

Sun*

AIでアプリを高速開発

HEART Development

生成AIを活用したSun*の次世代開発フロー

素早くアプリを
リリースしたい

ユーザー検証
を速くしたい

1. 背景・目的
2. HEART Developmentの概要
3. HEART Developmentの進め方
4. FAQ

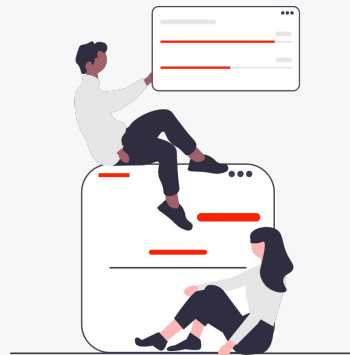
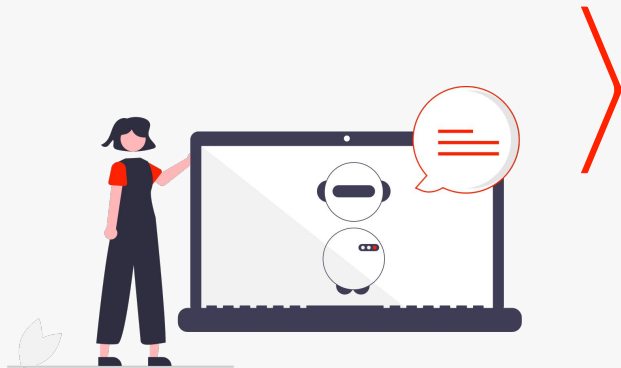
背景

背景

アプリケーション生成ができる生成AIツールが普及しており、誰もが短期間で安価にアプリ開発できる世界が到来しつつあります。

一方で、「AIで素早く作れる」ことの便利さのあまり、創作者の自己完結型のアプリケーション開発になりユーザー視点での提供価値の検証が後回しになりがちです。

そこで我々は、AI時代にもユーザー中心の価値創造ができるよう、AIを最大限活用しつつユーザーのフィードバックを得ながらの迅速なプロダクト改善によってユーザーにとって価値あるプロダクトを創るための新しいワークフローを作りました。

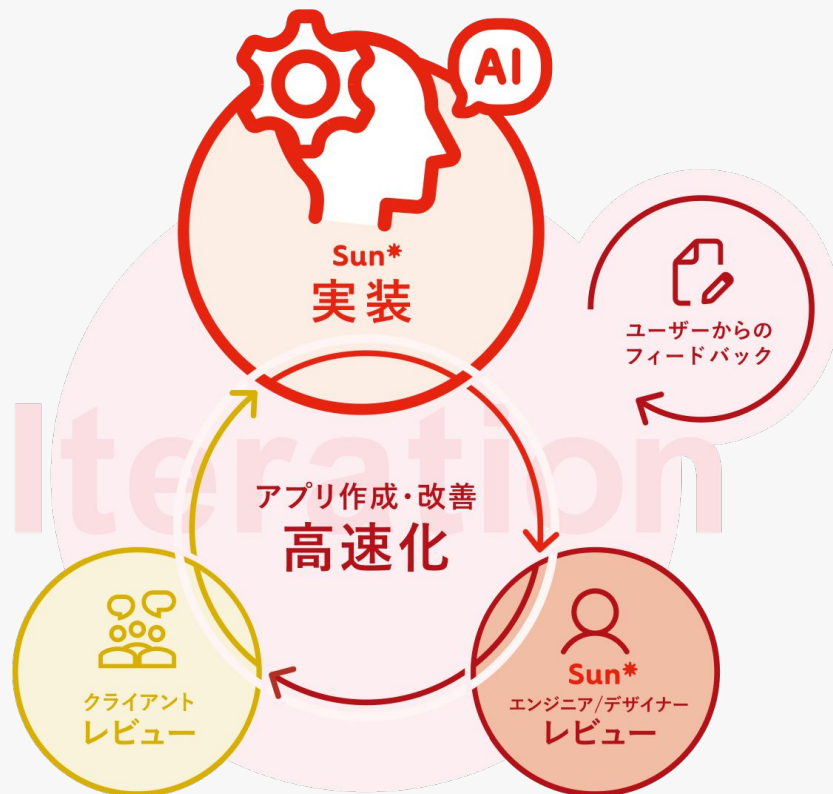


HEART Development の概要

HEART Developmentとは

生成AIを活用した 開発フローで アプリ作成・改善を高速化

生成AIの活用により、
要求整理～アプリリリース～アプリ改善を高速化。
ユーザーにとって価値あるプロダクトを
高速、安価、高品質で作成します。



AIによるHuman-centered design

AIを活用し、迅速なプロダクト開発と改善を通じてユーザー中心の価値を実現

1. Human-centered by Experiential and Rapid iteration

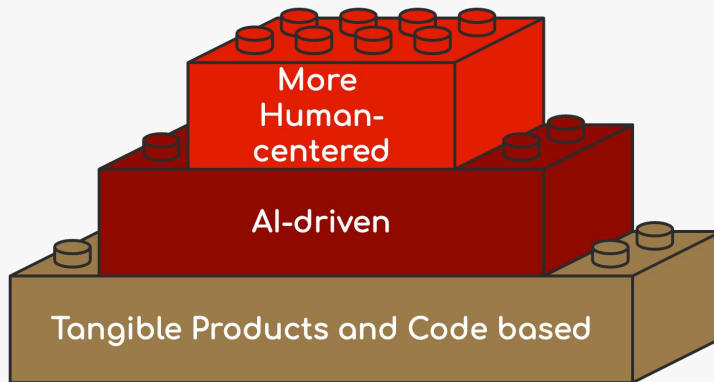
価値あるプロダクトを創るには、ユーザー体験から得たフィードバックを基に、迅速な改善とイテレーションを繰り返し、インサイトの精度を高めることが重要です。

2. AI-driven

AIにより開発が高速化したことで、こうした進め方がよりやりやすくなりました。すなわちAIの活用は目的ではなく、ユーザーに価値を届けるための有効な手段です。

3. Tangible Products and Code based

コードをAIとの共通言語として使うことで開発スピードを大幅に速めます。また実際に動くプロダクトを人間同士のコミュニケーションに使います。



HEART Developmentの5つの要素

Human-centered

ユーザーにとって価値あるプロダクトを創る

Experiential

実際の体験を通じて素早くインサイトを得る

AI-driven

AIを活用して開発を加速させる

Rapid Iteration

素早いフィードバックと改善のサイクル

Tangible Products and Code based

動くプロダクトとコードを中心としたコミュニケーション

HEART Development のメリット

As-Is

クライアントと
要求すり合わせに時間が
かかってしまう

ユーザーの声を
取り入れるのに
時間がかかる

デザインと実装が
乖離してしまう



HEARTの
提供価値

動くものがあることで
り合わせが早くなる

AIによる実装高速化で
素早いPDCAサイクルが
回せる

コードベースで進めるこ
とでズレがなくなる

HEART Development の進め方

HEART Development の進め方

HEART Developmentは5つのステップに分かれています。

AIを活用しながら、Business/Technology/Creativeのプロフェッショナルが、新規事業を包括的な視点でサポートします。



想定アウトプット

Value Design Syntax
リサーチ結果

要求定義書
画面一覧
機能一覧
DB設計

マイクロアプリ

アプリ



01 リサーチ・コンセプトメイキング

Sun*の新規事業開発において活用しているValue Design Syntax等を用いながら、ユーザーリサーチやマーケティングリサーチを行い、サービスコンセプト（やある場合は検証論点）を具体化して認識を合わせます。

サービスコンセプト		競争戦略		利益機軸
マクロ	ミクロ	競争優位性	仕組み	持続戦略
競争優位性のために		顧客が感じる	事業継続により	顧客が感じる
主要成功要因となる仕組み		知的資産 (知能データ顧客基盤)	知識	課金対象となる価値
対して		を	が重視し	に対しての
選ばれる理由		自社リソース	さらなる競争優位性	課金モデル (買切りの手数料/サブスクリプション等)
と		と	が強化されることで事業の持続性が高まる	による収入があり
選ばれる理由		パートナーリソース	により実現し	コスト (実働費/固定費)
で競争優位性を作る。		提供手段	を通じて届ける。	が主支出となり
実現可能な手法				単月黒字/リクープまでの期間
を獲って				までの資金繰りの算段がついている
を獲って				
最大公約数の価値				
超具体的価値				
を得られる。				
を得られる。				

Value Design Syntax

02 要求定義

必要な機能などの要求事項を言語化します。コンセプトをAIに入力することで、マイクロアプリ作成に必要な情報として機能要求や機能スコープなどを簡単に出力することができます。

通帳アプリプロダクト要求定義書

概要

本プロジェクトは、「通帳アプリ」と称するモバイルアプリケーションの開発を目的としています。このアプリは、ユーザーが自身の銀行口座の振込および残高を簡単に確認できる機能と、迅速かつ安全に送金を行う機能を提供します。ユーザーエクスペリエンスを重視し、直感的なインターフェースと高いセキュリティ基準を確保することで、日常の金融取引をサポートします。

背景

近年、スマートフォンの普及に伴い、モバイルバンキングの需要が急速に増加しています。従来の物理的な通帳に代わるデジタルソリューションへの移行が進む中で、ユーザーはより便利でアクセスしやすい金融サービスを求めています。特に、リアルタイムでの残高確認や暗証番号、迅速な送金機能は、多忙な現代人にとって不可欠な要素となっています。本アプリは、これらのニーズに応えることで、ユーザーの金融管理を効率化し、銀行サービスのデジタルトランスフォーメーションを促進します。

対象ユーザー

- 個人ユーザー：日常的に銀行取引を行う一般消費者。スマートフォンを利用して銀行サービスを利用することに慣れているユーザー。
- ビジネスユーザー：小規模事業者やフリーランス。迅速な送金や資金管理が求められるユーザー。
- 高齢者：結果や直感的な操作性を求める高齢層のユーザー。セキュリティに対する高い信頼性を必要とする。

要求定義書

リサーチ・コンセプトメイキング

要求定義

03 マイクロアプリ出力

UI/UXアップデート

ユーザーフィードバックからのアプリ改善

サービスデザイン

MVP作成

提供価値の検証・改善

03 マイクロアプリ出力

AIに要求を入力してマイクロアプリ[※]を出力し、MVPの骨格を作ります。動くプロダクトを作ること、ステークホルダー間で必要な要件が整っているか分かりやすい形でクイックに確認します。

※マイクロアプリ=動くワイヤーフレームのような、スタイルは作り込まれていないものの必要な骨格が揃ったアプリ

出力したマイクロアプリ

要求事項をAIで即座にアプリ化



リサーチ・コンセプトメイキング

要求定義

マイクロアプリ出力

04 UI/UXアップデート

ユーザーフィードバックからのアプリ改善

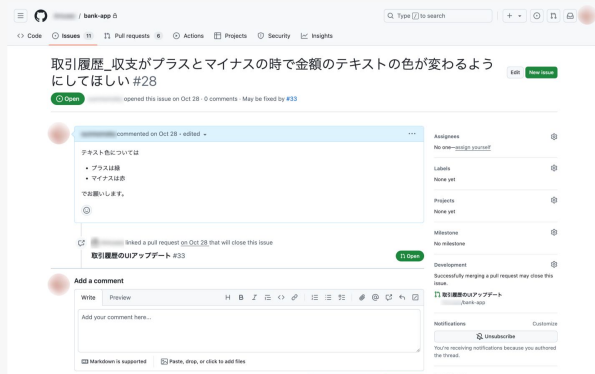
サービスデザイン

MVP作成

提供価値の検証・改善

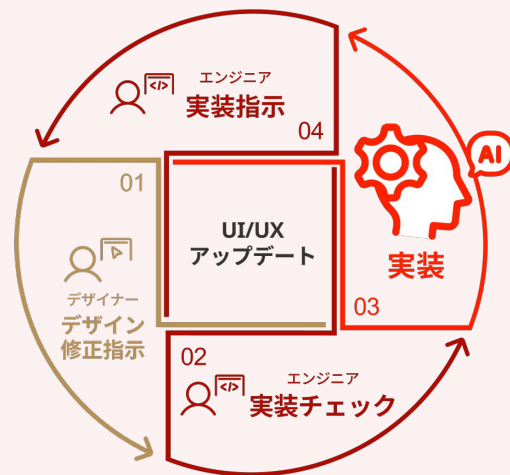
04 UI/UXアップデート

AIが出力したマイクロアプリをデザイナーがUI/UX観点で確認し、修正事項として起票。その修正事項をエンジニアが実装指示としてAIに伝え、検証フェーズまでの期間を早めることができます。



Githubによるデザイン修正指示

高速にUI/UXをアップデート



リサーチ・コンセプトメイキング

要求定義

マイクロアプリ出力

UI/UXアップデート

05 ユーザーフィードバックからのアプリ改善

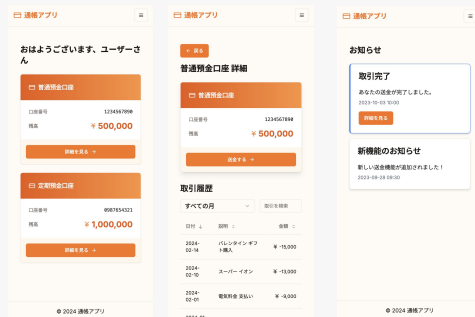
サービスデザイン

MVP作成

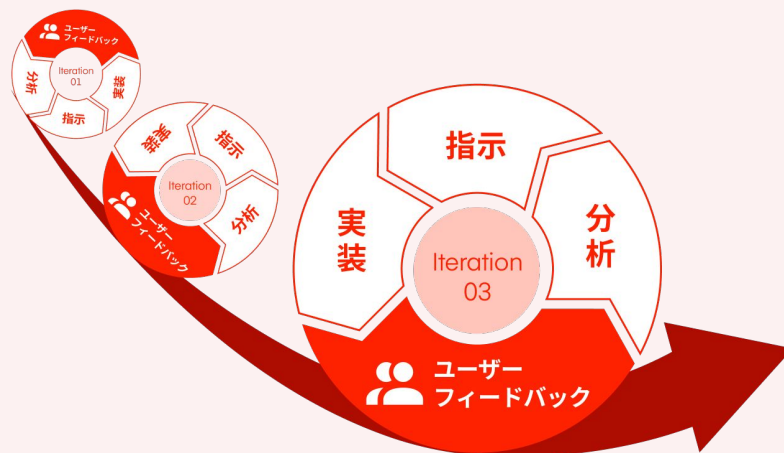
提供価値の検証・改善

05 ユーザーフィードバックからのアプリ改善

ユーザーからのフィードバックを得て、プロダクトの継続的な機能改善をしていきます。生成AIの高速な開発力、デザイナーのユーザー中心設計、そしてエンジニアのスムーズな連携により、ユーザーからのフィードバックを効果的かつ迅速にプロダクトに反映することができます。



ユーザーの声を取り入れ効果的にプロダクト改善



具体事例

通帳アプリの試作事例

メンバー : エンジニア1名 / デザイナー2名

UI/UXレビュー

アプリ

ごさいます、ユーザーさん

□ 座 号 1234567890

预 高 ¥ 500.000

残高 ￥ 500,000

☐ 定期預金口座

[詳細を見る →](#)

2週間

UI/UXレビュー

役割



デザイナー
デザイン確認
Issuesに修正指示を記載



エンジニア
実装指示



AI
実装

繰り返す

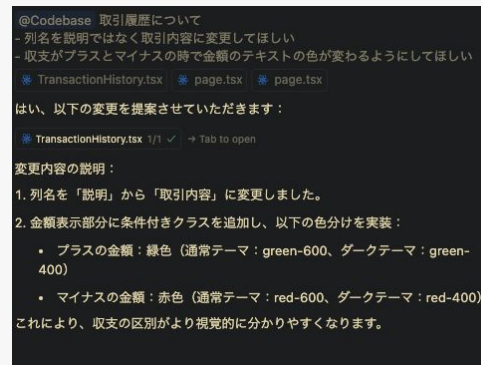
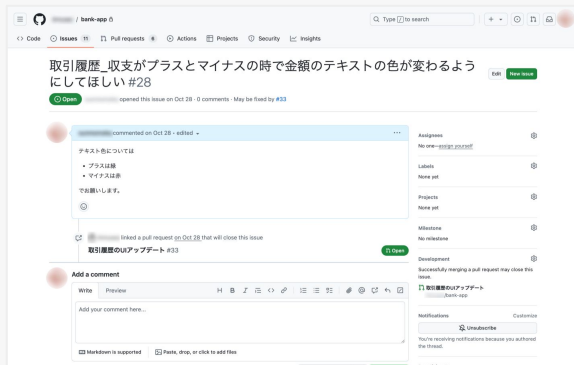
マイクロアプリ

デザイン確認・修正指示



AIに指示して実装

ステップ詳細



GitHub へ記入

CURSOR のAIで実装

FAQ

FAQ

Q なぜプロダクトベースの開発なのか？

A 早く作って改善できるからです。

プロダクトベースにすることで、動くものを早く作り、AIとBTCのメンバーが連携しながら迅速に改善できます。関係者が同じイメージを共有しやすく、ユーザーの意見も早く反映できます。これにより、方向性のズレやデザインと実装の乖離を防げます。

Q どんなプロジェクトに適している？

A スピードを重視するプロジェクトです。

新規開発プロジェクトや迅速な MVP 開発からの段階的な機能拡充を考えているプロジェクト、そしてユーザーフィードバックを重視するプロジェクトなど、開発のスピードが重視されるプロジェクトに適しています。

Q 品質は担保できますか？

A AIと人間の協働で高品質を担保します。

人間のエンジニアによるレビューと AI で生成するテストコードにより品質を担保します。さらに、ユーザーフィードバックと AI を活用した迅速な修正で、継続的に提供価値を向上させます。

Q 必要な体制・人員構成は？

A デザイナー1名、エンジニア1名、PM1名から始められます。

開発する規模に応じて、少ない体制からご支援可能です。お気軽にご相談ください。

FAQ

Q デザイナーの役割は変わりますか？

A はい。より創造的な役割になります。

ユーザー体験全体に注力し、AIを活用してアイデアを素早く形にしてユーザー検証を行い、サービスコンセプトの具体化と改善サイクルを主導するようになります。

Q なぜFigmaを使わないのですか？

A デザインと実装の齟齬を防ぐためです。

実装されたプロダクトをコミュニケーション手段として使用することで、デザインと実装の乖離がなくなり、早期にユーザーテストを開始できます。複雑なデザイン変更にはスポットでFigmaなどのデザインツールの活用も有効ですが、コードとの双方向変換の精度が低いため、AI時代には実装やフィードバックが遅くなるボトルネックとなる可能性があります。

Q デザイン検討しづらくないですか？

A 複数のデザイン案を作成比較できます。

Githubで複数のデザイン案を管理して比較することができます。また、細かいスタイルの調整はブラウザのデベロッパーツールを使うことですぐに確認できます。

Q デザインシステムはつくるの？

A エンジニアと協力して管理・運用します。

エンジニアと協力してコードを使いながら作り、運用していきます。まず、色やサイズなどのデザイントークンを決め、shadcn/uiをベースにコンポーネントを実装します。運用では、GitHubで変更を管理し、Storybookで部品を一覧表示して定期的に見直します。

FAQ

Q エンジニアの役割はどう変わりますか？

A デザイナーとAIの橋渡しをします

またこれまではテストコードで振る舞いを記述し、それを実装するということをしていたと思いますが、今後は自然言語で振る舞いを記述して、テストコード含めコードを生成するという仕事になるでしょう。

Q インフラ周りはどうしますか？

A 人間のエンジニアが責任を持ち進めます

人間のエンジニアがセキュリティ等最大限注意しながら設定します。AIも読めるフォーマットである terraformをAIに書かせ管理するのが有効です。データベースについては supabaseのAIを使うと簡単です。

Q ドキュメントは作らないんですか？

A リバースエンジニアリングで作ります

必要に応じて CursorやCopilot workspaceでcodebaseを参照することで生成できます。ただいつでも AIがコードを読んでその内容を理解できるためドキュメントの価値は薄れるでしょう。

Q どういうスキルが求められますか？

A AIと対話するスキルです

エンジニアはAIとの対話が重要になります。自分より優秀なエンジニアを部下に持ったとして上手に対話（#指示）をしながら進めるスキルが求められます。同時にデザイナーとの対話も重要で、UXデザイン領域に越境していくことも求められます。

Sun*

Thank you!