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

AI 時代のハッカソン

3 人のエンジニアが語る、AI と共創するものづくりの実践知

Kuu、Remio　著

2026-02-07 版　　発行

はじめに

3 人のエンジニアがハッカソンと AI の交差点で見つけたこと

この本は、ハッカソンを愛する 3 人のエンジニアが、AI 時代のものづくりについて語り合った記録です。

私たちは何度もハッカソンに参加し、運営し、審査してきました。その中で、AI の登場によってハッカソンの風景が大きく変わりつつあることを肌で感じています。Vibe Coding で誰でもプロトタイプを作れる時代に、ハッカソンで何を競い、何を楽しむのか。エンジニアのマインドセットはどう変わるべきなのか。そして、非エンジニアにとってのハッカソンの可能性はどこまで広がるのか。

2026 年 2 月、ブッカソン（本のハッカソン）という場で、私たちはこれらの問いに向き合いました。本書はその対話から生まれました。

この本は「対話」で読む技術書である

対話形式を採用した理由——１人の語りに２人が問いを投げる構成

本書は、一般的な技術書とは異なる構成を採用しています。各章では 1 人の著者が主に語り、残りの 2 人がツッコミや質問を入れる対話形式で進みます。

1 人で書くと出てこない気づきが、対話からは自然に生まれるからです。「それってどういうこと？」という素朴な質問が、読者にとっても理解の助けになります。また、ディープリサーチでは得られないリアルな経験——実際にハッカソンで AI を使って感じたこと、うまくいかなかったこと——は、対話の中でこそ引き出されます。

本書の読み方と各章の役割分担

本書は 4 つの章と付録で構成されています。

- **第 1 章「AI ネイティブという不可逆な変化」**（Kuu）—— AI がエンジニアの仕事やマインドセットをどう変えるのか、そして AI を「ガードレール」で囲んで活用する開発手法について語ります。
- **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」**（Remio）—— ハッカソンでのアイデア発想法、自己表現としてのものづくり、そしてアイデアを仕様に落とし込む技術について語ります。
- **第 3 章「ハッカソンという『枠組み』が人を動かす」**（Kuu・Remio）—— ハッカソンの構造的な力、非エンジニアへの門戸開放、運営の実務ノウハウについて語り

ます。

- **第 4 章「AI 活用の実践テクニック」**（全員）—— 第 1〜3 章の知見を横断し、ハッカソンでも仕事でも使える AI 活用テクニックをまとめます。

どの章から読んでも構いません。興味のあるテーマから読み始めてください。

想定する読者——エンジニアもそうでない人も

本書は以下のような方を想定しています。

- ハッカソンに参加したことがあり、AI 時代の変化を感じているエンジニア
- ハッカソンに興味はあるが、まだ参加したことがない方
- AI ツールを使ってものづくりを始めたい非エンジニア
- ハッカソンの運営・企画に携わる方
- 「手を動かすことの価値」を再確認したいすべてのクリエイター

技術的な予備知識は必要ありません。対話形式で進むので、わからない用語があっても文脈から理解できるよう心がけています。

目次

第 1 章

AI ネイティブという不可逆な変化——エンジニアの仕事はもう元に戻らない

AI が当たり前になった世界で、エンジニアに求められるマインドセットはどう変わるのか。具体的なツールの使い方ではなく、永続的に必要となる考え方の転換について、Kuu が語ります。

1.1 「AI を使う」から「AI と共に考える」への転換点

2026 年に起きているのは一過性のブームではなく思考様式の変化である

Kuu：自分が特に書きたいのは、AI ネイティブになった時にどういうマインドチェンジが求められるのか、というところです。2026 年 2 月の話だけじゃなくて、おそらく永続的に変わるもの。

Remio：それってエンジニアだけの話？

Kuu：エンジニアのマインドがまず大きいですが、他の職種も含めた話です。ハッカソンでの人間の役割分担がだんだん消えていって、将来的には全員が同じ一つの職種くらいまで染み出して溶けると思うので、全員向けの話でもあります。

TODO: AI を道具として使う段階から、思考プロセスの一部として組み込む段階への移行について詳述する

1.2 職種の境界線が AI によって溶解する

エンジニア・デザイナー・PM の肩書きが AI 時代に意味を失う理由

Kuu：AI のおかげで、職種という肩書きがどんどん取り払われる感じはありますし、エンジニアでなくても貢献できる形が増えてきています。

TODO: 職種間の壁が AI によって下がっている具体例を追加する（例: デザイナーが Vibe Coding でプロトタイプを作る、PM が API を直接試す等）

「コードが書けない人」がプロダクトを作れる時代の到来

TODO: Vibe Coding やノーコード/ローコードの進化により、非エンジニアがプロダクトを作れるようになった事例を紹介する

1.3 AI をガードレールで囲む開発手法

最新の API リファレンスを起点にしないと AI は古い情報で嘘をつく

Kuu：AI をそのまま使うと、古い情報でハルシネーションを生み出します。だからこそ「ガードレール」という考え方が重要なんです。AI に自由に動いてもらう前に、正しい方向に走れるよう枠を作っておく。

Remio：ガードレールを引いた上で AI に暴れさせることで、より良い結果が出ると。

Kuu：そうです。この「まず枠を作ってから走らせる」という発想が、AI 時代のエンジニアに求められるマインドセットの一つだと思います。具体的な方法——たとえば公式ドキュメントの URL をプロンプトに含める、バージョンを明示するといったテクニックは第 4 章「AI 活用の実践テクニック——ハッカソンでも仕事でも使える知恵」で詳しく紹介します。

TODO: ガードレール思考がなぜ AI 時代のマインドセットとして重要かを詳述する（具体的な設定方法は第 4 章で解説）

JS/Python のライブラリ依存地獄を AI が悪化させる構造的問題

TODO: ライブラリのバージョン違いによる AI 生成コードの問題と、その対策について詳述する

CI パイプラインと Lint を AI のガードレールとして活用する

Kuu：ガードレールを引くために、既存の CI や Lint のような仕組みは今でも使えますし、むしろ重要性が上がっている気がします。

TODO: CI/CD と Lint が AI 生成コードの品質管理にどう機能するかを具体例とともに解説する

1.4 ツールの選び方——特定ツールではなく「強いツールの条件」を知る

特定のツール名ではなく「強いツールの特性」を見極める

Kuu：具体的なツールの使い方というよりは、こういう特性のツールが強い、というのを書いていきたい。ティップスみたいな「こういう時はこう」というのはおそらくティッ

プス化しづらいかなと思っています。

TODO: 良い AI ツールに共通する特性（コンテキスト理解力、拡張性、ドキュメント品質など）を整理し、半年後もツールを選び直せる陳腐化しない選定基準（コミュニティの活発さ、API 設計の一貫性、エコシステムの広さなど）も含めて解説する

1.5 Vibe Coding が変えるハッカソンの難易度と面白さ

Vibe Coding とは何か——感覚駆動でプロトタイプを生み出す開発スタイル

Remio：最近 Vibe Coding が流行ってきて、ハッカソンが退屈に感じることがあって。

Kuu：ただのアプリケーションだと本当に簡単に 3 秒でポッとできてしまうので、それ以外の技術を絡めたものが求められます。

TODO: Vibe Coding の定義と、従来の開発スタイルとの違いを解説する

AI で「作る」が簡単になった今、ハッカソンに求められる新しい挑戦

Kuu：基本的な実装がすぐ終わるからこそ、新しいフォーメーションや挑戦に時間を割けます。もし詰まるハッカソンで冒険したら何も作れないリスクがありますが、余裕がある前提なら挑戦できる。

Remio：心理的安全性が担保された上で、「最悪ここまでは作れる、だからこそ新しいことに挑戦しよう」という発想ですね。

TODO: 開発が容易になったことで、ハッカソンの価値がアイデアや統合力に移行し、新しいフォーメーションへの挑戦が可能になった現象を解説する

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

ハッカソンでのアイデア発想法から、チームでの合意形成、アイデアを仕様に落とし込む技術まで。Remio が自身の経験をもとに、ハッカソンにおける「自己表現としてのものづくり」を語ります。

2.1　ハッカソンは競技ではなく「自分を披露する場」である

勝ちに行くのではなく「作りたいもの」を作る姿勢が結果を出す

Remio：今思うと、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なと。意外と自己表現の場でもあるのかなと感じます。

Kuu：ちょっとアーティストですね。

Remio：「今これがないから作りたい」というアイデアを貯めておいて、ハッカソンの場で出す。そのたびに作品が生まれるのは、自己表現なのかもしれません。

TODO: 勝ちを目的にするのではなく、自分の作りたいものを作る姿勢がなぜ結果的に評価されるのかを詳述する

2.2　アイデア発想の源泉――趣味と技術の引き出しを掛け合わせる

日常の「これ不便」「これ面白い」を技術で解決可能な形に変換する思考法

Remio：オタクの知識と技術の知識を両側から攻め合わせて、できそうなところでプロダクトが生まれる。

Kuu：それをどうやって生み出しているのか、普段からやっていることや、情報キャッチアップの方法を言語化してほしい。

Remio：技術的実現性がわかっているからこそ、「なぜ実現できないのか」を「どうい

う切り口なら実現できるか」に分解していけます。その過程を言語化したいですね。

TODO: 日常の観察から技術的なアイデアに変換するプロセスを具体的なステップで解説する

過去のハッカソン作品から見るアイデア発想の実例

TODO: Remio の過去のハッカソン作品を例に、コンセプトからプロダクトまでの過程を紹介する。うまくいかなかったことも含める

1 人で考えるより「雑談」から生まれるアイデアの方が強い理由

TODO: 対話や雑談がアイデア発想において持つ力、1 人のブレストとの違いを解説する

> ■コラム: Remio の過去プロジェクト紹介——アイデアの種がどう育ったか
>
> TODO: 過去のハッカソンで作ったプロダクトを数点紹介し、各プロジェクトのアイデアの発端と開発過程を振り返る

2.3 チームでアイデアの合意を取るのが最も難しい

個人のビジョンとチームの方向性がぶつかる瞬間

Kuu：初めましてで組んだ時のアイデアの合意形成も結構難しいですよね。

Remio：「それ面白い」「それでいけそう」とみんなが納得するアイデアなのか、みんなで練り上げて生まれてくるアイデアなのか。その違いは大きいですね。

TODO: チーム内でのアイデア対立と合意形成の難しさを、実体験をもとに掘り下げる

2.4 アイデアから仕様書へ——構想を実装可能な形に翻訳する技術

「面白い」だけでは開発が始まらない——アイデアと仕様の間にある崖

Kuu：アイデアと開発の間の仕様決めは大事だと思っています。「このアイデアをやりたい」と言っても、仕様を考えて実装まで持っていけないことがあるんですよね。

Remio：仕様に落とし込めないチームは多いと思います。アイデアを考えても地に足がついていなくて、「どうやって実現するか」を考えた結果、ありきたりなものになって

しまうパターンですね。

TODO: アイデアが仕様に変換されずに頓挫するパターンと、その回避策を解説する

仕様に落とし込むステップ——ユーザー体験→画面遷移→ API 設計

Remio：僕は最初に「入力がこの状態で、この処理をして、この結果を出す」という流れを全部決めてから、仕様書を作って開発します。わからない部分だけ、「ここからこうしたいけど、どの技術スタックを使えばいいか」を相談して壁打ちしながら進めますね。

TODO: 抽象的なアイデアを段階的に具体化するプロセスを体系的に解説する

AI を使ってアイデアを仕様書に変換するワークフロー

TODO: AI ツールを使ったアイデア→仕様変換のワークフローを、具体的なプロンプト例とともに紹介する

2.5　評論家にならず手を動かす人が最後に勝つ

「それ無理じゃない？」と言う人より「とりあえず作ってみた」人が評価される

Kuu：評論家にはなりたくないですね。何が良くて何がダメなのか、「すごい」と言うけれど本当にすごいのか、従来と何が違うのか。動かさないとわかりません。自分の手で動かして、生の声で語ってほしいですよね。

TODO: 実際に手を動かすことの価値と、評論だけに終わるリスクを対比して論じる

第 3 章

ハッカソンという「枠組み」が人を動かす——運営と参加の実践知

ハッカソンはなぜ人を惹きつけるのか。その構造的な力から、非エンジニアへの門戸開放、軽量ハッカソンの提案、運営の実務ノウハウまで。ハッカソンの「枠組み」が持つ力を掘り下げます。

3.1 ハッカソンが持つ構造的な力——締切・他者・発表の三要素

2〜3 日の制限時間が「完璧主義」を強制的に解除する

Kuu：ハッカソンはフレームワークとしてすごく良いと思うんですよ。2、3 日で締め切りを作って、初対面の人たちと何かを作る。必ず発表しなければならないというプレッシャーも良い。

TODO: 時間制限が完璧主義を打破し、「とりあえず動くもの」を作る文化を生み出す仕組みについて解説する

全員がデモを見せる義務がある制約がアウトプットの質を底上げする

TODO: 発表義務がもたらすポジティブなプレッシャーと、アウトプットの質向上への影響を解説する

3.2 ハッカソンは非エンジニアにこそ開かれた場である

Vibe Coding の登場で「コードが書けない人」の参加障壁が消えた

Remio：フランスで学生ハッカソンを視察したとき、Vibe Coding が初めてという参加者もいました。今までハッカソンはエンジニアのものでしたが、一般の方に広げていくのは面白いかもしれません。

Kuu：プロダクトを自分たちで作れなかった人たちは、すごく楽しんでいるんですよ。

作りたいものがたくさん溜まっていて、それが形になったときの感動は大きいです。

TODO: フランスの学生ハッカソン視察での具体的な体験と、非エンジニア向けハッカソンの参加者の反応を詳述する

デザイナー・企画者・ドメイン専門家がハッカソンで発揮できる固有の価値

Remio：デザイナーが実際にアプリまで作れるようになると面白いですよね。エンジニアとの会話も円滑になりますし。

Kuu：「ここまでは簡単にできて、ここからはエンジニアの技術が必要」という境界線を体感できるのも価値があります。

Kuu：Apple Vision Pro ハッカソンでは、Vision Pro を持っていない人が、数年先に普及するデバイスを実際に触って、しかもアプリまで作れました。クラウドのクレジットが提供されるハッカソンもあり、特に学生にとっては貴重な機会です。

Remio：新しい技術を触ってみたいと思いつつ、時間やお金がなくて挑戦できていない人が、最新技術のすごさを体感できる。しかも多くの人と話しながら進められるのは醍醐味ですね。

TODO: 各職種がハッカソンで持ち込める価値（ドメイン知識、UX 視点、ビジネス視点）と、Vibe Coding により職種間の相互理解が深まる効果を解説する。メンターの配置、チーム構成の工夫、敷居を下げるネーミング（「ブッカソン」等）の事例を紹介する。Apple Vision Pro ハッカソン等の具体例を通じて、最新技術に触れる場としてのハッカソンの価値、特に非エンジニアや学生にとって高価なデバイスやクラウド環境に無料でアクセスできることが参加障壁を下げる効果にも焦点を当てる

3.3 軽量ハッカソンのすすめ——5 時間で開催する即興型イベント

「新しい API が出た、今夜ハッカソンしよう」という文化の作り方

Kuu：エンジニアはもっと気軽に、「今日この技術が発表されたから、このあと 5 時間でハッカソンやりませんか」と招集するような、カジュアルなスタイルもいいかもしれないですね。

TODO: 即興ハッカソンの具体的な開催手順と、最低限必要な準備を解説する

即興ハッカソンで得られる「鮮度の高い技術キャッチアップ」

Kuu：エンジニアこそ、新しい技術が出たら必ずその技術を使ったハッカソンを小まめに開催した方がいいかもしれないですね。

TODO: 最新技術の発表直後にハッカソンを行うことで得られる学びの質について解説する

3.4 ハッカソン運営の実務——タイムライン・グルーピング・審査

デモ前のテスト時間を確保しないと発表が崩壊する

Kuu：事前に発表のテストをしっかり行って、テクニカルイシューを起こさせないことが大事ですね。

Remio：他のハッカソンでは事前の練習がないことが多いですね。接続したら動画が再生できないとか。

Kuu：僕も 1 回、テストでは音が出たのに本番で音が出なかったことがあります。

TODO: デモ前テストの重要性と、具体的なチェック項目を解説する

グルーピング手法——ランダム vs スキルミックス vs テーマ別

TODO: チーム分けの各手法のメリット・デメリットを比較する

審査基準と賞の設計——多くの参加者が受賞できる仕組みが次回参加を生む

Kuu：ハッカソン運営をする側としては、賞をなるべく増やしていきたいです。小さな賞でも、受賞することでハッカソンを好きになる理由が生まれます。ハッカソンの沼にはまるきっかけになりますね。

Remio：作ったものに正解はないので、本来はみんな優勝みたいなものです。作りきった人たちは全員優勝、というところはありますね。

Remio：全員のピッチを聞くと、練習できていないチームの発表は中身が薄いのに時間が長くなりがちです。

Kuu：予選・決勝方式もありだと思います。予選でプロダクトを触って評価し、ファイナリストだけがピッチをする。プロダクトの完成度とピッチの質の両方を評価できます。

Remio：予選側は動画を作る必要が出てくるので、参加者の負担は増えますが、1〜2 ヶ月かけたハッカソンであればファイナリストのピッチは精練されていて、聴く側も飽きません。

TODO: 賞の設計哲学（技術賞・アイデア賞・プレゼン賞の分離）と、受賞体験がリピーターを生むメカニズムを解説する。予選・決勝方式の審査のメリット・デメリットと、規模に応じた審査設計の指針も述べる

時間管理のコツ——なぜハッカソンは 2 時間押すのか、押さない運営の秘訣

Remio：他のハッカソンは絶対に 2 時間ぐらい押すんですよ。なんで押さないんですか？

Kuu：拍手で盛り上げるんです。みんなが「よかったよ」という雰囲気になると、「ここで終わった方がいい」という空気ができあがります。それでオンタイムに収まるんです。

Kuu：開発が簡単になった今、アイデアを練る時間を十分に取らないと、ありきたりな作品になりがちです。「開発しなきゃ」という焦りが、アイデアのブラッシュアップを後回しにさせてしまう。

Remio：逆に、ある時間まではアイデアを必ず議論しなければならない「開発禁止期間」を設けて、そこから先は開発に集中するというルールはどうですか。

Kuu：それはメリハリがあっていいですね。「ここまではアイデアを練ってください、ここからはアイデアを変えないでください」と明確に区切ることで、両方の質が上がりそうです。

TODO: ハッカソンが時間を押す原因と、タイムキーピングの具体的なテクニックを解説する。アイデア出しと開発の時間配分について、開発禁止期間の導入メリットと具体的な運用方法も述べる

3.5 ハッカソンに出続ける理由——承認・キャリア・アドレナリン

成果が即座に評価される快感は通常業務では得られない

Kuu：ファイナリストに選出されたとき、仲間同士で「おめでとう」と熱狂できたのは大きいですね。やみつきになりました。

Remio：勝つのはいいですよね。アドレナリンが高まりますし、認められるのはやはり大きいです。

TODO: ハッカソンでの即時フィードバックが参加者のモチベーションに与える影響を分析する

ハッカソン実績がポートフォリオとして転職・昇進に効く

TODO: ハッカソン実績がキャリアに与える具体的な影響（就活、スポンサー企業との接点、スキル証明）を解説する

「勝つ」以外のモチベーション——仲間・学び・自己更新

Kuu：プログラミングが好きで、ダラダラ作るよりも作りきった方が楽しい。たまに優勝できるのが良くて、ギャンブル性が高いんです。特に AI が出てきて、優位性がある中で戦えるのは最高じゃないですか。

TODO: 勝利以外のモチベーション（学び、出会い、新技術の習得、自己表現）を整理する

第 4 章

AI 活用の実践テクニック――ハッカソンでも仕事でも使える知恵

第 1〜3 章で語られた知見を横断し、ハッカソンでも日常業務でも活用できる AI 活用の実践テクニックを 3 人でまとめます。具体的なツール名ではなく、考え方とアプローチに焦点を当てます。

4.1　AI に正しく仕事をさせるための前提条件を整える

公式ドキュメントの URL をプロンプトに含めるだけで回答精度が劇的に上がる

Kuu：第 1 章「AI ネイティブという不可逆な変化――エンジニアの仕事はもう元に戻らない」でガードレールの考え方を話しましたが、具体的な方法として一番効くのは、最新の API リファレンスの URL をプロンプトに含めることです。AI に読ませるだけで回答精度が全然違います。

Remio：ドキュメントの URL を渡すだけでそんなに変わるんですか？

Kuu：変わります。特に JS や Python はライブラリのバージョン更新が速いので、公式ドキュメントを起点にしないと AI が古い API で書いてきます。URL を渡す、バージョンを明示する、この 2 つだけでつまずきにくくなります。

TODO: 公式ドキュメントを AI に読ませる方法、コンテキストの鮮度管理の習慣、AI のミスを自分のプロンプト設計の問題として捉える思考法を含めて解説する

古いライブラリバージョンのコードを AI が自信満々に生成する問題

Kuu：AI は学習データの時点で最新だったバージョンのコードを自信満々に書いてきます。動くけれど非推奨の API を使っていたり、そもそもメソッド名が変わっていたりする。

Remio：それ、動くだけに気づきにくいですよね。

Kuu：だからこそ、使うライブラリのバージョンとドキュメントをプロンプトで明示

する習慣が大事です。

TODO: AI が古いバージョンのコードを生成する具体的な事例と、その対策を解説する

4.2 AI を学習パートナーとして使い倒す

理解できるまで何度でも聞き直す—— AI は嫌な顔をしない最強の家庭教師

Remio：わかるまで聞けばいいんです。「わからない部分をもう一度説明してください」と何度でも聞く。そのやり方をレクチャーするのが、ググり方を教えるのと同じ感覚ですね。

Kuu：やり方さえわかれば、自走できますしね。

TODO: AI に対して「わからないところがわかるまで質問し続ける」手法と、その教育的効果を解説する

未知の概念を「絵を見せて聞く」ように学ぶ——わからないことがわかれば深掘りできる

Remio：昔テレビで見たんですが、アイヌの方にアイヌ語を教えてもらうとき、ぐちゃぐちゃの絵を描いて見せて、相手の反応から語彙を学んでいく方法がありました。わからないことさえわかっていれば、知識を広げられます。AI も同じように使えばいいんです。

Kuu：自分がわからないことさえわかっていれば、わかっている部分とわからない部分の境界を伝えて、わからない部分だけを深掘りしていけば全体が見えてきます。

TODO: 「自分が何を知らないかを知る」ことから AI 学習を始めるアプローチの体系化

音声 AI で曖昧な検索をする——オノマトペや身振りで伝える新しい検索体験

Kuu：テキスト検索だとテキストに言語化しなければなりませんが、音声なら擬音語やオノマトペを使って、ふわっとした形から絞り込んでいくことができます。

Remio：昔見た YouTube の動画で、タイトルが全然わからなくても、「こういう感じのやつ」と探していったら出てきたりするんですよ。

TODO: 音声入力と AI を組み合わせた曖昧検索の実践例と、テキスト検索との違いを解説する

■コラム:「なんかこう、ぐにゃっとしたやつ」で検索できる時代

TODO: 抽象的・感覚的な表現で AI に検索させる面白い事例を集めて紹介する

付録 A

ハッカソン参加・運営チェックリスト

本編で語られたノウハウを、すぐに使えるチェックリスト形式でまとめました。ハッカソンに参加する方も、運営する方も、この付録を手元に置いてご活用ください。

A.1　参加者向け——ハッカソン前日までに済ませておくこと

- 開発環境のセットアップと動作確認
- 使用予定の API キーの取得・設定
- Git 環境の準備（アカウント、SSH 鍵、基本操作の確認）
- AI ツール（チャット型 AI、コーディングアシスタント）のアカウント準備
- チームコミュニケーションツールの確認（Slack, Discord 等）
- デモ用のプレゼンツールの準備
- 当日の持ち物リスト確認（PC、充電器、変換アダプタ、延長コード等）

TODO: 各項目の詳細な手順を追記する

A.2　参加者向け——当日の持ち物とセットアップ手順

- ノート PC（フル充電済み）
- 充電器・電源タップ
- ディスプレイ変換アダプタ
- イヤホン・ヘッドセット
- 名刺（あれば）
- 飲み物・軽食

TODO: オンライン参加の場合のセットアップ手順も追記する

A.3　運営者向け——開催 2 週間前からの準備タイムライン

- **2 週間前**：会場確保、テーマ決定、参加者募集開始

- **1 週間前**：審査基準の策定、審査員への依頼、賞品手配
- **3 日前**：タイムテーブル最終確認、参加者への事前案内送付
- **前日**：会場設営、Wi-Fi・電源の動作確認、デモ環境テスト
- **当日朝**：受付準備、チーム分け、ルール説明資料の最終確認

TODO: 各項目の詳細とチェックポイントを追記する

A.4　運営者向け——当日のタイムテーブルテンプレート

1 日型ハッカソン（8 時間）の例

- 09:00 - 09:30　受付・チーム分け
- 09:30 - 10:00　オープニング・テーマ発表・ルール説明
- 10:00 - 11:00　アイデア出し・チーム内合意
- 11:00 - 17:00　開発タイム（昼食休憩含む）
- 17:00 - 17:30　発表準備・デモテスト
- 17:30 - 18:30　発表・デモ
- 18:30 - 19:00　審査・結果発表・クロージング

即興型ハッカソン（5 時間）の例

- 18:00 - 18:15　テーマ共有・ルール説明
- 18:15 - 18:45　アイデア出し
- 18:45 - 22:00　開発タイム
- 22:00 - 22:15　デモテスト
- 22:15 - 22:45　発表
- 22:45 - 23:00　投票・結果発表

TODO: 各時間帯の運営側の動きと注意点を追記する

A.5　AI 開発環境のセットアップ——最低限入れておくべきツールと設定

- チャット型 AI ツール（ブレスト・調査用）
- コーディングアシスタント（IDE 統合型）
- バージョン管理（Git + GitHub/GitLab）
- CI/CD 環境（GitHub Actions 等）
- API リファレンスのブックマーク

TODO: 各ツールの具体的な設定手順とおすすめの構成を追記する

あとがき

この本で伝えたかったこと—— AI 時代に変わるものと変わらないもの

この本は、2026 年 2 月のブッカソンで生まれました。

3 人で集まって、ハッカソンや AI について思いの丈を語り合い、その対話を本という形にまとめる——まさにハッカソンのフレームワークを使って本を書くという実験でもありました。

Kuu：AI で何でも作れるからこそ、手を動かすことが大事になってきています。

Remio：作りたいものを作ろう。

ツールは変わり続けます。でも「作りたい」という衝動は変わりません。AI の時代だからこそ、限られた時間で人と協働して何かを作る「ハッカソン」の価値は高まっています。

TODO: テクノロジーが変化し続ける中で、「作りたい」という人間の衝動が持つ普遍的な価値を述べる

TODO: AI の時代だからこそ、限られた時間で人と協働して何かを作る「ハッカソン」の価値が高まっていることを論じる

手を動かし、人と語り、形にする——その繰り返しが未来を作る。

AI が生成する文章と、人間の生の声から生まれる文章。その両方を活かしながら、この本は作られています。読者の皆さんに、私たちの熱量が少しでも伝わっていれば幸いです。

ハッカソンで会いましょう。

著者紹介

Kuu

TODO: 著者紹介を記入してください

Remio

TODO: 著者紹介を記入してください

AI 時代のハッカソン

3 人のエンジニアが語る、AI と共創するものづくりの実践知

2026 年 2 月 7 日　初版第 1 刷　発行

著　者　Kuu、Remio