



Citadel AI

生成AIを用いる サービス開発の原則

株式会社 Citadel AI

杉山 阿聖

© 2021-2025 Citadel AI Inc.

主旨

- 開発者との交流を通じて見えてきた生成AI活用の原則と、残されている課題について共有します

TOC

- **DevOps, MLOps, LLMOps <-**
- 生成AIを利用するシステム開発の原則
- 今後の展望

DevOps, MLOps, LLMOps

- DevOps
- MLOps
- LLMOps

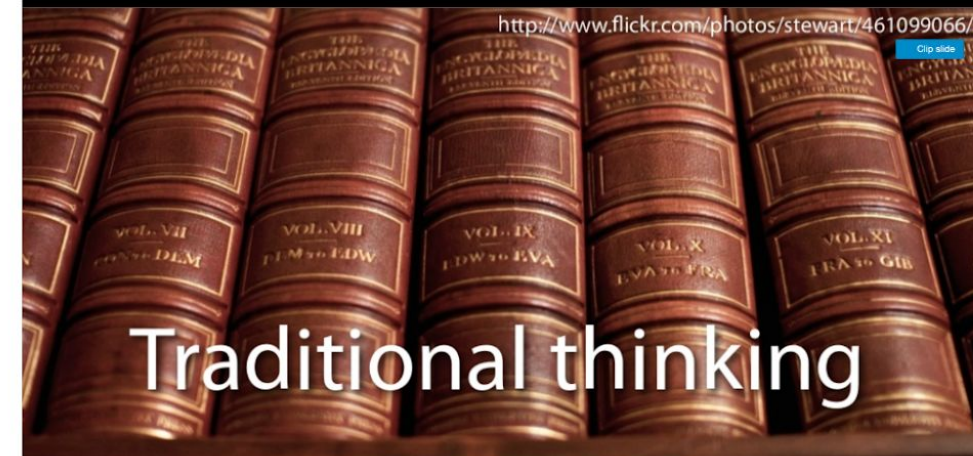
DevOps: Dev VS Ops

- Dev: 顧客に新しい価値を早く提供したい、多少不安定になるかもしれないが運用が頑張れば良い
- Ops: 顧客に安定的に価値を提供したい、新機能の追加で不安定になることは受け入れられない

10+ Deploys Per Day: Dev and Ops Cooperation at Flickr - Slideshare

<https://www.slideshare.net/jallspaw/10-deploys-per-day-dev-and-ops-cooperation-at-flickr>

Dev **versus** Ops



Traditional thinking

Dev's job is to add new features
Ops' job is to keep the site stable and fast

DevOps: Dev ♥ Ops

- Dev vs Ops から Dev & Ops に移行しようという提案 (2008)
- 「顧客に価値をすばやく安定的に提供しよう」という提案
- この提案に基づくのが DevOps
- DevOps: Dev と Ops の協調

Lowering risk of change
through **tools** and **culture**



Dev and Ops

DevOps: 継続的改善

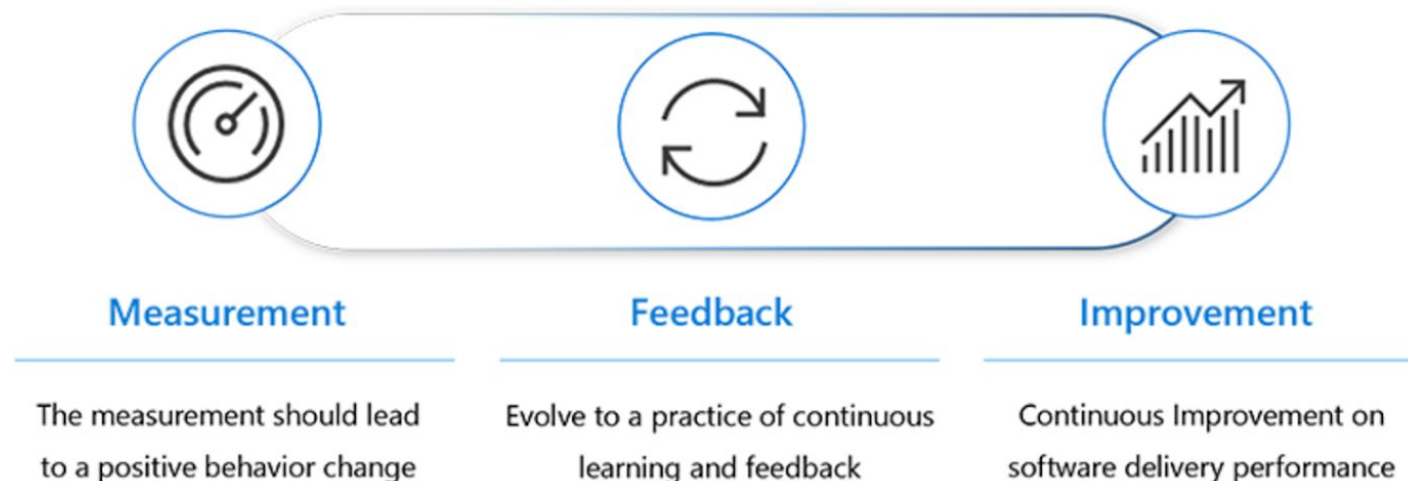
- DevOps の原則のひとつ
- フィードバックサイクルによる改善

What is Continuous Improvement?

Continuously and candidly observing your DevOps process allows teams to identify possible improvement points.

All improvement requires change, but not all change is improvement. This is why measurement is a critical success factor to organizations using DevOps. As Peter Drucker says, "If you can't measure it, you can't improve it."

The lack of an effective feedback mechanism makes it difficult to improve the impact of apps on business. That's why it's important to create an environment that fosters a learning-centric approach for DevOps Improvement, with a focus on making data-based adjustments.



Explore Continuous Improvement - Training |
Microsoft Learn <https://learn.microsoft.com/en-us/training/modules/characterize-devops-continuous-collaboration-improvement/3-explore-continuous-improvement>

MLOps: MLOps とは

- 機械学習の成果をスケールさせるためのさまざまな取り組み
- DevOps のプラクティスに基づく
- 機械学習システムを育てる取り組み



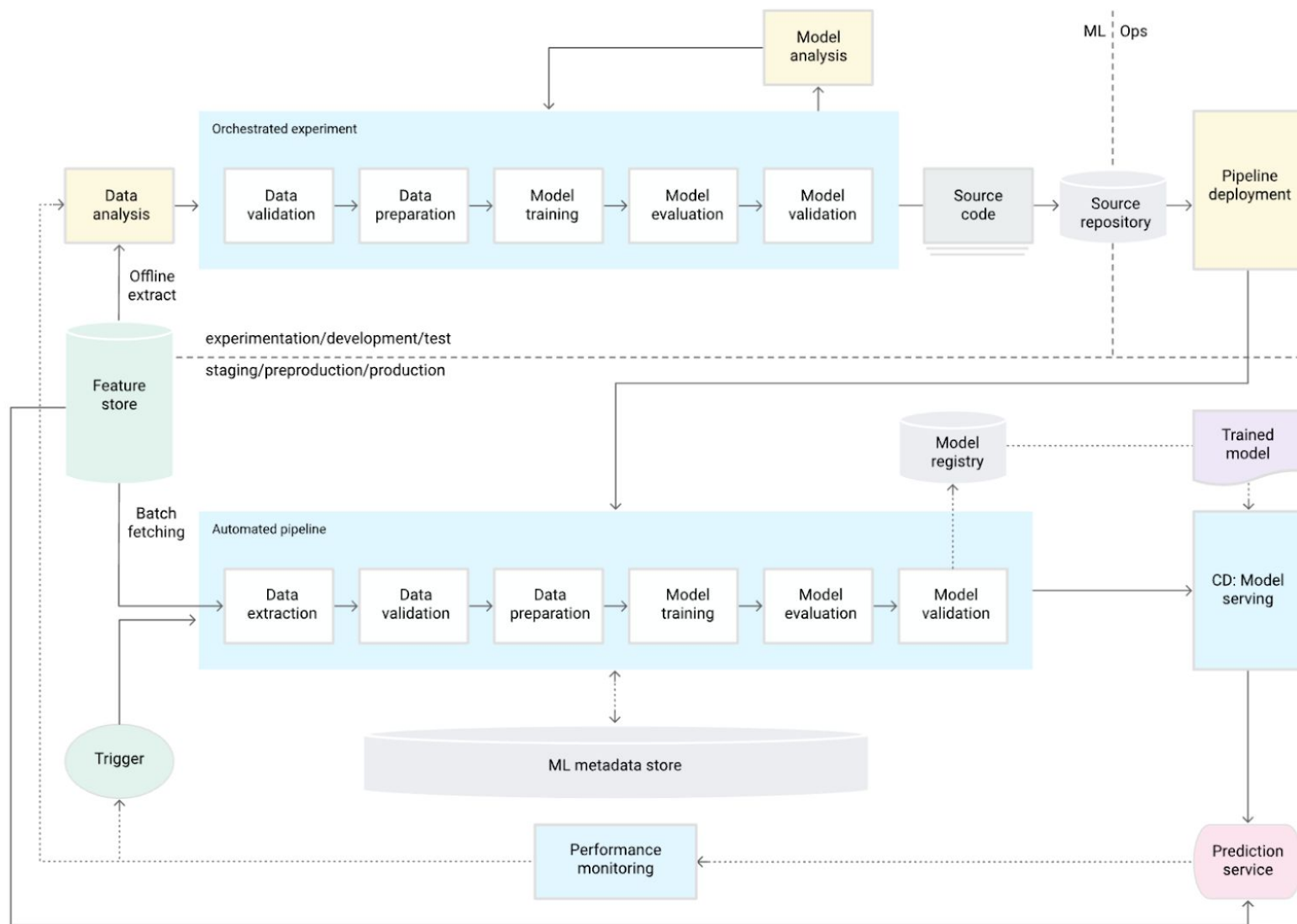
ML and Ops

MLOps: 継続的な訓練

- MLOps における継続的な改善の実装
- モデルを継続的に訓練して改善

MLOps: Continuous delivery and automation pipelines in machine learning | Cloud Architecture Center | Google Cloud

<https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning>



LLMOps: Demo hell

- デモまでは行き着くものの、本番化が著しく困難
- 品質を評価し、担保することが極めて困難

Escaping AI Demo Hell: Why Eval-Driven Development Is Your Path To Production

<https://www.forbes.com/councils/forbestechcouncil/2025/04/04/escaping-ai-demo-hell-why-eval-driven-development-is-your-path-to-production/>

Escaping AI Demo Hell: Why Eval-Driven Development Is Your Path To Production



By Albert Lie, Forbes Councils Member.

for Forbes Technology Council, **COUNCIL POST** | Membership (fee-based)

Published Apr 04, 2025, 07:45am EDT

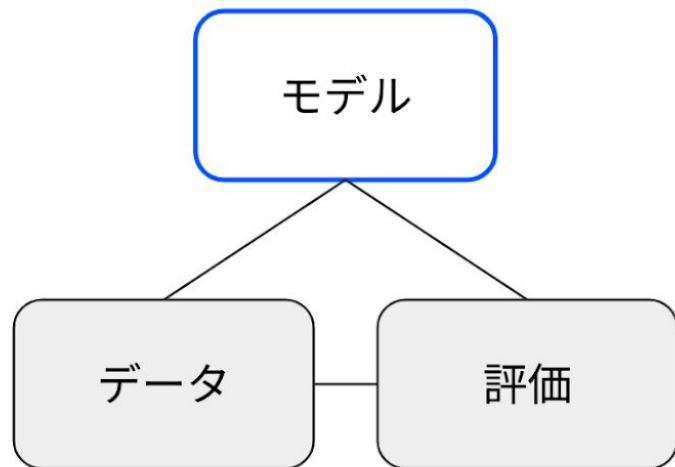
[Share](#) [Save](#)

Albert Lie, Co-founder and CTO at [Forward Labs](#), next-gen AI-driven freight intelligence for sales and operations.

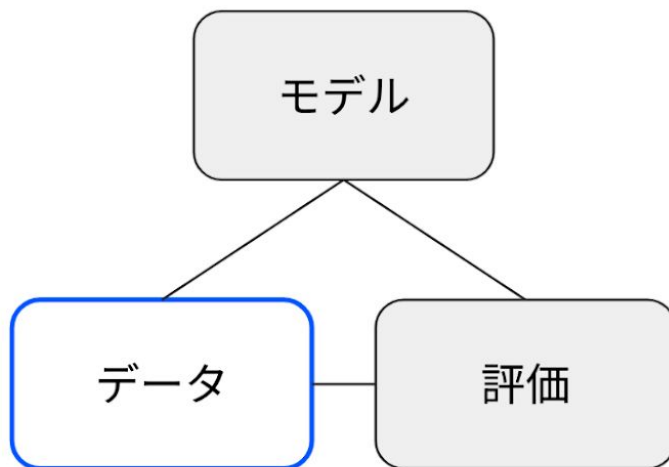


LLMOps: Eval-Centric AI

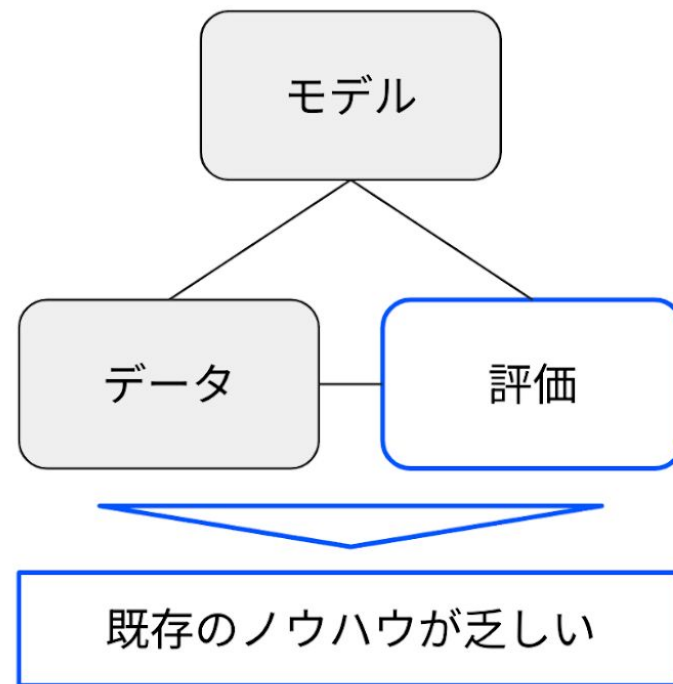
Model-Centric
モデルを改善



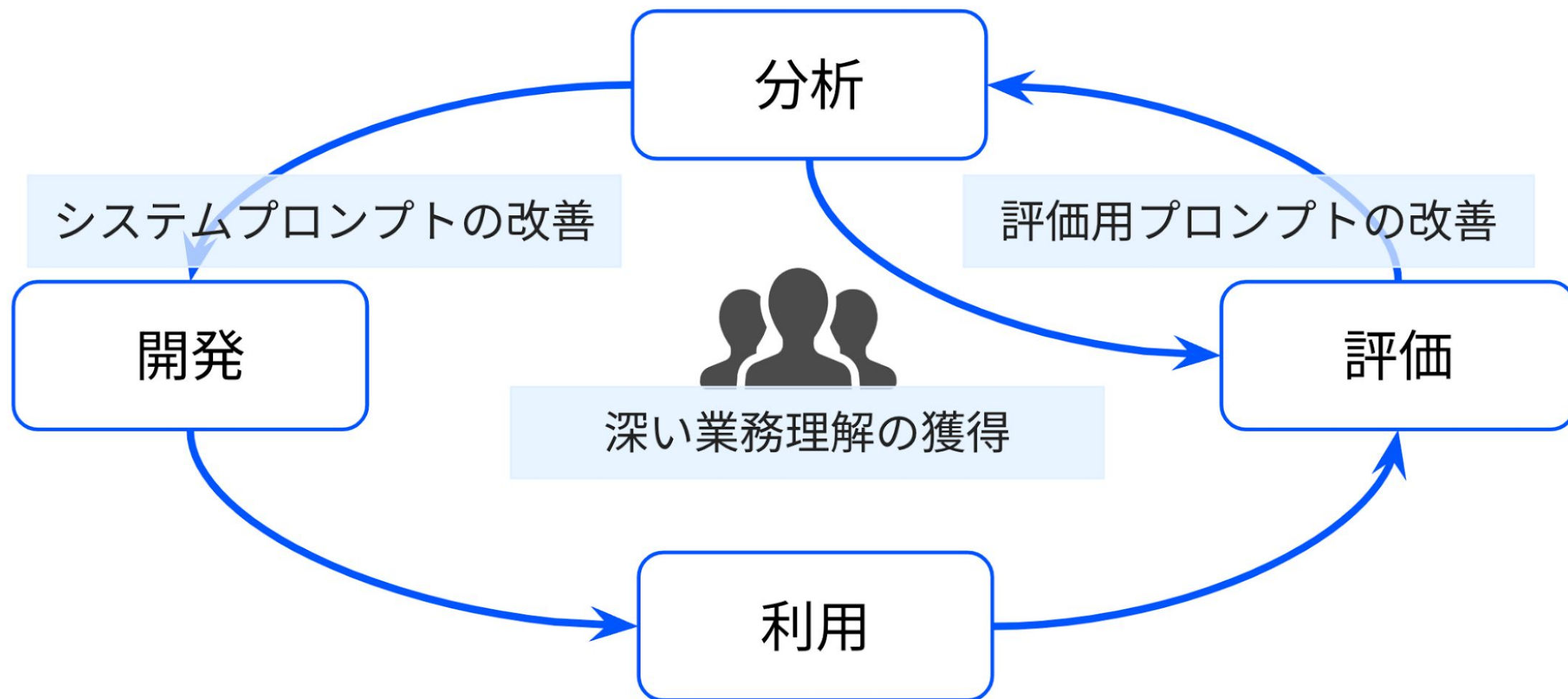
Data-Centric
データを改善



Eval-Centric
評価を改善



LLMOps: 継続的な評価による継続的な改善



TOC

- DevOps, MLOps, LLMOps
- **生成AIを利用するシステム開発の原則 <-**
- 今後の展望

生成AIを利用するシステム開発の原則

1. リスクと効果を考慮し小さく始める
2. 高速にプロトタイピングする
3. 独自のデータを定義し評価データを育てる
4. 専門家を開発チームの一員にする
5. 本番環境でテストする

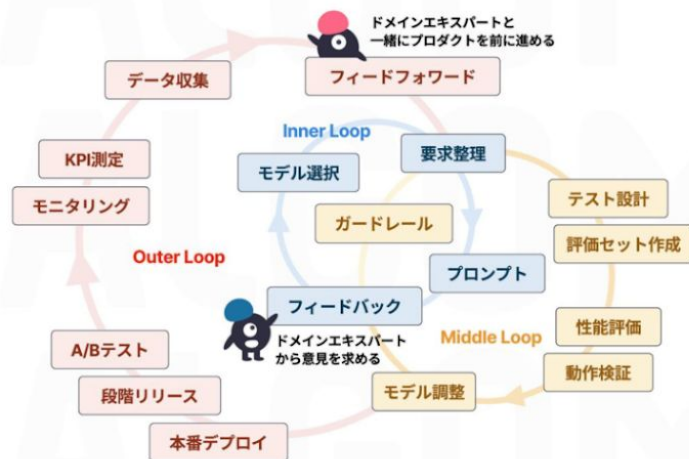
1. リスクと効果を考慮し小さく始める

- ユースケースを安全性と効果の2軸で分類
 - 安全性: サービス提供、人にフォールバック、対応不可
 - 効果: システム化が進んでいない、人の経験や勘に頼っている、などで判断
- 安全かつ効果の高いユースケースを特定し推進する

2. 高速にプロトタイピングする

- 専門家も自分の行っていること・やりたいことを明確にできない
- 要件定義よりもプロトタイプを優先する
- 手戻りを恐れるのではなくイテレーションを回す

AIエージェントの地上戦 ～開発計画と運用実践 / 2025/04/08 Findy ランチセッション #19 <https://speakerdeck.com/smiyawaki0820/08-findy-w-and-bmitoatupu-number-19>



Ito, Ogawa, Onabuta氏 - Step-by-Step MLOps and Microsoft Products
https://speakerdeck.com/shisyu_gaku/step-by-step-mlops-and-microsoft-products

Inner Loop

モデルの選択・プロンプト作成などをすばやく試行し、ドメインエキスパートとペアリングセッションを行う。

Middle Loop

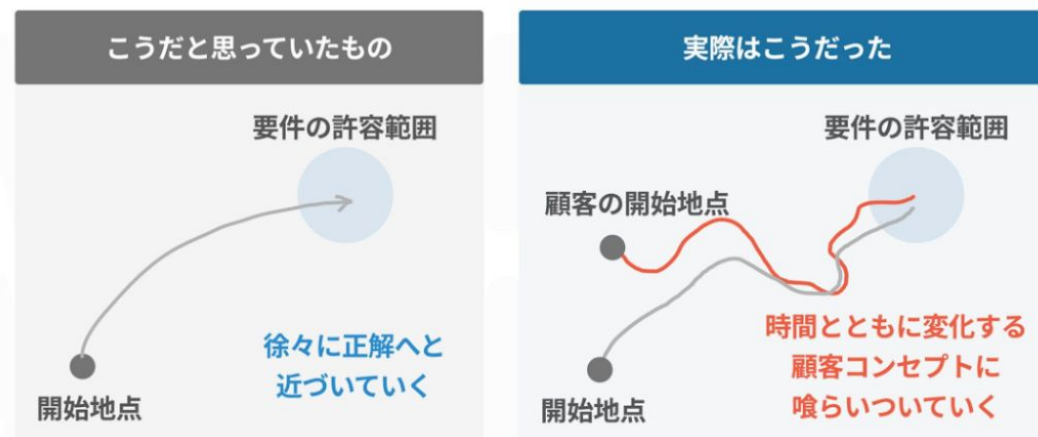
ステージング環境でエージェントの性能や動作を検証する。ガードレール等によりエージェントの安全な動作、可観測性、制御可能性を担保する。

Outer Loop

回帰テストやカナリアリリース等によりAIエージェントを本番環境にデプロイする。デプロイ後は継続的に監視を行いプロダクトのメンテナンスを行う。

35 | なぜ改善サイクルを回し続けなければならないの？

品質評価の基準は運用してはじめて浮き彫りになることも多く、継続的に評価・改善のサイクルを回すことで要件の許容範囲へと収束させていく



3. 独自のデータを定義し評価データを育てる

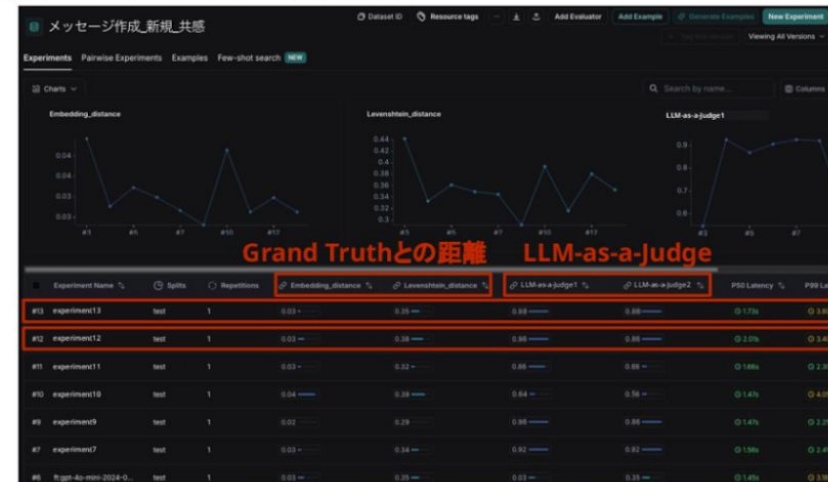
- 「自分の業務」というベンチマークはない
- 生成 AI に対するユニットテストのように扱う
- 専門家によるレビュー結果を評価データに追加する

AIエージェントの継続的改善のためオブザーバビリティ

https://speakerdeck.com/pharma_x_tech/aiezientonoji-sok-de-gai-shan-notameobuzababiritei

データセットに対して評価を実施

リリース前に評価用のデータセットを作って評価を実施



(C) PharmaX Inc. 2025 All Rights Reserve

46

リリース前の評価

4. 専門家を開発チームの一員にする

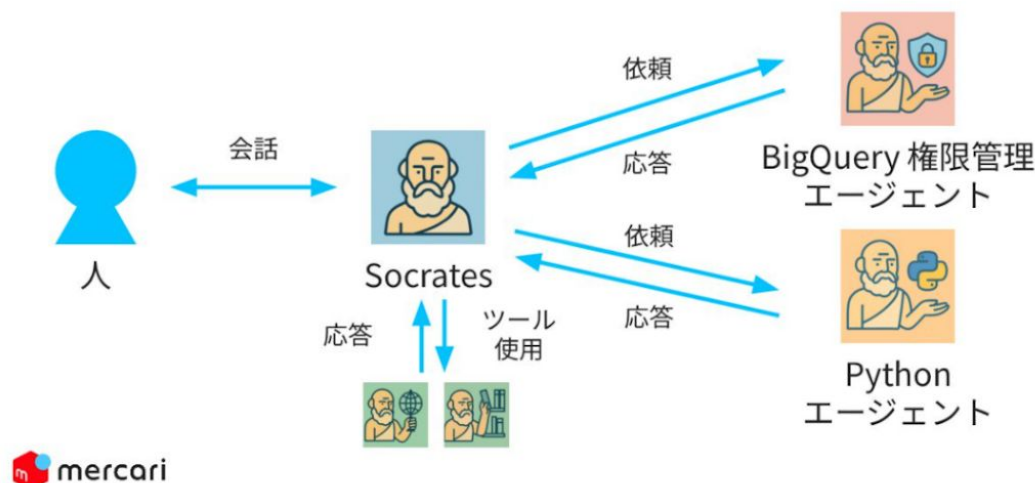
- 自然言語は開発者と専門家の共通言語
- 専門家がプロンプトを開発するのは強力なプラクティス
- 業務ごとに小さなエージェントに分割して開発

メルカリにおけるデータアナリティクス AI エージェント「Socrates」と ADK 活用事例

<https://speakerdeck.com/na0/merukariniokerudetaanariteikusu-ai-eziento-socrates-to-adk-huo-yong-shi-li>

Socrates とその他の社内ツールとの共存関係

Socrates は、ADK を基盤としたマルチエージェントシステムとして設計。各エージェントが専門タスクを担当し、協調動作。



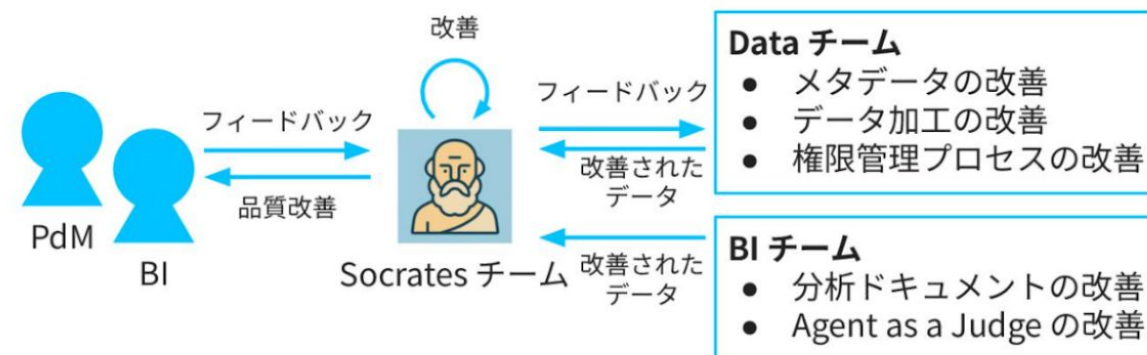
mercari

36

Skip

Socrates を育むエコシステム：チーム

- チームの連携：部門横断的な開発・運用体制とフィードバックループ
 - 利用状況モニタリング、ユーザーヒアリング、エラーログ分析、Agent as a Judge を通じて、プロンプトやツールの迅速な改善を実施



mercari

56

5. 本番環境でテストする

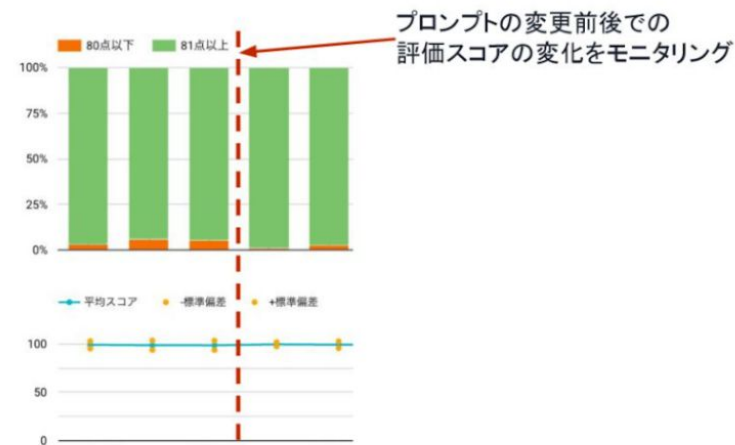
- どうしても「やってみないとわからない」
- 限定的にリリースして想定外の事象が発生しないか確認する
- モニタリングでリスクと効果を確認

AIエージェントの継続的改善のためオブザーバビリティ

https://speakerdeck.com/pharma_x_tech/aiezientonoji-sok-de-gai-shan-notameobuzababiritei

LLM-as-a-Judgeでの評価結果を日次で可視化

各LLM-as-a-Judgeのスコアを可視化することで、プロンプト変更による改善の可否を判断する

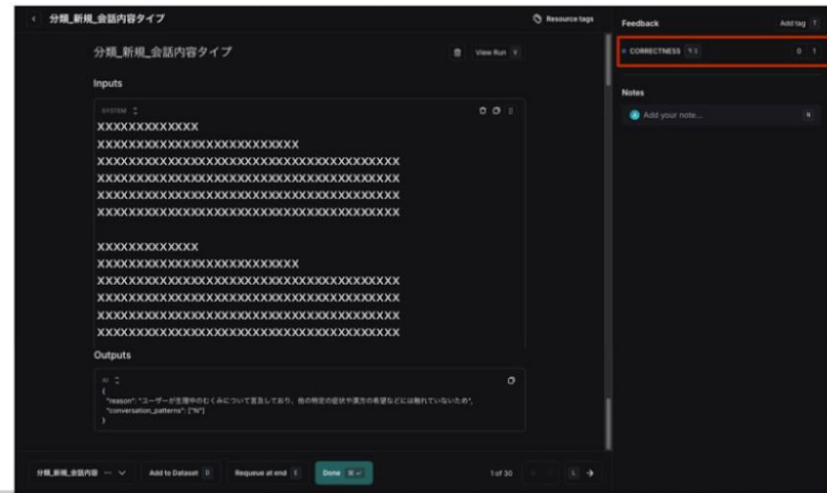


(C) PharmaX Inc. 2025 All Rights Reserve

48

アノテーションすることで本番環境での正答率も測定

LangSmithのAnnotation Queuesに蓄積して正解・不正解をチェックすることで正答率が測定できる



正解・不正解
/ 良し悪し
を手でチェック

(C) PharmaX Inc. 2025 All Rights Reserve

TOC

- DevOps, MLOps, LLMOps
- 生成AIを利用するシステム開発の原則
- **今後の展望 <-**

今後の展望

- AI セーフティ
- AI ガバナンス
- セキュリティ

AI セーフティに関する取り組み

- 現状は「ユースケースの分類を行い、リスクのあるユースケースは避ける」という取り組みが主流
- リスクがあるものの効果の高いユースケースをカバーする方法に苦心している
- 独自データセットでの評価が主流なので、データセット自体の「良さ」を計測する必要がある

AIガバナンス

- AI ガバナンスの3つの要素

1. リスクマネジメント
2. 提供価値の最大化
3. 学習する組織

- リスクマネジメントは取り組まれるものの、提供価値の最大化や学習する組織については知見が少ない

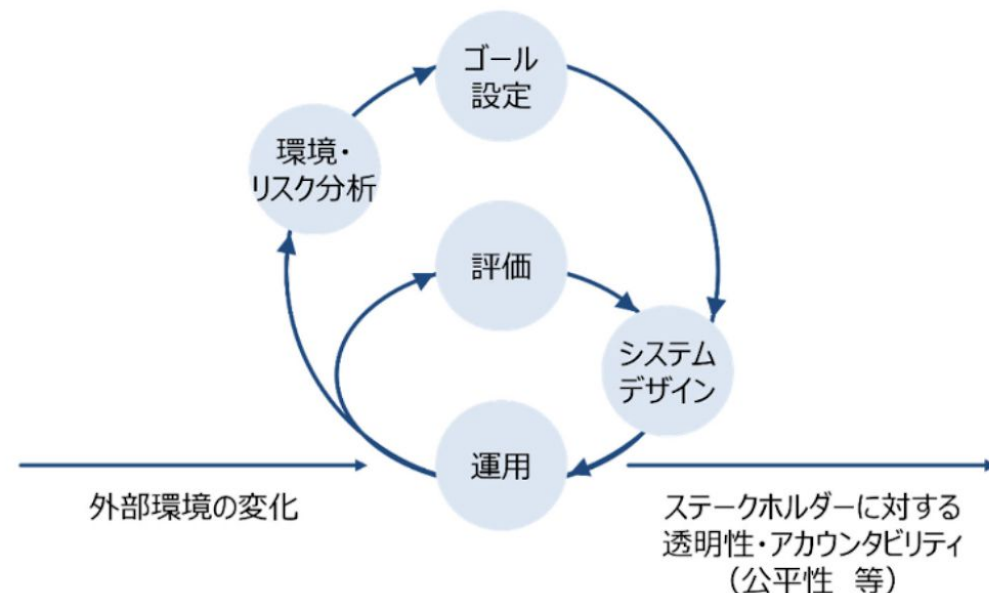


図 6. アジャイル・ガバナンスの基本的なモデル

セキュリティ

- Gmail などのサービスに組み込まれた生成 AI への新たな攻撃が開発されている
- MCP や A2A などのプロトコルはまだ仕様や周囲のエコシステムが未成熟
- 自律的なエージェントが主流となった際に何が起きるのか不透明

人間をプロンプトインジェクション攻撃する Gemini

<https://zenn.dev/carenet/articles/e7780fc0253d89>

Zenn

Log in

CareNet Engineers

目次 ▾



人間をプロンプトインジェクション攻撃する Gemini

 Yam 2025/07/02に公開

 AI

 Security

 Tech

こんにちは、セキュリティエンジニアの Yam です。AI 技術の進化は目覚ましいものがありますが、それに伴い新たなセキュリティリスクも顕在化しています。特に最近注目されているのが「プロンプトインジェクション」と呼ばれる攻撃手法です。今回はその中でも、より巧妙で気づきにくい「間接的プロンプトインジェクション」を利用し、AI に偽のエラーメッセージと、悪意のある指示を表示させるという攻撃シナリオについて、その仕組み、危険性、そして私たちが取るべき対策について解説します。

間接的プロンプトインジェクションとは？

プロンプトインジェクションは、AI（特に大規模言語モデル）への指示（プロンプト）を悪用し、開発者が意図しない動作を引き起こさせる攻撃です。その中でも「間接的」な手法は、攻撃者が AI への指示を、ユーザーが直接入力するプロンプトではなく、AI が処理する外部のデータソース（文書ファイル、ウェブサイト、メールなど）に潜ませるのが特徴です。

AIセーフティ強化に関する研究開発

- 産総研、株式会社コピー、Citadel AIで共同提案
- 「企業向け実装解説書」
としてベストプラクティス
集・事例集を作成
- さまざまな企業にヒアリングを実施中

NEDO「AIセーフティ強化に関する研究開発」の採択について - Citadel AI
<https://citadel-ai.com/ja/news/2025/04/30/nedo/>

NEDO「AIセーフティ強化に関する研究開発」の採択について

Filed under:
[News](#)

2025/04/30
[Share Article](#)



Citadel AI

NEDO
**AIセーフティ強化に関する
研究開発の採択について**

株式会社Citadel AI（本社：東京都渋谷区、代表取締役：小林裕宜）は、国立研究開発法人新エネルギー・産業技術総合開発機構（NEDO）の事業「AIセーフティ強化に関する研究開発」に、国立研究開発法人産業技術総合研究所（産総研）ならびに株式会社コピーと共同提案を行い、採択されましたことをご報告いたします。

1. 本事業の目的

まとめ

- DevOps, MLOps, LLMOps といった形で分野は変わったものの、イテレーティブに改善を行う手法は開発された来た
- 生成 AI を用いるサービス開発について、さまざまな取り組みが進んだ結果、いくつかの原則が見えてきた
- 一方で、AI セーフティ、AI ガバナンス、セキュリティについてはまだまだ取り組みは半ば
- いろいろな企業にヒアリングした結果をまとめていくのでご協力お願いします