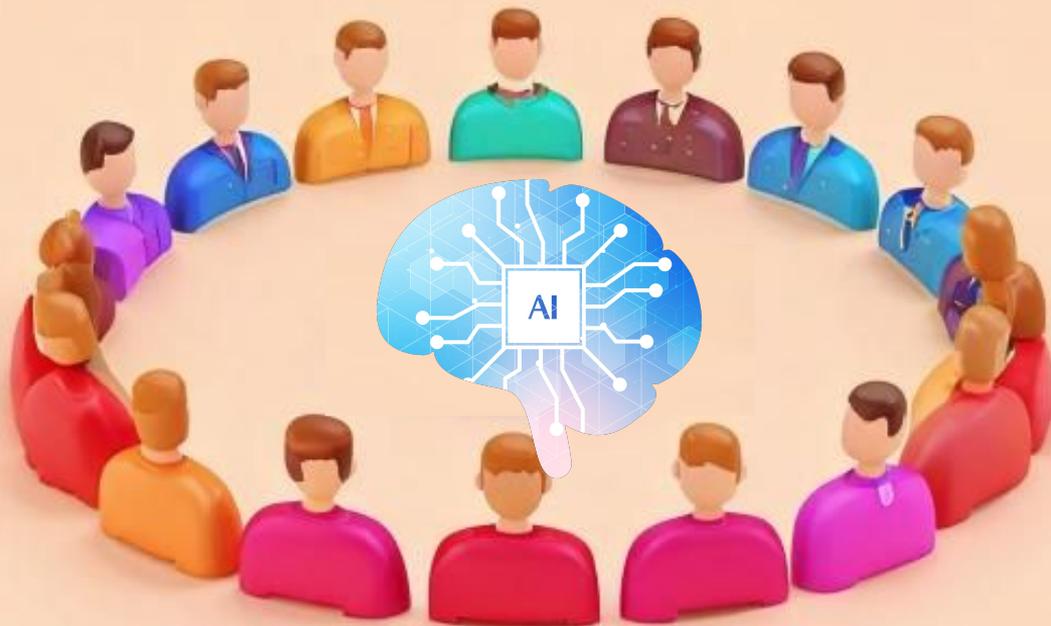


チャットにて

自分「きちんと相手に伝わっているか、わからないから、
伝わっていたら、Ack、伝わらなかったら、Nackって返して」

相手「バカにすんな」

AIによって人間のコミュニケーションを促進する



ChatOps

言う方も言われる方も嫌なことはボットに言わせる

- 提出物の催促
- 勤怠管理
- 静的解析結果の通知

すでに使われているこのボットの用途を生成AIで拡充する

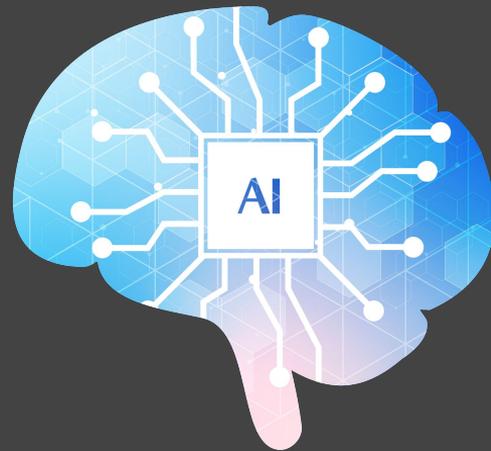
生成AIの活用

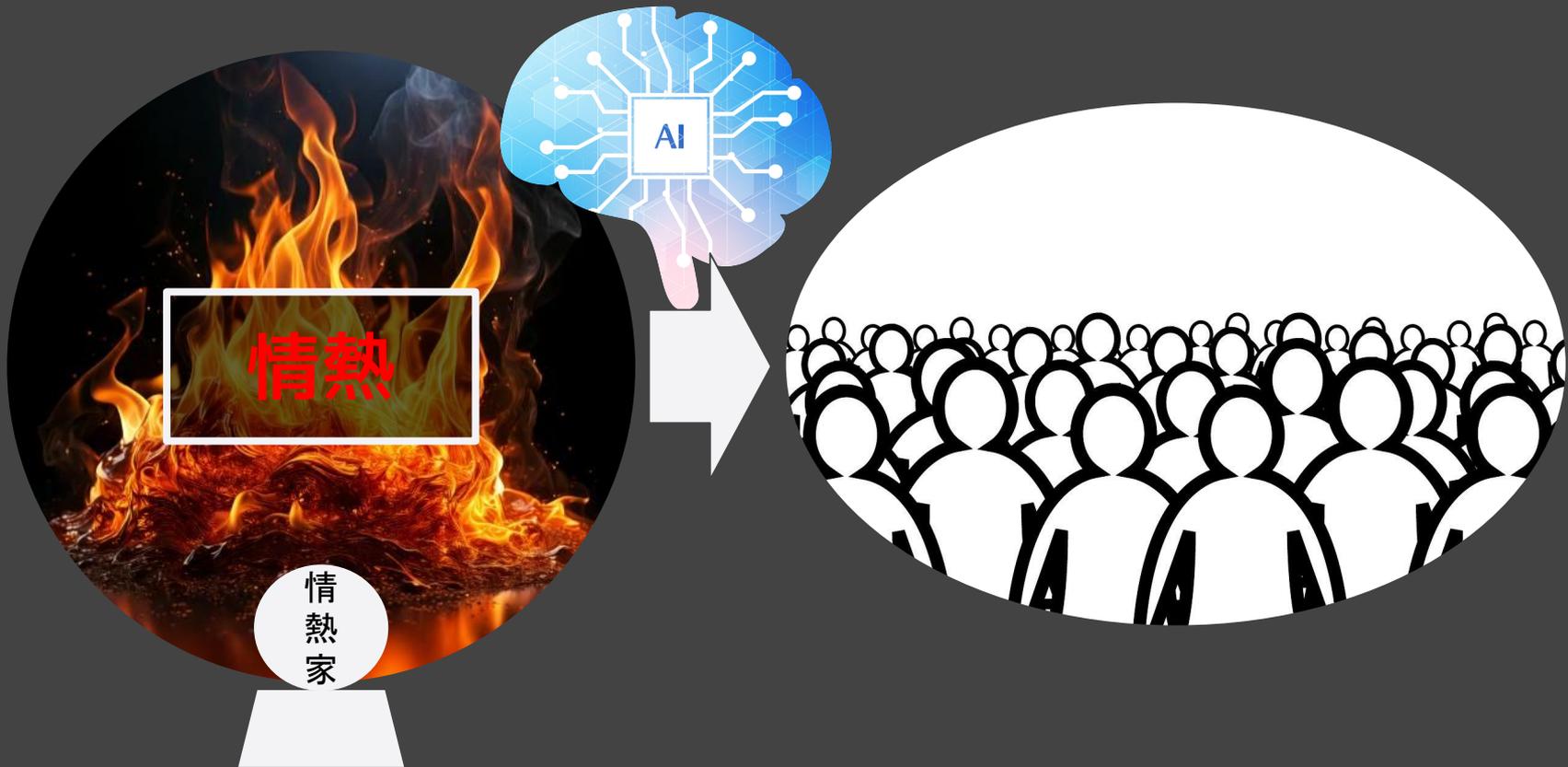


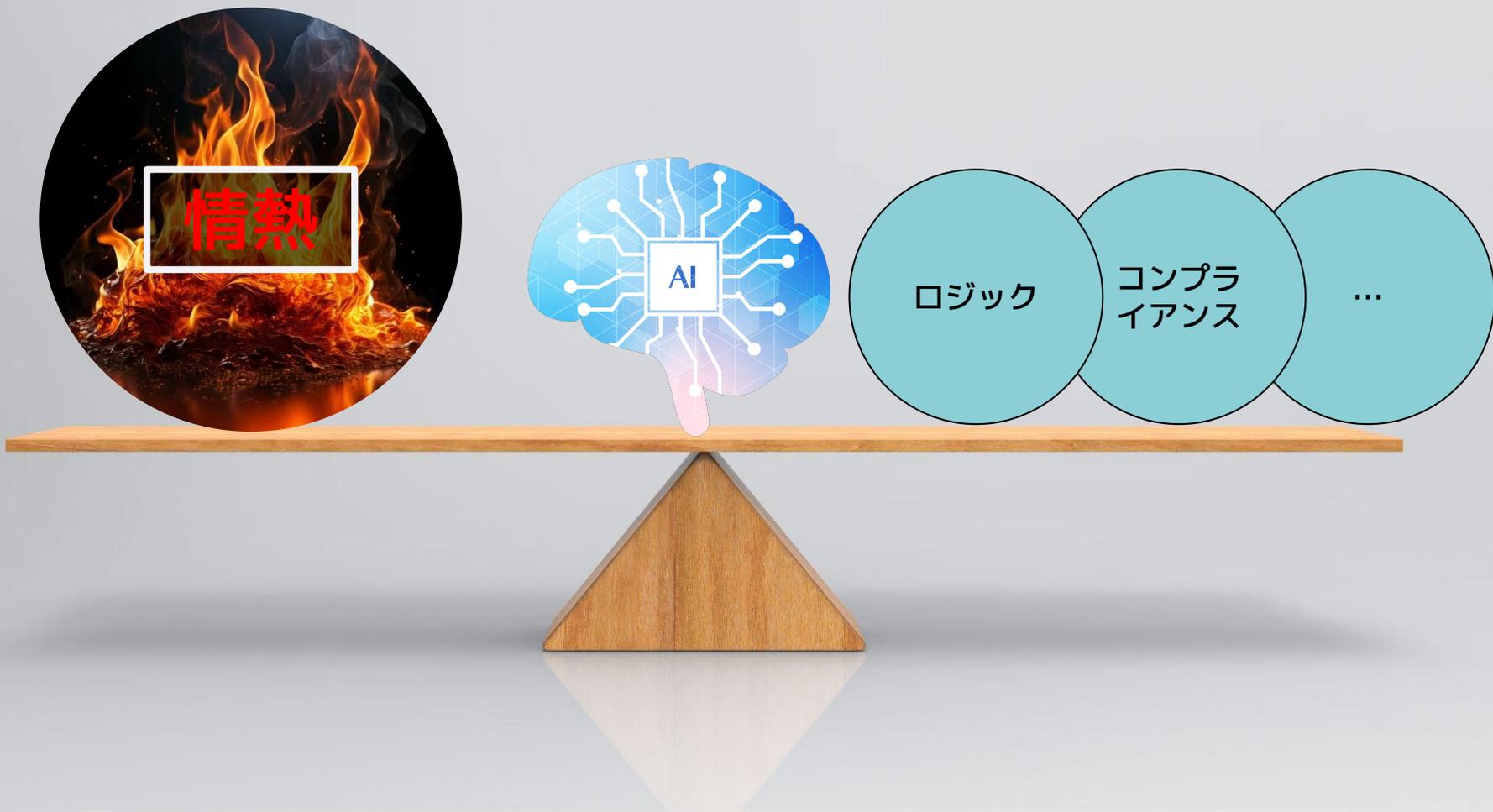
- 自分自身のメンターとして活用
- 上司・起案者に対する意見の橋渡しとして活用

- チームに対する外部知見者として活用
- ファシリテーター・モデレーターとして活用









情熱

AI

ロジック

コンプライアンス

...

A large, messy pile of dry sticks and branches is stacked on a dark metal tray in a grassy field. The sticks are of various lengths and thicknesses, some with small twigs protruding. The background shows a green field, some bushes, and distant hills under a clear sky. The lighting suggests it's either early morning or late afternoon.

情熱がない

情熱に火を付ける



情熱とは本質に向き合うこと

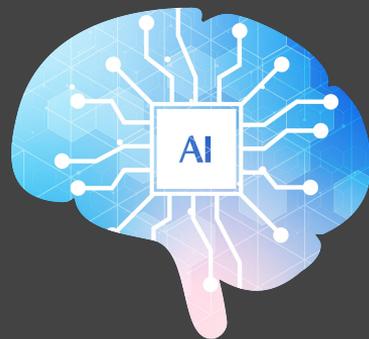
- 情熱 = エネルギーの元を注入する
 - 情報収集
 - 理解
 - 共感

情熱に火を付ける

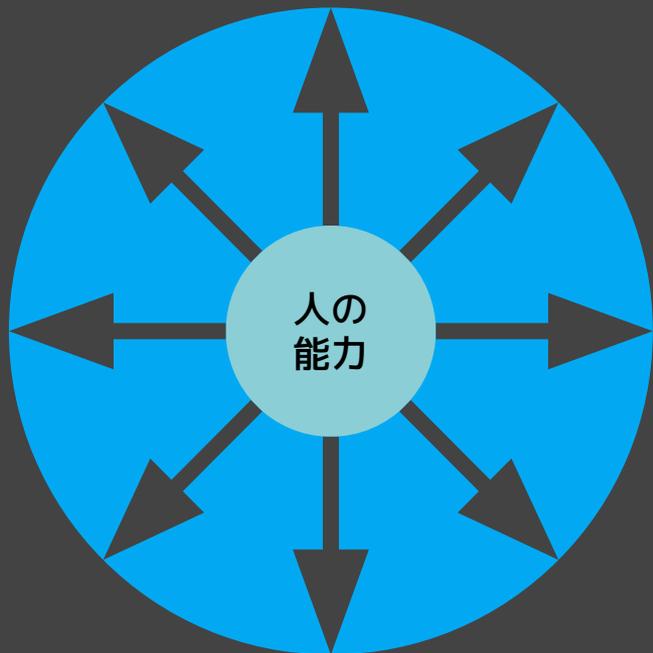


情熱とは本質に向き合うこと

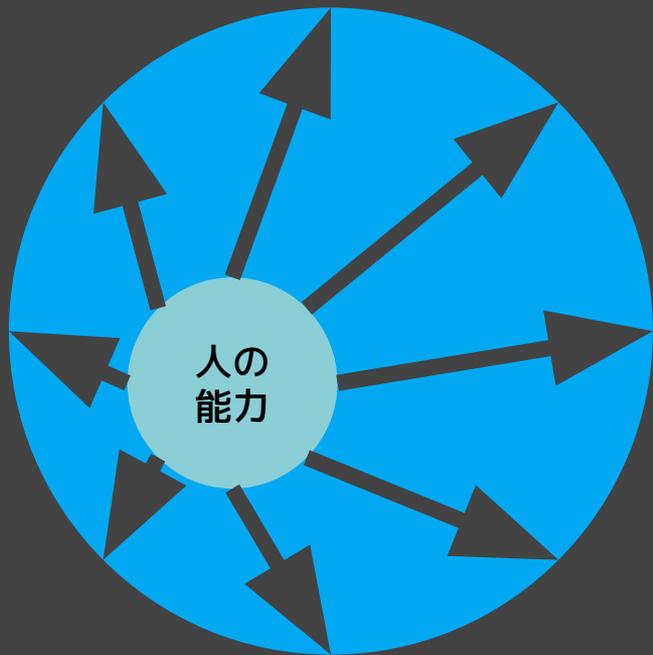
- 情熱 = エネルギーの元を注入する
 - 情報収集
 - 理解
 - 共感



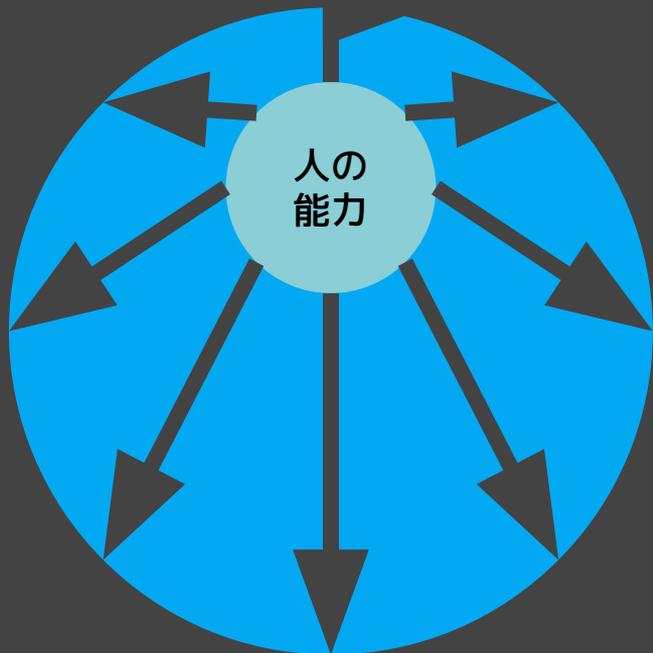
AIは人の能力を拡張するもの

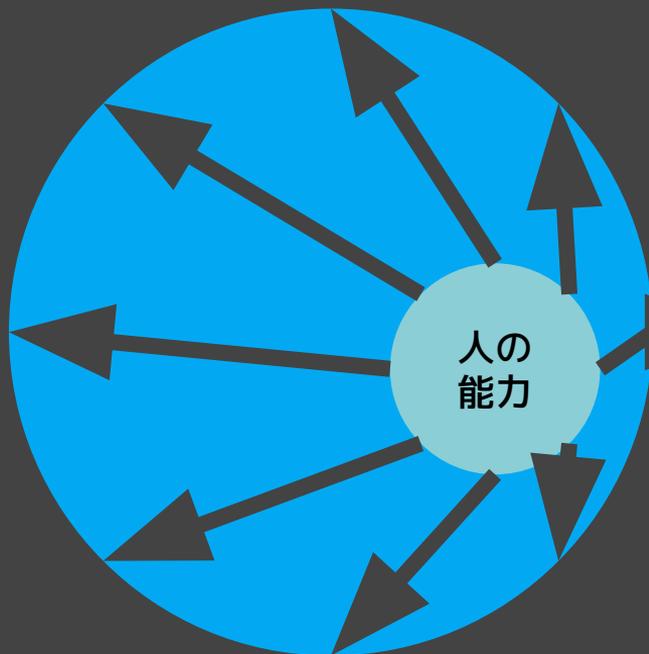


AIは人の能力を拡張するもの

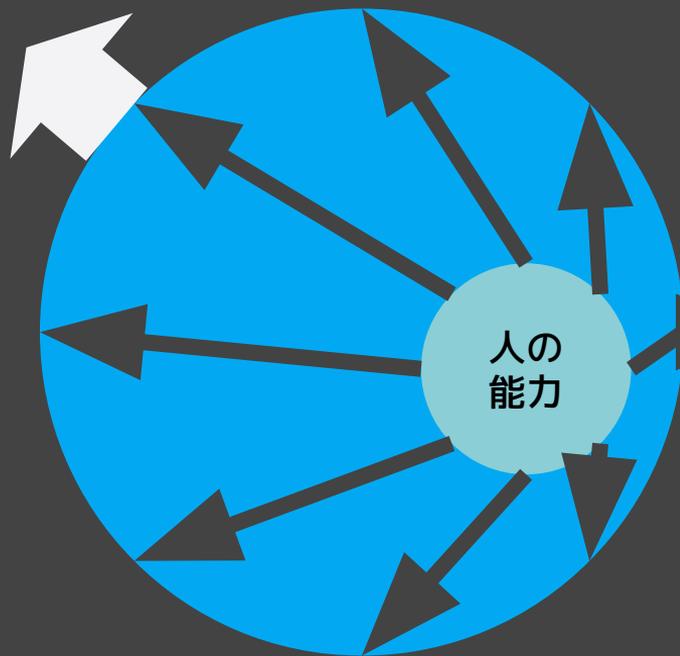


AIは人の能力を拡張するもの





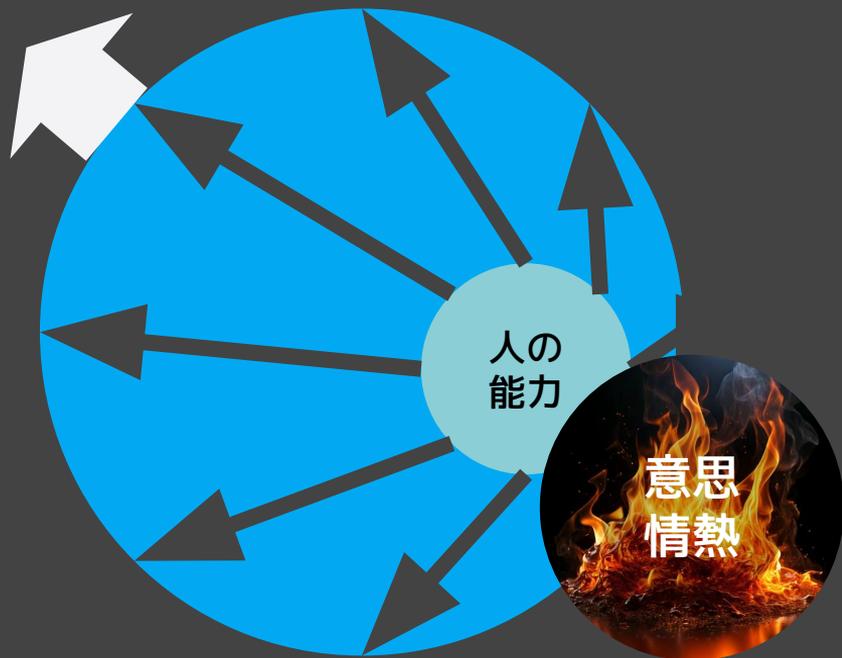
AIは人の能力を拡張するもの
どんな方向に拡張するか



AIは人の能力を拡張するもの

どんな方向に拡張するか

どんな方向に進もうとしているのか



AIは人の能力を拡張するもの

どんな方向に拡張するか

どんな方向に進もうとしているのか

人間の意思・情熱が必須

情熱を持って開発を進めましょう

情熱が人間の行動のエネルギー源。情熱を持って進めましょう

情熱を持つためにも生成AIを使いましょう

情熱をうまくコントロールするためにも生成AIが使えます

人間のコミュニケーションを円滑にするのに生成AIが使えます

A sunset landscape with a large, bright yellow sun in the sky. The scene is bathed in warm orange and red light. In the foreground, a dirt path leads towards the horizon, where the lower legs and feet of a person are visible. The background features rolling hills and mountains under a cloudy sky. The word "END" is overlaid in large, white, bold, sans-serif capital letters across the center of the image.

END