

Google Workspace拡張開発 の現在地

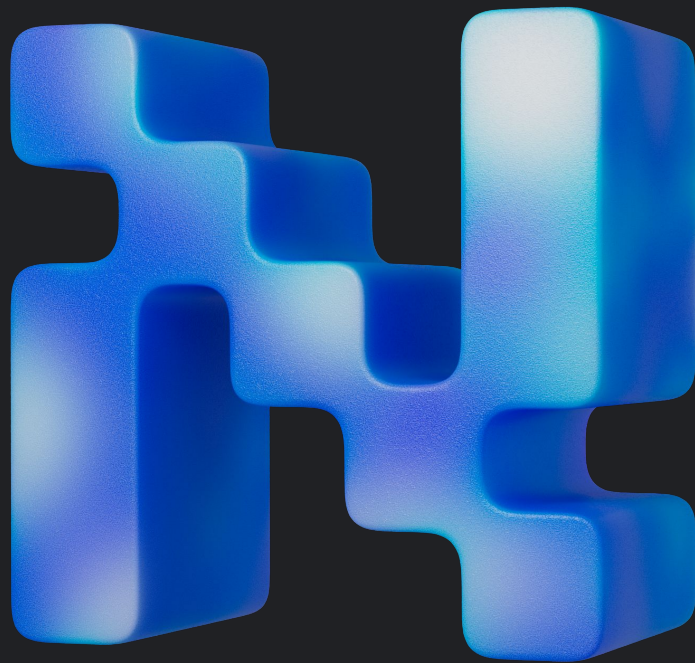
— "AI駆動開発"がもたらす変化と成果

Google
Cloud
Next

Tokyo



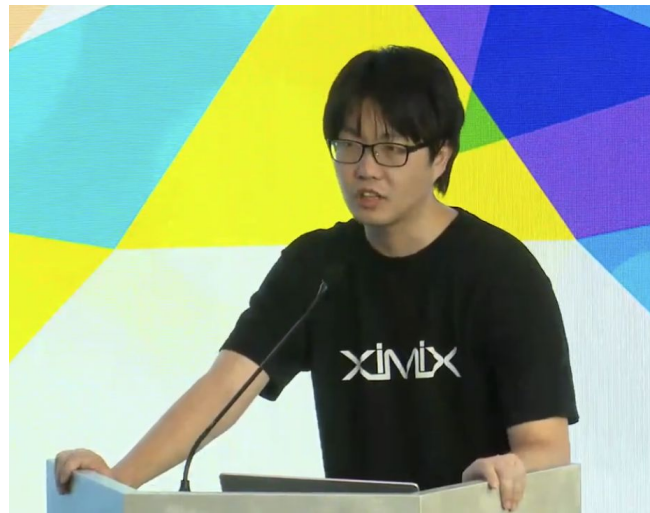
登壇者紹介





大岡 咲菜

NTTインテグレーション株式会社
データ&アナリティクス事業本部
クラウドインテグレーション部



北脇 孝太

NTTインテグレーション株式会社
データ&アナリティクス事業本部
クラウドインテグレーション部
主査

Agenda

- 01 Google Workspace の拡張開発について
- 02 拡張開発における「運用の課題」
- 03 生成AI を利用した拡張開発の取り組み
- 04 本日のまとめ
- 05 XIMIX のご紹介

01. Google Workspace の拡張開発について



Google Workspace 拡張開発の手段

	ノーコード開発	ローコード開発	フルスクラッチ開発
開発技術	AppSheet, Google フォーム, Google サイト など	GAS (Google Apps Script)	各種言語 (Python, Java など)
開発難易度	低: プログラミング不要	中: プログラムが必要	高: 高度なプログラミングが必要
開発例	セミナー参加者アンケート作成, ポータルサイト開発	フォーム回答時のお礼メール 自動送信	他システムと連動したアカウント・権 限の自動一括設定

Google Workspace 拡張開発の特徴

ノーコード/ローコード

現場完結・スピード最重視

エンジニアに頼らず、現場のビジネス担当者のみでアイデアを最速で具現化したいケースにおいて業務の即時改善と圧倒的なコストパフォーマンスを実現する

フルスクラッチ

大規模・複雑・高品質

品質・保守性・高度なシステム連携が要求される領域で、プラットフォームの制約に縛られず、企業の独自業務やセキュリティ要件を満たす

なぜ拡張開発が必要なのか



最大の理由は「統制とガバナンス」の強制

運用ルールと「人に依存するセキュリティ管理」の限界

- うっかりミスにより発生するインシデントのリスク
- 故意のルール破りにより発生するインシデントのリスク

→マニュアルをどれだけ整備しても、情報漏洩などのリスクを完全に防ぐことは困難

システムの強制力による統制

- 人の意識に頼らない仕組み

→システムがルールを強制するため、確実なガバナンスと安全な運用設計を担保

標準機能の安全性をベースに、自社のセキュリティポリシーを「**拡張開発**」でシステムが強制

(拡張開発事例) サイト作成申請



【開発手法】AppSheet + GAS

複数の運用の仕組みを短期間で
開発・展開する必要があり、
AppSheet + GAS を採用



【課題】誤公開・情報漏洩

制限なしの作成・公開による、
意図しない「組織全体」や
「インターネット」への漏洩リスク



【要望】勝手な公開の防止

ユーザーが勝手にGoogle サイト
を作成・公開するのを防止

サイト申請から払い出しまでの統制フロー

STEP 01

申請

ユーザーが作成申請の実行



STEP 02

承認

管理者による確認と承認プロセス



STEP 03

自動格納

GAS がGoogle サイトを、
指定の共有ドライブに格納



拡張開発がもたらした効果

拡張開発による効果

拡張開発なし

- ✗ 誰でも自由にGoogle サイトを作成
- ✗ 機密情報の誤った公開

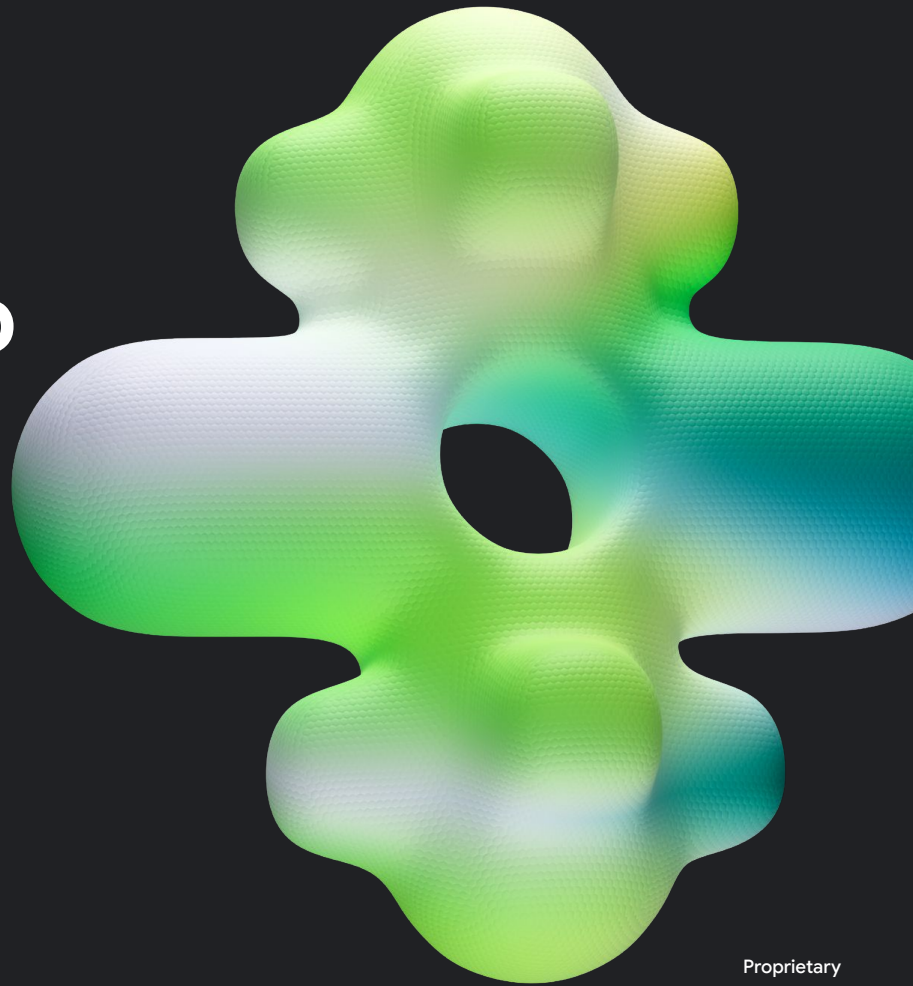


拡張開発あり

- ✓ 承認を経てサイトを作成
- ✓ 承認フローを通すため機密情報の誤公開を防止
- ✓ 「誰が」「いつ」申請・承認したか追跡可能

承認プロセスのシステム化で、情報漏洩を防ぎ強固なガバナンスを実現！

02. 拡張開発における 「運用の課題」



本セッションの概要

本セッションでは大規模開発で見たノーコード / ローコードでの「運用の課題」と、それを解消する「生成 AI+フルスクラッチ」という新たな選択肢について解説する

直面した3つの
「運用の課題」



「生成 AI+フルスクラッチ」の
登場



課題の解決と
その他手法との比較

拡張開発をすることで「統制とガバナンス」を強制！



しかし大規模かつ複雑な業務要件への適用で予見し得なかった**運用上の課題**に直面



生成AIによる新たな手法の登場

直面した3つの課題

大規模・複雑な業務への適用によって、開発時には予見し得なかった「3つの運用の課題」に直面した

リリース管理の難しさ

CI/CD によるリリースができない
本番環境への安全な公開・切り替え・
バージョン管理に手間がかかる

強引な設計による複雑化

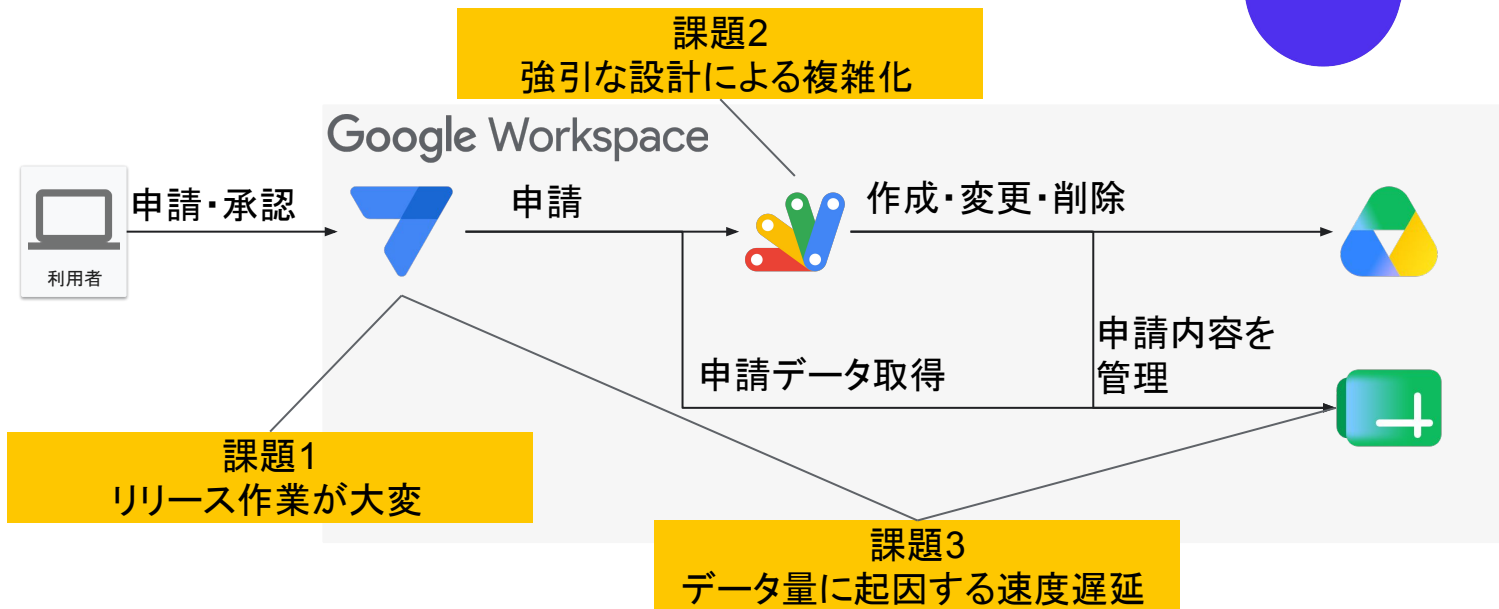
自由度の低いノーコードの枠組みに複雑な業務ロジックを無理に実装
不足機能を GAS で補うことになり、不具合の温床に

データ量による速度遅延

データ量が数万件となった場合
AppSheet上での表示速度がとても遅い
数ヶ月に1回程度メンテナンスが必要と
なってくる

(拡張開発事例) ワークフロー申請システム

AppSheet+GASで作成した、共有ドライブを作成・変更・削除するための WF申請システムをご紹介します



これらの課題を「**生成AI+フルスクラッチ**」で解決！

「生成 AI+フルスクラッチ」の登場

これまで「速く作れること」はノーコード / ローコードならではの強みだった。しかし生成 AIを活用すれば、フルスクラッチ開発でも同じスピードを出せるようになり、その差はなくなる

ノーコード / ローコード

✓ 開発スピードが早い

現場ニーズに素早く対応・低コストで開発可能

✗ 大規模・複雑な業務には限界

規模が大きくなるにつれ品質・保守性に課題が生じる

フルスクラッチ

✗ 開発スピードが遅い

ゼロから実装するため時間がかかる

✓ 自由度が高く複雑な業務にも対応可能

柔軟に設計ができるため複雑な要件にも対応可能

「生成AI+フルスクラッチ」の登場

これまで「速く作れること」はノーコード / ローコードならではの強みだった。しかし生成 AIを活用すれば、フルスクラッチ開発でも同じスピードを出せるようになり、その差はなくなる

ノーコード / ローコード

✓ 開発スピードが早い

現場ニーズに素早く対応・低コストで開発可能

✗ 大規模・複雑な業務には限界

規模が大きくなるにつれ品質・保守性に課題が生じる

生成AI+フルスクラッチ

✓ 開発スピードが早い

生成AIが素早く実装

✓ 自由度が高く複雑な業務にも対応可能

柔軟に設計ができるため複雑な要件にも対応可能

生成AIの登場により

スピード(ノーコード / ローコード)と**品質(フルスクラッチ)**のトレードオフが消滅

開発手法の比較

品質とスピードのトレードオフを解消し、QCD すべての項目で「生成 AI + フルスクラッチ」が最適な選択肢と言える

開発手法	Q: 品質・保守性	C: コスト	D: 納期・スピード	総合評価
A. フルスクラッチ	○ 高い自由度・堅牢な 基盤・高保守性	✕ 専門エンジニアのコ ストが比較的高	✕ ゼロから実装する ため時間がかかる	△ 費用と期間がネック
B. ノーコード/ローコード	✕ 大規模化で保守・リ リースが崩壊	◎ ライセンス費用 開発費も抑えめ	◎ ノーコードのため 爆速	△ 小規模なら可 大規模は危険
C. 生成 AI + フルスクラッチ	◎ A. と同じ高品質	○ 期間短縮で コスト削減	○ AI駆動で A. より 高速	○ B. より柔軟、A. より短 期間

「運用上の課題」を解決する

生成 AI のスピードとフルスクラッチ の自由度の掛け合わせで課題を解消する

課題 1

リリース管理の難しさ



✓ 標準的な開発基盤に乗せることができる

Git・CI/CD・バージョン管理をそのまま活用でき、本番環境への安全な公開・切り戻しが可能に

課題 2

強引な設計による複雑化



✓ 業務要件に合わせて自由に設計できる

プラットフォームの制約に縛られず、複雑な業務ロジックも素直に実装。増える実装量は生成 AI が肩代わり

課題 3

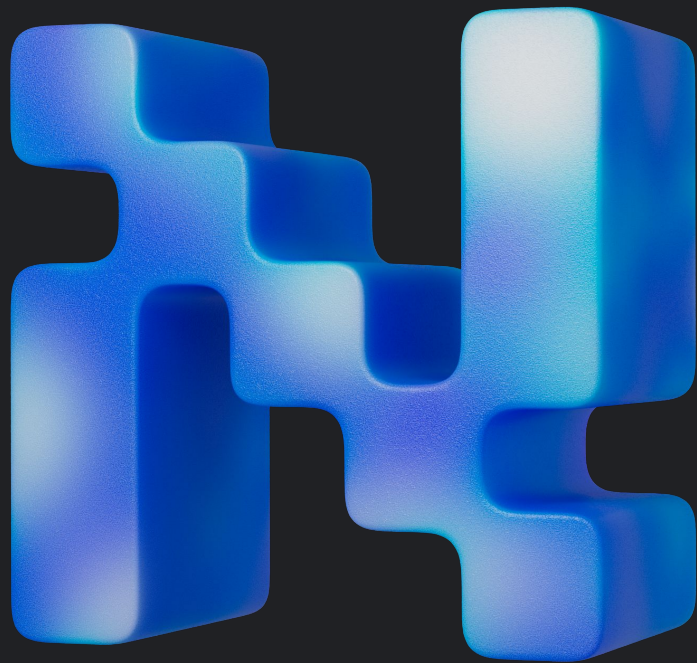
データ量による速度遅延



✓ ページネーションによる応答速度の維持

データ量が多い場合でも、ページネーション等により一括で取得することがないため応答速度をある程度担保できる

03. 生成AIを利用した 拡張開発の取り組み



生成AIを開発に導入する上での課題

技術的なメリットを最大化するためには、品質管理とプロセスの標準化が不可欠

品質の担保

生成AIは高速にドキュメント・コードを生成する一方、ハルシネーションや非効率な実装が混入するリスクがある。

属人化のリスク

AI活用開発には従来と異なるプロセス設計が必要。属人化を防ぐための標準化が不可欠であり、チーム全体で効果的に使いこなすための教育体制の構築が必須。

NI+Cは独自の **開発標準プラットフォーム** とノウハウでこれらの課題を解決し、
高品質・高スピード開発を実現

AIによる品質改善サイクル

構築だけでなく、AIによるレビューを導入することにより品質を向上させる。

生成AI時代の新たな開発標準

開発プロセスの標準化により、AI活用のノウハウを個人の経験に依存させず、チーム全体で高水準で活用できる仕組みを定着させる

NI+Cでの取り組みご紹介

要件定義から試験まで、システム開発の各工程を生成 AIを活用し効率化
人の役割は各工程での成果物レビューと生成 AIへの修正指示

要件定義

ドキュメント化

ヒアリング内容から要件定義書を自動ドラフト。迅速なレビューを支援。

システム試験項目書生成

要件に基づき、システム試験項目書をこの段階から自動生成。

設計

設計書生成

論理構造を整理し、合意形成のための詳細設計資料を自動生成。

結合試験項目書生成

内部インターフェース設計から結合試験項目をシームレスに出力。

製造

コード自動生成

設計書をインプットに、言語に最適なソースコードの雛形を生成。

品質レビュー

AIによる静的解析で脆弱性や非効率なロジックをリアルタイムに指摘。

試験

テストデータ生成

多様な境界値を網羅した試験データやスクリプトを自動作成。

不具合解析

実行ログやエラーから原因を特定し、修正案を即座に提示。

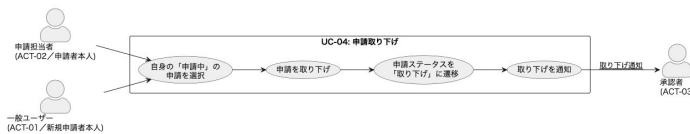
開発工程の全体を通して生成AIを活用し、「**開発効率**」の劇的な向上を実現します

実際のアウトプット

2.5 UC-04: 申請取り下げ

概要

- 申請者本人（新規申請の場合は一般ユーザー本人、その他は申請担当者本人）が、承認前の「申請中」ステータスの自身の申請を申請一覧から選択する
- 申請者本人が取り下げを実行し、システムが申請ステータスを「取り下げ」に遷移させる
- システムが承認者へ取り下げ通知を送信する



項目	内容
主アクター	申請担当者（ACT-02／申請者本人、新規申請の場合は一般ユーザー（ACT-01）本人）
トリガー	申請者本人が承認前（「申請中」ステータス）の自身の申請を取り下げたいとき
事前条件	申請ステータスが「申請中」であり、操作者が申請者本人である
事後条件	申請ステータスが「取り下げ」に遷移し、承認者に取り下げ通知が送信される

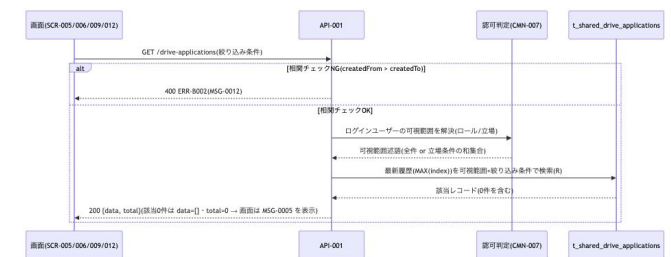
要件定義書

GWS管理ツール				
申請一覧				
+ 新規申請 絞り込み				
申請ID	ドライブ名	申請区分	申請者	承認者
mock-app-001	プロジェクト共有ドライブ	新規申請	user@example.com	approver@example.com
mock-app-002	営業部共有ドライブ	新規申請	sales@example.com	approver@example.com
mock-app-003	営業部共有ドライブ（変更）	変更申請	sales@example.com	approver@example.com
mock-app-004	旧プロジェクト共有ドライブ	削除申請	user@example.com	approver@example.com
mock-app-005	テスト用ドライブ	新規申請	user@example.com	-
ページあたりの行数: 10 1-5 of 5				

製造

2. 連携シーケンス

参加者はコンポーネント単位に留める。API内部の絞り込み手続の詳細は業務ルール決定表(3章)を正とする。単項目チェック(型・桁・enum・日付形式)はOpenAPI schemaが正でありここでは描かない。



認証エラー(X-User-Email 欠落・IAP セッション失効 → 401 / ERR-A002)は全API共通のミドルウェアで判定するため本シーケンスで

設計書

2.2.1 一覧表示			
テストケースID	テスト対象	入力値/条件	期待結果
IT-FE-010	SharedDrivePage (正常: ローディング中)	isLoading=true	CircularProgress が表示され、DataTable は表示されない
IT-FE-011	SharedDrivePage (正常: データあり)	useGetSharedDriveApplications がデータを返す	DataTable に申請一覧が表示される
IT-FE-012	SharedDrivePage (正常: データなし)	useGetSharedDriveApplications が空配列を返す	DataTable に行が表示されない
IT-FE-013	SharedDrivePage (異常: APIエラー)	error が truthy	「データの取得に失敗しました」メッセージが表示される
IT-FE-014	SharedDrivePage (正常: sessionStorage からメールアドレス取得)	sessionStorage に 'user_email' = 'user@test.com' が設定済み	userEmail が 'user@test.com' として取得される (isApproverMode 判定に使用)
IT-FE-015	SharedDrivePage (正常: sessionStorage 未設定)	sessionStorage に 'user_email' が存在しない	userEmail が "" になる (isApproverMode=false)

2.2.2 申請詳細ドロー (G-105)			
テストケースID	テスト対象	入力値/条件	期待結果
IT-FE-020	SharedDrivePage (申請行クリック → ドロー表示)	申請行をクリック	selectedApplicationId が設定され、ドローが開く (drawerOpen=true)
IT-FE-021	SharedDrivePage (ドロー: 詳細ローディング中)	申請行をクリック後、useGetSharedDriveApplication が isLoading=true	ドロー内で CircularProgress が表示される
IT-FE-022	SharedDrivePage (ドロー: 詳細データ表示)	useGetSharedDriveApplication がデータを返す	ドロー内で申請詳細が表示される (申請ID、ステータス、ドライブ名等)
IT-FE-023	SharedDrivePage (ドロー閉じる)	ドローの閉じるボタンをクリック	drawerOpen=false、selectedApplicationId=null になる

試験項目書

導入効果と残された課題

導入効果

・ 開発標準化による品質の担保

独自の開発標準プラットフォームにより品質管理を仕組み化

属人化を排除し、誰がやっても同じ品質・安定した開発フローを実現

・ 各工程の自動化による劇的な効率向上

要件定義から試験まで一連の工程に生成 AI を組み込むことで、ドキュメント・コード・試験データの生成スピードを高めた

残された課題

・ 非エンジニアの壁

生成 AI の支援があっても、実用的なシステムの構築には設計知識とコード理解、インフラ知識が依然必要

「誰でも開発できる」状態にはまだ届いていない

・ 継続的な開発標準のアップデート

進化し続ける生成 AI 技術や新たなユースケースに対応するため、標準化ルールやプロンプトを常に最新化し続ける運用が必要

04. 本日のまとめ



まとめ

◎ 開発手法の 4つの選択肢



ノーコード

- ・プログラム不要
- ・手軽さとスピード重視



ローコード

- ・最小限のコードで構築
- ・拡張性と手軽さを両立



スクラッチ

- ・ゼロから自由に設計
- ・高い柔軟性と堅牢性



新たな選択肢

生成AI +スクラッチ

- ・AI が実装やテストを強力補助
- ・自由度とスピードを両立

◎ 意思決定と次のステップ



用途や目的に応じた『取捨選択』

用途や開発目的に合わせて最適な手法を適切に選定する、従来の確かな意思決定プロセスが必要

ブースにて詳細をご案内

生成AI の活用に向けた具体的な準備や実践ステップの詳細は、ぜひ当社展示ブースへお越しください！

05. XIMIX のご紹介



What's XIMIX?

デジタルとリアルの境界が融合していく現在において、
人々の生活を豊かにするためのソリューションを提供するために作られたNI+Cのブランド



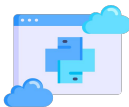
未知の可能性や未来を表す「X」
私たち自身を示す「I」

デジタル、リアル、人、技術、経験、スピード…
あらゆる価値を「MIX」し、想像を超えるインパクトを。

誠実や安心な印象を持たせる symmetry のロゴに込めたのは、
揺るぎない誠実さと、お客様へのコミットメント。

私たちは【XIMIX】は、常に新しく未知の価値を追求し続けます。

事業領域



アプリケーション開発



インフラ構築



BigData & ML支援



監視 / マネジメント支援



導入後のサポート



グループウェア移行

拡張開発で「XIMIX」が提供する価値

ローコード/ノーコード

「AppSheet + GAS 開発」を安全に推進したい
企業様へ

『ガバナンス設計・教育トレーニング支援』

- 安全で統制の取れた運用体制の構築
- 野良アプリ化・ブラックボックス化の防止
- 教育とトレーニングによる内製化推進

生成AI + フルスクラッチ

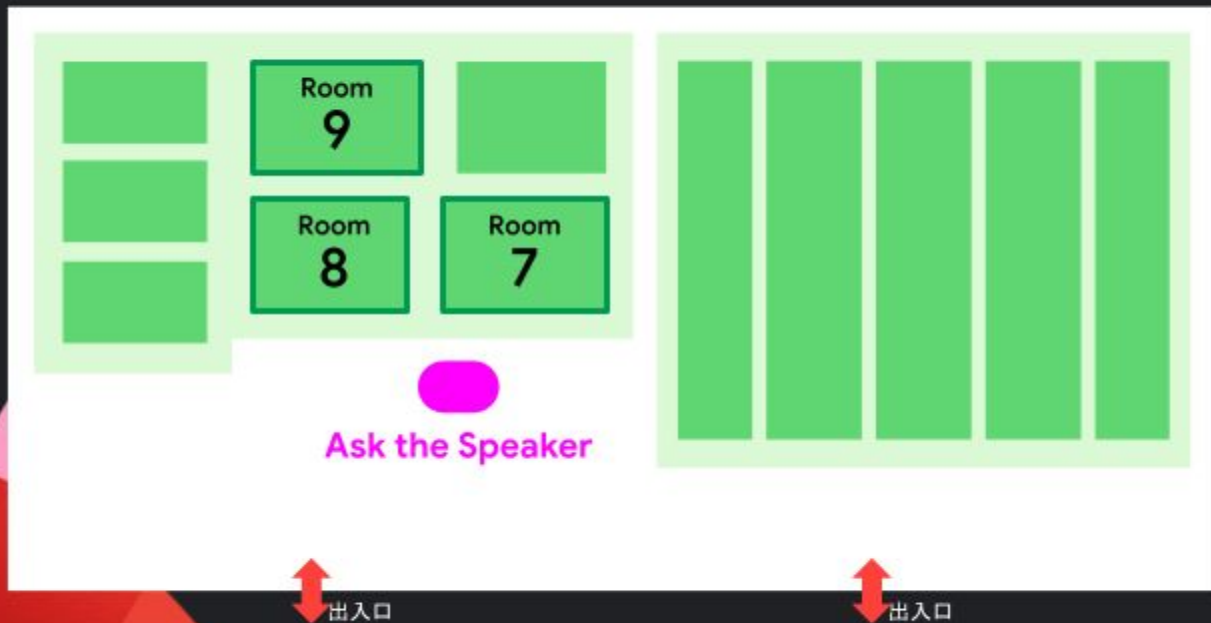
高品質かつ本格的なプロ開発を求める
企業様へ

『一気通貫のDX支援』 (開発標準プラットフォーム)

- 生成AI × フルスクラッチ開発の融合
- 圧倒的な「高スピード・低コスト」
- エンタープライズ基準の高品質保証

Ask the Speaker にぜひお越しください(会場は 4F)

セッションに関する質問にスピーカーが直接お答えします！



Thank you

Thank you

Google
Cloud
Next

Tokyo

