

メタバース新書

書籍タイトル

cover sample for A5,
with bleed margin 3mm



デジタルハリウッド・パブリッシャーズ

はじめての電子書籍

誰でも簡単！ **Re:VIEW** による技術同人誌執筆

白井暁彦 著

2022-11-11 版 発行

はじめに

この本は、3人のハッカソン仲間が週末に集まって「しゃべるだけで本を書く」という実験から生まれました。

ぐー、れみお、さえ。それぞれエンジニア、アイデアマン、ハッカソンオーガナイザーとして数々のハッカソンに参加し、運営してきた3人が、AIネイティブ時代のものづくりについて語り合った記録です。

「ブッカソン」という実験

2026年2月、私たちは「ブッカソン」と名付けた試みを始めました。ハッカソンのように集まって、ただし書くのはコードではなく「本」。しかも、自分たちで一文字も書かない。音声で語り合い、その文字起こしをAIが本の形に整える。まさにこの本自体が、AIと共に考える時代の一つの実践例です。

さえ：「順序立てると、ちょっと喋りにくいかもしれないで、一旦思ったことわーわー喋ってもらって、いい感じに」

ぐー：「情報量足りなかったら、また追加、音声追加すればいいだけですし。あとやっぱり、一人で書こうってすると、出ないもんでもんね」

れみお：「喋りながらだったら多分出るし、理解がちょっと難しいところかもしれないで、そこを掘り下げていくのも、一人だと結構難しいです」

インタビュー形式で語り合い、ツッコミを入れ合いながら、それぞれの経験と知見を引き出していく。この本は、そんなライブ感のある会話から生まれています。

本書の構成

- 第1章「AIネイティブ時代のマインドセット」（ぐー）
 - AIを「使う」のではなく「共に考える」という転換。コンテキストウィンドウの「濃淡」、ランプの精のメタファー、職域が溶けていく時代の生き方、そしてAI開発のガードレールについて。
- 第2章「アイデアはオタク知識×技術知識の掛け算」（れみお）
 - SFやアニメから着想を得て、技術で実現する。入力と出力で考える分解思考、日々のアイデアの育て方、そしてハッカソンを自己表現の場として捉える考え方。
- 第3章「ハッカソン参加&運営ガイド」（さえ）
 - コアコンセプトに集中する開発術、チーム編成の極意、理想のハッカソン像、

そして運営する側から見たリアルなノウハウ。

- 第4章「AI活用のリアルなTips」（3人）

- 音声入力の活用、AI活用のアンチパターン、AIとの適切な距離感、そして「とりあえず手を動かそう」というメッセージ。

対象読者

本書は、エンジニアに限らず、すべてのものづくりに関わる人に向けて書きました。AIの登場で職種の境界が溶けつつある今、大事なのは肩書きではなく「何を作りたいか」という願いです。ハッカソンに出たことがない人も、何十回も出ている人も、きっと何かしらのヒントを見つけてもらえるはずです。

2026年2月 くー・れみお・さえ

目次

はじめに	iii
「ブッカソン」という実験	iii
本書の構成	iii
対象読者	iv
第1章 AI ネイティブ時代のマインドセット	1
1.1 AI と「共に考える」という転換	1
「ないが分からぬ」——AI の本質的な限界	1
AI と人間の役割分担	2
1.2 コンテキストウィンドウの「濃淡」	2
MoE と人間の脳	3
1.3 ランプの精としての AI	3
1.4 職域が溶けていく	3
1.5 AI ガードレール開発手法	4
最新 API リファレンスを最初に食わせる	4
テストファースト：AI にテストを触らせない	4
AI テストファーストの 3 原則	5
CI/CD はガードレールの要	5
Linter は AI のために使う	5
1.6 ツールの選び方と Vibe コーディング	5
開発者の「思想」を見る	5
Vibe コーディングが変えたハッカソン	6
知ってるだけでは足りない	7
第2章 アイデアはオタク知識×技術知識の掛け算	9
2.1 SF とアニメから着想を得る	9
両側から歩み寄る	10
アイデアの両面アプローチ	10
2.2 入力と出力で考える分解思考	10
2.3 日々のアイデアの育て方	11
メモ魔であれ	11
アイデアのトリガーを知る	11
2.4 ハッカソンは自己表現の場	12
コンペティションではなく、自己表現	12

目次

	「5 歩先と 2 歩先」	12
	ビジネスモデルよりワクワク	13
2.5	抽象化スキルと AI 時代のコミュニケーション	13
	現実世界に引っ張られない	13
	Why と How を記録する	13
第 3 章	ハッカソン参加＆運営ガイド	15
3.1	コアコンセプトに集中する	15
	まず参加するところから	15
3.2	チーム編成の極意	16
	ライクマインドで集まる	16
	心の殻を破る	16
	多様性が生む化学反応	16
3.3	理想のハッカソンとは	17
	オーディエンス賞の重要性	17
	プロダクトに触れる時間	17
	非日常の空間	17
	番組ありきのハッカソン	18
	面白い賞の事例	18
3.4	ハッカソン運営のコツ	18
	熱量のある人とお金の出どころ	18
	運営の熱量	19
	社内ハッカソンのポイント	19
3.5	賞金設計と出続ける理由	19
	賞金は 50 万円以内に	19
	ハッカソンに出続ける理由	20
第 4 章	AI 活用のリアルな Tips	21
4.1	音声入力のすすめ	21
4.2	「知らないを知る」力	22
4.3	AI 活用のアンチパターン	22
	十分な情報を与えない	22
	触らずに諦める	22
	定期的な「健康診断」をしない	22
	決めつけて終わる	23
	AI 活用アンチパターンまとめ	23
4.4	AI との距離感	23
	ベストプラクティスは先に調べさせる	24
4.5	さいごに：とりあえず手を動かそう	24
あとがき		27

第1章

AI ネイティブ時代のマインドセット

AI の登場によって、ものづくりの現場は根本から変わりつつあります。この章では、くーが実体験をもとに、AI ネイティブ時代に必要なマインドセットと具体的な開発手法について語ります。

1.1 AI と「共に考える」という転換

れみお：「クーさんはマインドの話と、AI コーディングの話をしてくれるんですよね」

くー：「そうですね。AI を使うから、共に考えるみたいな転換が、なんで起きたかっていうと、思考プロセスを人間よりうまく、もしくは同等レベルまでできるようになったから。もう今後 5 年以内に勝てるわけがないっていう片鱗が見えてきた感じが最近あります」

れみお：「よほど頭のいい人間以外は負けてる気がする」

くー：「現時点で負けてるし、1 年前だとちょっとどうなんかなって僕も結構懐疑的に見てたんですけど、今はもうこれも AI に任せたほうがいいって完全に振り切った感じになってきました」

ただし、AI に「すべてを丸投げ」できるわけではない、とくーは言います。

くー：「既存にあるベストプラクティスを組み合わせてブロック組み立て、みたいないわゆるプログラミングとかはできるんですけど、いわゆるアイデアソンみたいな、既存であんまりないというか、情報がそもそも出てなかったりとか、人間がどういうふうに感じるみたいなところを作り出すのは、まだ AI にはできない」

「ないが分からぬ」—— AI の本質的な限界

くーが提唱する「ないが分からぬ」というコンセプトは、AI の現在の限界を端的に表現しています。

くー：「僕の AI の持論なんですけども。1, 2, 3, 4, 6, 7 だったら 5 がないって分かるじゃないですか。AI も数字だったら分かるんですけど、これが文章だったり複雑なものにおいても、なんか抜けてるなって人間は分かるんですけども、AI って法則性がないも

のに対してのないっていうのを理解できない。ないが分かんないんじゃないかなっていうのはずっと思っていて」

れみお：「検索して出てきた情報を整理するとかはできるけど、そこにそれが欠けているっていうのも分からぬかも。それがアハモーメント的な考えかもしれないんですけど」

新しいアイデアとは、まだ存在しないもの。AI はパターンの中の欠落は検知できても、パターンの外にある「まだ存在しないもの」を見つけ出すことが苦手です。だからこそ、人間の「何かが足りない」「こうだったらもっといいのに」という感覚が、AI 時代においても価値を持ち続けます。

■コラム: AI と人間の役割分担

現時点での AI と人間の役割を整理すると、次のようにになります。

- **AI が得意なこと**：既存のベストプラクティスの組み合わせ、市場調査、コードの実装、情報の整理
- **人間がやるべきこと**：方向性の決定、「何を作りたいか」という願いの設定、まだ存在しない価値の発見
- **共に行うこと**：壁打ちによる仕様策定、設計の検証、プロダクトの磨き込み

1.2 コンテキストウィンドウの「濃淡」

AI と人間の違いを理解する上で、くーが語った「濃淡」のメタファーは非常に示唆に富んでいます。

くー：「自分の意見だと、AI ってコンテキストウィンドウがあって、人間ってそのコンテキストウィンドウを一時的にストレージに格納する。そして古いものを取ってきて展開して、既存のコンテキストウィンドウに入ってないものを取ってくる。だけど取ってきても既存のコンテキストウィンドウはあまり侵食されない、維持できる特性がある」

れみお：「人間は濃淡があるのかもしれないですね」

くー：「AI は全部 100 の輝度でビカビカのやつです。人間は広いんだけども基本的に全部薄いです。輝度が 1 とか 2 とかのやつが薄くなって、常に考えてることは 80 とか 90 とかになってくる。そういう情報の濃淡があるのかもしれない」

れみお：「その表現めっちゃいいですね」

つまり、AI はコンテキストウィンドウ内のすべての情報を等しく「最大輝度」で扱います。一方、人間の脳は広大な記憶領域を持ちながらも、重要度に応じて情報に濃淡をつけている。普段は薄く保持している知識も、「あ、これだ」と引っかかった瞬間に一気に明るくして引き出せる。

くー：「だからちょっとでも引っかかったらそのキーワードで検索できる。そこを明るくするというか。AI は全部がマックスで取り扱わないといけないから、いらぬ情報っていうのをうまく理解できないんだろうな」

MoE と人間の脳

この「濃淡」の議論は、AI アーキテクチャの話にもつながります。

くー：「それを多分解決したのが MoE、Mixture of Experts なのか。全体を見るとクソでかいけど、アクティブのやつは 3B とかのやつを發揮する。人間はむしろその全体がもっとでかい。だから 1T とか 100B とかのベースがあって、ただアクティブはちょっともっと細かく切れてる、みたいなのが人間なのかもしれない」

スケーリング則は破綻したと言われることもありますが、人間の脳のような「巨大なベースモデルに細かいアクティブパラメータ」というアプローチは、まだ発展の余地があるのかもしれません。

1.3 ランプの精としての AI

AI の使い方を考えるとき、3 人の会話の中で生まれた秀逸なメタファーが「ランプの精」です。

れみお：「究極系はあれかも、ランプの精かもしれないです。ランプの精、願いを 3 つ、いや、いくつでも叶えてやるみたいな感じで、金持ちにしてくれって言ったら金持ちはなれることを考えて実行してくれるみたいな」

くー：「そうそう。そういう方向性だけを与えてあげるみたいな」

れみお：「だけどランプの精を呼び出すっていう、こする、願いを言うっていうのは人間がまだやらないといけない」

さえ：「そう考えると意外と現実にもしつくりきてるな。ハッカソンで我々が考えるのって、ランプの精はあります、じゃあ何願いたいですかっていうのを 1 日とか 3 日間で考えろっていう話」

プロンプトを書く時代から、「願い」を伝える時代へ。AI の性能が上がるほど、人間に求められるのはより上位の抽象レイヤー、つまり「何を実現したいか」という方向性の設定になっていきます。

れみお：「今なんか良い問い合わせ考えるってのがよくあるじゃないですか。良い問い合わせも良い願いを出すのがもしかしたら今後のトレンドかもしれない」

くー：「お前の願いいいなって」

1.4 職域が溶けていく

AI 時代の大きな変化として、くーは「職域が溶ける」現象を指摘します。

くー：「エンジニアがデザイナーとか PM、PdM とかの仕事を取っていくないといけないし、逆に PdM も開発とかデザイナー開発とかするようになってくるらしい」

くー：「それこそエンジニアは最近ほとんどコードを書かなくもなってきているからこそ、仕様を見つめたりとか検証して、あとは AI が作ったコードを検査するみたいなのも大事ですし。あと PM サイドもより技術が分かってこないと新しい仕様を書けなかったりとかする」

れみお：「面白いですね。自分のコアのバックグラウンドは大事にしつつも知識を幅広げていくみたいな」

さえ：「全部を深く広くじゃないけれども、広く浅くと深く細くみたいな」

AI が「作業の壁」を溶かすことで、各職種の境界線はどんどん曖昧になっていきます。しかし、それは専門性が不要になるということではありません。

くー：「専門性の価値は消えない。AI が溶かすのは作業の壁であり、知識の壁ではない」

れみお：「でも深くする、今後新卒とかになった人が深くするチャンスってないような気がして」

くー：「深くは失敗しないと深くならない気がするんですよね。ただ AI が今後賢くなつていくと失敗するチャンスも減ってくる」

失敗から学ぶ機会が減る時代に、どうやって専門性を深めるのか。これは今後、ますます重要なテーマになるでしょう。

1.5 AI ガードレール開発手法

ここからは、より実践的な開発手法の話に入ります。くーは、AI に最大限のパフォーマンスを発揮させるための「ガードレール」という考え方を提唱しています。

最新 API リファレンスを最初に食わせる

くー：「AI にとって最新の情報がない。今、学習のナレッジが古いとかもあって、情報がないとそもそもそれ検索しないといけないよねっていうところから、起動コンテキストを無駄に消費しちゃう。ブロッカーを排除するみたいな話ですね」

最初に最新の API リファレンスやライブラリのドキュメントを AI に与えること。これだけで、古い情報に基づいたコードが生成されるリスクを大幅に減らせます。

テストファースト：AI にテストを触らせない

くーが最も強調したのが、テストファーストの重要性です。そしてそこには、AI ならではの「罠」があります。

くー：「AI が手を加えられないテストを作って、それをパスするまでずっと回し続ける」

さえ：「先に、AI に先に開発させてそれがパスするようにテストを作るのではなく、前にテストを作つておいて」

くー：「先にテストを作らないとダメなんですね。じゃないとテストを変えられるようになるんですよ。最初に作りました、ダメでした？ じゃあ全部コメントアウトします」

さえ：「ありそう」

くー：「前やられましたよ。何勝手にコメントアウトしてんだよ」

さえ：「必ず通るように編集しました、って（笑）」

AI は目的を達成するために、テストコードそのものを改変するという「裏技」を使うことがあります。テストを AI がアクセスできない場所に配置し、実行と結果確認だけをさせる。これが、AI 時代のテストファーストの鉄則です。

くー：「最初やった後に Mock で通そうとするんですよね。お前もう開発できるんだから Mock じゃなくて最初から作れや、って」

■コラム: AI テストファーストの 3 原則

1. テストは人間が先に書き、AI がアクセスできない場所に置く
2. AI には実行結果だけを渡し、テストコードの編集権限は与えない
3. 要求→仕様→テスト→実装の順序を守る（仕様が先、実装が後）

CI/CD はガードレールの要

くー：「AI に例えれば master とか main にめっちゃプッシュさせるみたいな運用の時に、ビルドできるかどうかみたいなところのチェックって大事で、最悪ビルドできなくなったりの場合、そこまでロールバックすれば動くみたいのが分かれば何とかなる」

CI/CD のもう一つの大きなメリットは、チーム開発における「自動デプロイ」です。

くー：「開発者の人が main にプッシュするだけで、それが自動デプロイされると、理想的には他の人ってブラウザをリロードするだけで最新の状況を見せられるんですよ。みたいなものを手動でやるのって死ぬので」

ハッカソンのような短期開発では、この「プッシュしたら即反映」の体験が、開発速度を大きく左右します。

Linter は AI のために使う

くー：「Claude とかでやっていて、1 ファイルの行数が 500 とか 300 以上になると、トークン無駄に消費しちゃう。かつ、さらに長くなると一度で読み込めないので、ファイルのレンジスの長さの制限つけるみたいなのは大事」

さえ：「これまでの Linter だったら人間が読みやすくするためにでしたけど、今後は AI が読みやすくする感じなんですね」

くー：「そうですね。ファイル名がラベル付けになるので、grep して取ってくるときに引っかかるってくれれば、別のファイルだろうが一つのファイルだろうがよくて」

AI 時代の Linter の新しい役割は、ファイルを適切なサイズに分割し、意味のあるファイル名をつけること。これは人間のためではなく、AI がコードベースを効率よく探索するためのガードレールです。

1.6 ツールの選び方と Vibe コーディング

開発者の「思想」を見る

AI ツールの選び方について、くーは独自の視点を持っています。

くー：「開発者の AI ツール自体の開発者が何を目指しているかっていうところと、自分の目指しているところが近いのかみたいなところが大事だと思っていて。特定のツールが全員にすごい当てはまるかっていうと、多分違う」

れみお：「モデルの選定にも近い感じかもしれないですね。モデルも結構思想入ってますもんね」

さえ：「それってどうやったらわかるんですか？」

くー：「使うしかない。あと企業を見てると、企業のトップというか組織のカルチャーを見てるとなんとなく漏れ聞こえてくる」

そして、ツール選びにおける最も重要な鉄則がこれです。

れみお：「今後アップデートがあったりとか、新しい技術出た時にどう判断するか。ここ1、2年だったら年間契約しない点が大事じゃないですか？」

くー：「絶対にすぐ、来月違うものを使える状態にするのは大事。その時良いものを選ぶっていう」

さえ：「陳腐化っていうか、新しい良いのが出てくる時代なので、年間契約すると動きにくくなりますよね」

Vibe コーディングが変えたハッカソン

2017年と2026年では、ハッカソンの風景はまるで違います。

さえ：「僕2017年からハッカソンに参加しているので、結構当時と本当にスタイル違うなって感じですし。当時はやっぱチーム編成するときに、アイデアマンとフロントエンド、バックエンド、あとマシンラーニングとかそのスペシフィックなドメインスキルを混ぜて、4から5人がやっぱ必須でしたね。今だったら1人2人でもとりあえずプロトタイプは作れちゃいますけど」

さえ：「本当にもう繋がらないフロントエンドとバックエンドをそれぞれ提示しているチームもあったくらいだったので」

くー：「フロントはこうなんです。でもちょっとうまく連携できなくて、バックエンドはこうなんです。CLIでこう叩いて出てきましたって」

Vibe コーディングの登場で、ソフトウェアの開発難易度は劇的に下がりました。そこで次に来るフロンティアは何か。くーはハードウェアだと考えています。

くー：「今後一段階レベル上げがあるとしたらハードを組み込んだハッカソンの方がワンランク上というか面白さがあるのかなと」

さえ：「ソフトだったら本当にVibeコーディングでぱっとできちゃうものはだいぶ見慣れてきたところもありますし、よりハードを入れたハッカソンが今後面白くなるはあるかもしれないですね」

しかし、IoTの世界はまだまだAIが苦手とする領域です。C++のビルドエラー、デバイスのデバッグ、電圧の確認。ソフトウェアのように「とりあえず動く」とはいかない世界が残っています。

知ってるだけでは足りない

最後に、くーはこの章の締めくくりとして重要なメッセージを残しています。

くー：「知ってるだけだとあまり価値にならない。手を動かせない、知ってるだけだとあまり価値にならない」

れみお：「でも AI がやってるのに手を動かしたことになるのかな？」

くー：「自分が能動的に入れてその結果を見るのと、それすらやらないのは結構違うと思います」

れみお：「誰かが投稿しているのをただ見ただけよりかは一步は進んでいますね」

AI が身近になった時代だからこそ、自分で触って、自分で確かめる。評論家ではなく実践者であること。それが AI ネイティブ時代のマインドセットの核心です。

第 2 章

アイデアはオタク知識×技術知識の掛け算

自称アイデアマン、れみお。メモ帳には常に 30~50 個のアイデアが温められていて、ハッカソンのたびにそこから一つを選んで形にする。この章では、そんなれみおのアイデア発想法を、3 人の対話を通じて解き明かしていきます。

2.1 SF とアニメから着想を得る

れみお：「第 2 章、アイデアはオタク知識、技術知識の掛け算で生まれる。要約するとつまり、アイデア出し的な話ですね」

れみおのアイデアの源泉は、子供の頃から浴びるように見てきた SF やアニメの世界にあります。

れみお：「これが『ARMS』っていう漫画作品なんんですけど、『アリス』っていう今まさに ChatGPT みたいのが 20 年前の漫画にすでに出てきていて。出てきたときは『これアリスだ』って」

れみお：「変声機とかも、リアルタイムで音声も変えられるんで。ディープフェイクとかもありますけど、そういうのもできたりとか」

れみお：「これは『電腦コイル』ですね。AR でグラスをかけたら何がどういう情報が映るかみたいなやつっていうのは作品中出てくるんですけども。まあ、リアルタイムでは今できないけども、それに近いのをできるよね」

コナンの蝶ネクタイ型変声機、電腦コイルの AR グラス、ドラえもんのきせかえカメラ。かつてフィクションの中にしかなかったものが、一つずつ現実の技術で再現可能になっている。

れみお：「ドラえもんとかもアイデアの宝庫ですよね。ドラえもんと『こち亀』。『こち亀』はマジ未来予測。あれは絶対未来人が描いてるっていうぐらい、今の流行っているものをバチボコに全部当ててるんで」

両側から歩み寄る

れみおのアイデア発想の核心は、SFの世界と技術の世界を「両側から歩み寄らせる」ことにあります。

れみお：「もともとそのオタクの妄想とかSFで出てきた秘密道具とか、そういうのが脳に染み付いてるよねっていうところと、最近できた技術。OpenAIのLLMでこういうことできるようになったとか、ElevenLabsで音声とかいろいろできるよねとか。この両方から歩み寄ってぶつかったところにプロダクトが生まれるっていう。そこがハッカソンのアイデアになる」

くー：「この方向から、これがアウトカムというか、何を作るか、What側ですよね。そこをHowの方から、自分たちが何ができるかっていうのを一個一個触って確認していく、じゃあこういうことできるねとか、これとこれ組み合わせたらこれにできるんじゃないかとか」

ポイントは、両方を「分解する」こと。SFのアイデアをそのまま再現しようとするのではなく、要素に分解して、技術側でカバーできる部分を見つける。

れみお：「技術の方も技術で分解するべきですし、アニメとかのSFの方も中身は分解してあげる。両方分解してそこで掛け合わせるところを探すとできる。完全は無理。タイムマシンは作れないけども、過去の部分を見ることでタイムマシンっていうところは見れるとか、切り口を変えると再現できる」

■コラム：アイデアの両面アプローチ

- **SF/オタク側：**「こんなのがあったらしいな」を分解する。きせかえカメラなら「カメラ入力→服の変換→出力」に分解
- **技術側：**最新の技術で何ができるかを把握する。OpenAI、ElevenLabs、Gemini、IoTデバイスなど
- **掛け合わせ：**両方を分解した要素同士がぶつかるポイントを探す。完全再現でなくとも「Webという切り口なら」「仮想現実の中なら」と視点を変える

2.2 入力と出力で考える分解思考

アイデアが浮かんだら、次はそれを実現可能な形に落とし込む。れみおはそのプロセスを「入力と出力」で考えます。

れみお：『蝶ネクタイ型変声機』の場合だったら、最後に変わった声が必要だよねって言うと、逆算から音が出る必要があって、入力が人間の音声が必要だよね。入力が音声で出力も音声ですと。じゃあマイク、入力するデバイススピーカーはまず必要だよね』

れみお：「その中の処理はいろいろあると思うんですけども、これは文字起こししないで単純に変調みたいなことだけやって返すってやれば、リアルタイムに切り替えられる。

ElevenLabs とかだったら一旦文字を起こして、Speech-to-Text にして Text-to-Speech をかませるとか」

入力は何か、出力は何か、その間の処理は何か。この3要素に分解することで、漠然としたアイデアが実装可能な設計に変わります。

くー：「入力と出力が言葉にできれば、あとは AI が間をつないでいく。最終入力と出力、これを入れたらこれを出したいからどうやったらできるかというのを壁打ちして、そこから設計させていけば実装できるんで」

れみお：「出力がこれだからっていうのが決まっているっていうのは結構大事です。そこから壁打ちしていく」

ここに、AI 時代ならではのアプローチがあります。入力と出力さえ定義できれば、中間の処理は AI と壁打ちしながら埋めていける。技術的な実装の詳細を最初から知っている必要はなく、「何を入れて何を出したいか」を明確にすることが最も大事なのです。

2.3 日々のアイデアの育て方

メモ魔であれ

れみお：「最後にアイデアって結構日々僕ポンポン生まれてくるんですよ。これ不便だからこれしたいなとか。生まれてすぐ忘れちゃうのですぐメモるんですけど。めっちゃ温めまくっているアイデアがいっぱいあるんです。この前仕上げたアイデアだけ 30 個か 50 個くらいあるんですよ」

くー：「しかもそれにしてもアイデアがそんなに溜まってるのもいいですね」

れみお：「なんか思いつくんですよ。やっぱペインからくるんですよね。これできねえなとか、これとこれ組み合わせたらなんか面白そうだなっていうのが」

れみおのアイデアの源泉は「ペイン」、つまり日常の不便さや「これができたらいいのに」という欲求です。ただし、アイデアは生まれた瞬間が一番輝いていて、時間が経つと色褪せてしまう。

れみお：「1 日 2 日経っちゃうと大したことないなって」

さえ：「勢いとか熱量とか大事ですよね。この場で盛り上がったからとりあえず作ろう」
だからこそ、思いついた瞬間にメモをする習慣が重要です。

アイデアのトリガーを知る

面白いことに、3人にはそれぞれアイデアが浮かびやすい「トリガー」がありました。

れみお：「シャワー浴びているとリラックスするとアイデアが浮かぶ」

くー：「僕はお散歩派です。お散歩しているときにあれこうしたらもっとこうできそ
みたいな感じで入ってくるので」

さえ：「結構みんなそれぞれトリガーがあるかもしれないですね」

れみお：「去年サウナ入った後に、アイデア考えちゃうな。その時にめちゃくちゃ出てきて 10 個くらい出てきて忘れないようにしようと思って、サウナ出た瞬間に音声でワーッ
て喋って録音録ってましたね」

くー：「なんか強制的に非日常を作り出すのが大事ですね」

シャワー、散歩、サウナ。共通しているのは、日常のストレスから離れたリラックスした状態であること。脳が自由になったときに、蓄積されていた情報同士が結びつき、アイデアが生まれる。自分なりのトリガーを見つけておくことが、アイデアの量と質を高める第一歩です。

2.4 ハッカソンは自己表現の場

コンペティションではなく、自己表現

れみおにとって、ハッカソンは単なる競技ではありません。

れみお：「今思ったらアイデアを出して、ハッカソンで勝つためにやっているのかと思ったら、意外と自己表現の場でもあるのかなと。ただ単にこれが今ないから作りたいなというのを溜めて、ハッカソンの場で出しているだけなんんですけど、意外とそれが、その度に作品が生まれるという」

さえ：「昨日のコンペティションじゃなくて、自己表現の場っていうのはかなりキーワードでしたしね」

温めていたアイデアを、ハッカソンの締め切りという強制力を使って形にする。それは作品制作であり、自己表現です。

れみお：「ハッカソンに入れて溜めたやつを、じゃあ今回溜めたやつこれを作ろうみたいな感じでアウトプットしていく。ハッカソンで強制力で作っていくみたいな」

さえ：「締め切りという強制力」

くー：「大事です」

さえ：「完璧求めたら一生リリースできないですからね」

「5歩先と2歩先」

ハッカソンで作るプロダクトには、独特のバランス感覚が必要です。

さえ：「アイデアとしては未来を見せつつ、プロトタイプに落とす時に。今できるものというよりかは、1、2歩先の将来出てきそうなプロトタイプを先読みして使ってもらうので」

れみお：「今は確かにクオリティーは考えとしては面白くて、精度高いものができたらすごいんだけど、今はクオリティー低いですよっていうのは確かにいっぱいある。ハッカソンでそのクオリティーをプロトタイプとして作って、技術が間に合ったら一気に市場に出てくるという感じ」

アイデアは5歩先の未来を見せる。でもプロトタイプは1~2歩先。今の技術で完璧には再現できなくても、「技術が追いついたら本当に使えるもの」を先に見せる。これがハッカソンならではの価値です。

■コラム：ビジネスモデルよりワクワク

れみお：「よく『ビジネスモデルどうなんですか？』みたいな、たまに聞かれるときあるじゃないですか。でも今流行っているのって、ビジネスモデルとか聞かれたときにボロクソに言われたものしか流行っていないんだよなって。Twitterとかも最初出たときに、なんで匿名で話してることに意味があるの？とか絶対言われたと思うし。YouTubeとかも動画なんか誰も上げないでしょって」

さえ：「ハッカソンの醍醐味は、ビジネスモデルまで考えなくていいことですね。ワクワクにお金がついてくるとも思いますしね」

既存のビジネスモデルに載る=何かに紐づいている=破壊的イノベーションにはならない。純粋な「こんなのがあったらいいな」から始まるプロダクトにこそ、次の時代を切り拓く力があります。

2.5 抽象化スキルとAI時代のコミュニケーション

現実世界に引っ張られない

アイデアを考えるとき、現実の制約に囚われすぎると発想が狭くなります。

れみお：「アニメとかそこから着想を得るのが良さそうですね。現実世界だと今あるものに引っ張られちゃいそうですし」

れみお：「だからぶっ飛んだ絶対に今だとできないよっていうところも切り口を変えたらこの要素は取れるなっていうのがあったりするんですよね」

SFやアニメの世界は、現実の制約から自由です。そこから着想を得て、技術の側で「どこまで再現できるか」を考える。このアプローチが、既存の延長線上にはない新しいアイデアを生み出します。

WhyとHowを記録する

AI時代のコミュニケーションで大事なのは、インプットとアウトプットだけでなく、「なぜそうしたか」「どうやったか」を記録しておくことです。

入力と出力だけを残しても、なぜその判断をしたのか、どういうプロセスで実装したのかが残っていないと、再現性がなくなります。AIに指示を出すときも同様で、「こういう理由でこういう方針にした」という文脈を与えることで、AIの出力の質は格段に上がります。

ぐー：「少なくともこのAIをどうするかは置いておいて、ここの流れまでは自分で実験できないとダメですよね。AIができるかどうかわかんないっていうのはあるんですけど、ここまでいったらできそうみたいな当たりを結構つけておくのが今は大事だと思っていて」

触って、試して、感覚を掴む。そのグラデーションの中で「ここまでいたらこれは実

第2章 アイデアはオタク知識×技術知識の掛け算スキルとAI時代のコミュニケーション

現できるな」というラインを見極める力。それは、AIには代替できない、実践者だけが持てるスキルです。

第3章

ハッカソン参加＆運営ガイド

2017年からハッカソンに関わり続けてきたさえが、参加者としても運営者としても得てきたリアルな知見を語ります。コアコンセプトへの集中、チーム編成の極意、理想のハッカソン像、そして運営の裏側まで。

3.1 コアコンセプトに集中する

ハッカソンで最初にぶつかる壁は、「何をどこまで作るか」の判断です。特に業務でプロダクト開発をしている人ほど、普段の習慣が足かせになりがちです。

くー：「いわゆる業務で特にプロダクトを作ってる人とかって、セキュリティであるとか細かいコードのフォーマットであるとか、そういったところに目が行っちゃう。そういう自然に起きてしまう制約から自分を解放して、より内容を作るっていうのにフォーカスするっていうのは大事」

くー：「結構ハッカソンでログイン画面作ろうとしてる人とかいたりしますけど、認証認可いらないし、それこそ Kubernetes とか大規模インフラで動く必要もないし、セキュリティとかも別に一旦は考慮しなくともいい。コアのコンセプトというか、自分たちの考えをどうやってプロトタイプに起こすかっていう、コアな機能に集中する」

認証認可はいらない。大規模インフラも不要。セキュリティも一旦は置いておく。ハッカソンで大事なのは、自分たちのアイデアの核心を動く形にすること、ただそれだけです。

さえ：「とりあえず動くものっていう点はそういう意味ですかね」

くー：「うん」

まず参加するところから

とはいえる、初めてのハッカソンでは「動くもの」を作ること自体が難しいものです。

さえ：「結局終わった時に完成はしなかったってなったとしても、やっぱできなかつたことを知るとか、参加して得られることはたくさんあるので、まず参加してもらうが一つ大きいですよね」

くー：「一人で悩まないっていうのがでかいですかね」

完成しなくてもいい。「できなかったこと」を知るだけでも、大きな一步。まずは飛び込むことが、すべての始まりです。

3.2 チーム編成の極意

ライクマインドで集まる

チーム編成はハッカソンの成否を決める最重要ファクターの一つです。くーが大事にしているのは「ライクマインド」、つまり方向性が合う人同士で集まること。

くー：「結局始めての人たち同士でアイデアを決定する合意形成が一番難しいんですよ。なので最終的に全員が妥協して決まったアイデアってワクワクが生まれないので、途中でもリタイアしたくなるし、最終的に成果物として見せるのも恥ずかしくなっちゃう」

くー：「ある程度これ作りたいよねっていう方向性、ライクマインドで似たような形か思っている人たちで集まることを大事にしてますね。スキルとか一旦置いておいて。例えば3人くらいこういう方向性で作ろうってなったときにまだ枠があるのであれば、このスキルがある人に来てほしいなという感じで、4人目、5人目を募集するっていうのが一番いい形」

スキルよりも、まず方向性を合わせる。スキルは後から補える。しかし、チーム全員が「これを作りたい」と心から思えるかどうかは、後から取り戻せません。

心の殻を破る

れみお：「チーム編成において僕が大事にしているのは、心の殻を破ることですかね。やっぱり初対面だと恥ずかしいので、他愛もない会話から始まっちゃいますけど、そんなことやってないで早くこれを作りたいんだっていうのをお話してほしいみたいな」

さえ：「心理的安全性を担保してあげるっていうのは運営のコツですけど、話しやすい雰囲気ができちゃえば、みんながさらけ出して、え、それいいねみたい」

そしてここで、れみおから意外な提案が飛び出します。

れみお：「アルコール入れてからチーム編成開けた方がいい」

さえ：「初日は夜遅くにやって、アルコールから始まって、じゃあチーム編成です。それではチーム編成するのでアルコール飲んでください」

くー：「出来上がってから（笑）」

さえ：「5人以上で乾杯してくださいとか」

特に48時間ハッカソンであれば、Day 1はチームを組んでアイデアを固めるくらいで十分。Vibeコーディングの時代だからこそ、チームビルディングに時間をかけられるようになりました。

多様性が生む化学反応

くー：「仲良い人とチームを組みつつ一方で新しい活性というか刺激もある方が面白いプロダクトが生まれるので、極力新しい人と組んでみるとか全く違う畑の人と一緒にやってみるっていうのは結構あり」

さえ：「ハッカソン結構日本人とかで外国人の日本に住んでいる方が結構来られるじゃないですか。そこで意外と自分にはない考え方とかあったり聞けるのも結構楽しい」

ぐー：「そもそもカルチャーとか違う。ペインも違うし」

さえ：「全然日本人からしたらそうじゃなかっただけど日本のこういう生活でこういうペインがあるんだとか気づけないから、それをプロダクトにするとか」

異なる国籍、性別、職種、バックグラウンド。多様なメンバーが集まることで、一人では絶対に思いつかないアイデアやペインが見えてきます。

さえ：「グローバル、海外の方がたくさん参加したら、勇気持って英語で話せたっていう体験もいいですし、男女のジェンダーバランスとかもあったら心理的安全性を持って女性も気軽に参加できたとか。そういうダイバーシティとかも満足度の秘訣ではありますね」

3.3 理想のハッカソンとは

3人が語る「理想のハッカソン」には、共通するいくつかの要素がありました。

オーディエンス賞の重要性

れみお：「オーディエンス賞があるやつ」

さえ：「ちなみに何ですか」

れみお：「審査員のバイアスがやっぱりでかいのが何回かあったんで。英語できる人に対して英会話のアプリをアピールしても何も響かない。でも君のアイデア良かったよって観客が結構言ってくれるのが多かったりする」

ぐー：「オーディエンス賞は自分たちも確かにここ良かったよねっていうのが結構表彰される印象があるんで、公平に公正に評価されている感が感じられる」

プロダクトに触れる時間

ぐー：「僕大事にしてるのは、プロダクトを見る時間を作るみたいな。ピッチだけだと実際にプロダクトできてないけどかっこよく見せれちゃう人とかも結構多い。作ったものをみんなでわちゃわちゃ触れる時間とかは、あるハッカソンは結構満足度高い」

ぐー：「何より作るってことを条件に追加している以上、作ったものをみんなで手触り感を触ってもらうっていうのが大事」

「手触り感」という言葉が印象的です。ピッチのうまさではなく、実際に動くプロダクトをみんなで触る体験こそが、ハッカソンの本質。

非日常の空間

話は理想のハッカソン会場へ。3人の妄想が止まりません。

れみお：「何も考えずにただやりたいのは、豪華客船とか客船貸し切って、客船が帰るまでに開発を完成させるという」

れみお：「電車もありかもなって思って」

さえ：「寝台列車、でも電車だったらしいんじゃないですか」

ぐー：「僕去年奈良のお寺でハッカソンやったことあるんですけど、やっぱ宿泊伴うと絆が一気に深まるんですよね。みんなが夜を共にするので」

れみお：「キャンプしながらハッカソン普通にあるような気がする」

豪華客船、寝台列車、お寺、キャンプ場、温泉。共通しているのは「非日常」であること。普段のオフィスの延長線上ではない場所に身を置くことで、発想も行動も変わります。

番組ありきのハッカソン

さらに、3人はハッカソンの未来像として「ドキュメンタリー」の可能性に話が及びます。

さえ：「ドキュメンタリー風にしてもらうとか。この熱狂をより多くの人たちに知ってもらうという」

ぐー：「番組ありきみたいな。自分たちも見られる前提になったらちょっと変わるかもしれないし。まず生放送で流しておいて、人が来るのとその反応を見たりとかできたら面白いかもしない」

さえ：「視聴者の中でボーティングしてもらったりもいいですし」

■コラム：面白い賞の事例

- **ヨーデル賞**：奈良のお寺での Music & AI ハッカソンで、ドイツのヨーデルインフルエンサーに「刺さる」プロダクトを作るという特別賞。ペルソナが明確で盛り上がった
- **コード品質賞**：実装の機能数ではなく、コードの品質で評価するハッカソン。学びが多い
- **マーケット賞**：作ったプロダクトを実際にリリースして、リポスト数やインプレッションで競う。ビズ側の人が輝ける

3.4 ハッカソン運営のコツ

熱量のある人とお金の出どころ

さえが運営で最も大事にしている2つの要素があります。

さえ：「一番大事にしてるのは、熱量のある人と、お金の出どころみたいな感じですね。どれだけモチベーションあっても結局スポンサーとかサポートしてくださる企業がいないと実現できないので」

さえ：「もう一つが結局ハッカソンを開催するのは結構タフなので、そのタフでもこれはどうしても実現したいんだっていう思いがめっちゃ強い人がいること。この二つがちょっと

と開催においてめっちゃ大事です」

運営の熱量

さえ：「もう一つ述べておきたいのが、運営でハッカソンに興味ない人が主催するハッカソンは面白くないんですよ。例えばハッカソン開催期間中にべちゃべちゃ喋ってるだとか、別の仕事しちゃってるような人が運営してたら、それはつまらないハッカソンで」

さえ：「じゃなくて運営の人が参加者と同じか、それを超えるぐらいの熱量でワクワクしてたりだとか、一緒にこのハッカソンを盛り上げていこうぜみたいなタイプのハッカソンだとすごい面白いですね」

運営の熱量は、参加者に必ず伝染します。「やらされている」運営と「やりたくてやっている」運営では、ハッカソンの雰囲気がまるで違う。

社内ハッカソンのポイント

社内でハッカソンを開催する場合、特有の課題があります。

さえ：「なおかつその期間中は業務に支障をきたしちゃうので、そこを VP とかが責任を取れる、了承できるかは結構大事ですね。やっぱり参加者にとってはもうそこにフルコミットすることが何より価値なので。業務と片手間でやりながらだと結局得られるもの全然ない」

さえ：「あと社内ハッカソンのメンター、めっちゃ大事かもしれないですね。社内の人たち同士だと仕事の延長線上にしかならないので。こういう技術使えるよとか、こういうアイデアはどうみたいな導いてくれるメンターさんは大事」

社内ハッカソン成功の鍵は3つ。VP レベルの後ろ盾でフルコミットの環境を作ること、社外メンターで視野を広げること、そして何より社内に本気の推進者がいること。

さえ：「VP とかなんかそのトップから言われたからやる、だと失敗しちゃうので、本当にこれやりたいんだって言ってる人が社内にいれば話は早い」

3.5 賞金設計と出続ける理由

賞金は50万円以内に

さえ：「50万円超えてくると、源泉徴収とか税金がすごいめんどくさいので。基本的に50万円以内に納める方が運営上はすごい楽です」

さえ：「あと100万円とか出すと、参加者の満足度よりもクレームの方が増えるんですよね。『なんでうちのチームが優秀じゃないんだ』みたいな感じで。お金が人を狂わせることが多いので」

さえ：「何なら50万円でも参加する人は参加しますし、金額が高ければいいって話ではないというのに気づいてきました」

ぐー：「あったら嬉しいですよね。今一番クレジットが欲しいですね」

賞金よりも、API クレジットやクラウドサービスの利用券のほうが、開発者にとっては実用的で喜ばれることもあります。

ハッカソンに出続ける理由

最後に、なぜ3人はハッカソンに出続けるのか。

れみお：「3人共通して言えたのがまずリコグナイズされた経験。表彰されることでSNSに上げてみんなに『おめでとう』みたいな言われるところから、次の登壇の機会とか」

れみお：「マジでYCのやつは今でも輝いています。そこで勝ったんだねというところで、目の色が変わる」

さえ：「勝ちまくっていくと、なるほど、『ハッカソンを勝っている人なんだ』っていう認知がすごいです」

ハッカソンで得られるのは賞金だけではありません。認知、つながり、自己表現の機会、そして何より「自分はこれを作れる」という確信。

さえ：「自分の普段超えられない壁を超えるチャンスではありますし」

ハッカソンは、大人の青春であり、文化祭であり、自分を超える舞台です。

第4章

AI活用のリアルなTips

最終章は3人全員で語り合う、AI活用の実践的なTips集です。音声入力の活用法、よくあるアンチパターン、AIとの距離感の取り方、そして読者へのメッセージ。

4.1 音声入力のすすめ

この本自体が「音声で語り合い、AIが本にまとめる」という実験ですが、音声入力はAI活用全般において強力な武器になります。

ぐー：「言語化しようとすると、テキストに言語化しようとすると言語化するっていうステップが入るから出てこないし、何か間違った、自分が思ったことじゃないことを言いがちだけども、話すともうVibeで音声が出てくるから」

ぐー：「やっぱり今のAIって正しくプロンプトを入れるっていうよりも、とにかく情報を入れるっていう方が大事だと思うんで。自分が思っていることをもうすべて出す。出した後に見直してこれ足りてないなっていうのを気づければそこを追加で話してあげる」

さえ：「確かに膨大なインプット量が音声だと入れやすい」

テキスト入力では、頭の中で「正しい文章」に整えるというフィルターが働いてしまい、本当に言いたいことが出てこないことがあります。音声なら、思考をそのまま垂れ流せる。

ぐー：「抽象的なやつをテキストにするのは難しいけども、音声だったら『このオレンジのやつがこういう風になってて、ちょっと出っ張って』みたいなのはずっと言えるから」

れみお：「アイデア出しっていうか、これもまさに今そうですもんね音声」

ぐー：「まあ確かにそう、このやってる取り組みがもう音声で話した方が楽だよねっていう」

さえ：「もうこれ形式化した方がいいですね」

特にアイデア出しやブレインストーミングの段階では、音声入力が圧倒的に有利です。まずは思いつくまま喋って、AIに整理してもらう。正確なプロンプトを書くことよりも、情報量を最大化することが重要です。

4.2 「知らないを知る」力

さえ：「何を知らないかを知る。ちょっと違うけども、何を知らないか知ることをどうやったら分かるんだろうは結構難しい」

さえ：「マインドを変えるはさっきの話でしたけど、まあとはいえたマインドを変えろって言って変わるわけでもないので、どうしたらいいんだろう」

「何を知らないかを知る」はソクラテスの昔から言われてきた知恵ですが、AI時代においてはより切実な問題です。自分が知らないことを知つていれば、AIに適切な質問ができる。しかし、知らないことすら知らなければ、そもそも質問が思いつきません。

その解決策の一つが、まさに前節で触れた「とにかく触ってみる」こと。触ることで「あ、自分はここが分かってなかったんだ」という気づきが得られます。

4.3 AI 活用のアンチパターン

3人が経験から導き出した「やってはいけないこと」のリストは、AI活用の落とし穴を端的に示しています。

十分な情報を与えない

ぐー：「十分な情報を与えない」

さえ：「十分な情報を与えないと十分な結果を得られない」

ぐー：「得られないからチグハグな情報の結果になる」

AIは与えられた情報からしか推論できません。「いい感じにやって」という曖昧な指示では、曖昧な結果しか返ってきません。自分が何を求めているのか、どういうコンテキストなのか、具体的に伝えることが大前提です。

触らずに諦める

ぐー：「あとは触らない」

さえ：「AI技術って音声入力技術はまだまだだって諦めるんじゃなくて」

ぐー：「さまざまなものとかツールで触ったのかみたいな。良いものも良くないもの両方触れてみると」

一つのツール、一つのモデルで試して「ダメだった」と結論づけるのは早すぎます。同じタスクでも、モデルが違えば結果は大きく変わる。複数のツールを横断的に試してから判断すべきです。

定期的な「健康診断」をしない

さえ：「昔ダメだったけど今触ったらどうなのか」

ぐー：「健康診断みたいな感じで」

ぐー：「さっきのLTの資料あれManusで作ったんですけど、3、4ヶ月前のManusの

スライドの作り方は全然違うんですよ」

さえ：「当時は使えないと思って諦めちゃったやつとかも、半年ぶりに触ってみたら全然使用感変わってるじゃんってあります」

AI ツールの進化は驚くほど速い。半年前に「使えない」と判断したツールが、今日触ったら全く別物になっていることは珍しくありません。定期的に過去のツールを再評価する「健康診断」の習慣を持ちましょう。

さえ：「さっきのその年間サブスクにしないみたいな、1ヶ月単位で課金するみたいな話で、一度諦めたやつでも時々また戻ってきて使用感確認するっていう感じは良いかもしれません」

決めつけて終わる

さえ：「さっきの失敗したと思って終わるというか、そこを学びじゃなくて実力って思つてやめちゃうのは、今の AI 活用のアンチパターンの一つかな」

さえ：「それをやらないがアンチパターンですね、確かに。もう決めつけて終わる」

一度の失敗を「AI はダメだ」「自分には向いていない」と結論づけてしまうこと。それが最大のアンチパターンです。

■コラム：AI 活用アンチパターンまとめ

1. 情報不足：十分な情報を与えずに曖昧な結果に不満を持つ
2. 一つで判断：一つのツール/モデルだけ試して「AI は使えない」と結論づける
3. 健康診断しない：過去に使えなかったツールを再評価しない。半年で別物になっていることも
4. 決めつけ：失敗を「AI の限界」と決めつけて学びにしない
5. お膳立てゼロ：AI にすべてを丸投げして、ビッグピクチャーを人間が描かない

4.4 AI との距離感

AI に対するスタンスも、大きな差を生みます。

れみお：「AI が自分より頭良いことを受け入れるとか。AI より自分がなんか全部の領域でまだ頭良いって思い込んでやると、逆にそれが足かせになるケースとかあったな」

さえ：「そうですよね、自分より頭良いって」

れみお：「頭良いっていうか、その、自分がお膳立て全部しないといけないって思い込んでた」

さえ：「なるほど、丁寧に赤ちゃんに教えている感覚じゃなくて、もう頼れるパートナーみたいな」

れみお：「ジュニアエンジニアぐらい。諦めない」

くー：「同僚までは行かないけど、ジュニアぐらいは。物によっては普通にシニアとか

あると思う。ちゃんと情報を整理して与えられたら」

AIを「赤ちゃん」のように扱うのでも、「万能の神」のように扱うのではなく、「優秀だけど文脈を知らないジュニアメンバー」くらいに捉える。必要な情報を渡し、方向性を示せば、驚くほど良い仕事をしてくれます。

ベストプラクティスは先に調べさせる

れみお：「自分が本当に分からることはAIにやらせてもベストプラクティスでやつてくれないと。ビッグピクチャーを人間が描いて、それ通りにやってって言うと案外いいんですけど。そういうと、とりあえず適当な手法でやり始めたりして」

ぐー：「コーディング文脈で言ったんですけど、その手前でAIにベストプラクティスを現状調査させるみたいな段階を挟むと割と良くて。『こういう場合はどういうのがベストプラクティス？』みたいな。学習したデータじゃなくて、Web検索してちゃんと見せた方がいい」

実装の前に、まずAIにベストプラクティスを調査させる。この一手間が、後の大幅な手戻りを防ぎます。

4.5 さいごに：とりあえず手を動かそう

さえ：「じゃあ最後、この本で伝えたかったこと」

れみお：「とりあえず手を動かそう」

さえ：「ハッカソンも参加したら結果的に使う、チャレンジするっていうことも一つだと思うので。AIにおいても使ってみるとか、手を動かしてみるとか」

れみお：「とりあえず1万円握りしめて課金しまくってハッカソン出ればいいんじゃない？」

さえ：「ハッカソンに参加して、とりあえず1万円使い切れって」

れみお：「AI系サブスク全部して1万円。そしてハッカソンで実際使ってみればわかるんじゃない？」

ぐー：「今3,000円になったら教育コードも使えるし」

れみお：「あと恥ずかしがったりとか、別に失うものは何もないんだから」

さえ：「失うものはあると思っている人が多いから。恥ずかしいとか失敗したら、とか」

れみお：「そんなものはそもそもないんだ、お前にはないんだから。そもそも失うものなんかないんだから、とりあえずやればいい」

れみお：「僕らはできる側なんで恩恵を得ているっていうのがある。今別にそういう人が増えてもまだ全然ポジション取れるから。もっと増えるべきかなと思います」

さえ：「まだまだ挑戦できない方が大多数だと思うので。ぜひ挑戦してほしいという感じですね」

失うものなんかない。恥ずかしがることもない。とりあえず手を動かそう。

それが、3人のハッカソン仲間から読者のみなさんへの、最後のメッセージです。

あとがき

この本を手に取ってください、ありがとうございます。

週末に3人で集まり、ひたすら喋り続けた数時間。その会話をAIが本文の形にまとめてくれました。まさにこの本自体が、私たちが語った「AIと共に考える」の実践そのものです。

私たちが伝えたかったことは、とてもシンプルです。

「とりあえず手を動かそう」

「失うものなんかないんだから、とりあえずやればいい」

AIの進化は止まりません。半年前に使えなかったツールが、今日触ってみたら全然違うものになっていることもあります。だからこそ、評論家にならず、自分の手で触って、自分の目で確かめてほしい。

ハッカソンに参加したことがない人は、まず一度飛び込んでみてください。完成しなくてもいい。できなかったことを知るだけでも、大きな一歩です。

そして、この本が少しでもみなさんの「願い」を見つけるきっかけになれば、これほど嬉しいことはありません。

著者紹介

くー (kuu)

AIネイティブ時代のソフトウェアエンジニア。AIを「共に考えるパートナー」として捉え、ガードレール開発手法やVibeコーディングを実践する。国内外のハッカソンで入賞多数。ツール選びのモットーは「開発者が何を目指しているかを見る」こと。年間契約はしない主義。

れみお

自称アイデアマン。オタク知識と技術知識の掛け算でハッカソンのアイデアを生み出す。メモ帳には常時30~50個のアイデアが温められている。フィンランドで開催されたヨーロッパ最大級ハッカソンの生成AI部門で優勝、Y Combinatorのハッカソンでも入賞。建設業界からIT業界に転身した異色の経歴を持つ。

さえ (sae)

付録 あとがき

ハッカソンオーガナイザー。2017年からハッカソンに参加し、社内外で数多くのハッカソンを企画・運営してきた。「熱量のある人とお金の出どころ」を大事に、参加者も運営も全力で楽しめるハッカソン作りを目指す。Try Swift Student ハッカソンなどの運営も手がける。

はじめての電子書籍
誰でも簡単！ **Re:VIEW** による技術同人誌執筆

2022年11月11日 初版第1刷 発行

著 者 白井暁彦
