

『AIに負けない』より『AIと遊ぶ』 ～ワクワクが最強のテスト・QA学習戦略～



大段智広
(DMM.com/ASTER)

大段智広(おおだん ともひろ)



個人情報:

- 元JaSST Kansai実行委員
- 6月に育休明けの二児の父

SNS:

- X : @dandan_611
- LinkedIn: Tomohiro Odan
- Facebook: Tomohiro Odan

略歴:

- 2012年4月 関西のメーカーソフト子会社に入社。その後、ソフトウェア開発コンサルティング会社を経て、2021年4月よりDMM.com所属。現在はQA業務のAI活用を部内で推進。

専門スキル(≡好きな領域)

- ソフトウェア開発プロセスアセスメント(CMMI)、ソフトウェアテスト
- UMLモデリング、モデルベースドテスト
- E2E自動テスト導入・開発・保守、開発インフラ構築・保守

主な社外活動

- NPO ASTER 理事 (テスト設計コンテスト担当)
- ASTER テスト設計コンテスト実行委員兼U-30クラス審査委員長
- ISO/IEEE SC7 WG26 ソフトウェアテスト 委員
- JSTQB技術委員



目次

本講演で伝えたいこと



QA/テストエンジニアの現状



いまからの学び方



AI時代の実践事例紹介



まとめ



1.本講演で伝えたいこと

「シン・ソフトウェアテスト」

～ 新時代に築き直すソフトウェアテストの入門・基礎

- 節目となる第20回 — 原点に立ち返り、次の10年に向けた方向性を描く
- AI駆動開発時代に「テストとは何か」「SQAの役割とは何か」を問い直す
- 複雑化するソフトウェア社会で品質を支える役割の重要性はさらに増す

**本講演は「AIと遊ぶ」視点で、
このテーマを個人の学び方から掘り下げます。**

QAforAI and AIforQAのどっちの話？

この講演では主に「AIforQA」の実践を深掘りしながら、
QAエンジニア/テストエンジニアの学び方を考えます。

QA for AI

AIシステム自体の品質保証

&

▼本講演の主軸

AI for QA

QA/テスト業務へのAI活用
テスト生成・実行・分析・自動化

QA/テストって、
ワクワクする技術なんだぜ!!

今日の私の講演ゴール

Goal 1

AI活用の
具体的なイメージを
1つ持って帰ってもらう

Goal 2

明日から試せる
学習アクションを
1つ決めてもらう

Goal 3

「ワクワク」で
テスト・QAに向き合う
気持ちになる

難しい話は後回し — まず「楽しそう」と思ってもらうことが最優先！

2.QA/テストエンジニアの現状

これからのQA/テストエンジニアはどうなる？

AI活用によりベテランは若手に手を借りる
必要がなくなり、**若手とベテランの格差が拡大**

これまで若手が担ってきた「作業」が
AIに置き換わることで、
若手が体験して学ぶ場がなくなるのではないかと

AIによって
バグを狙えていない
テストが量産されるのではないかと

テスト作業の多くは効率化され、
手を動かすことが減り、見極めることが増える

組み込み系など
ツールやAIが代替しにくい領域はなくなる

開発チームとの壁がなくなるのではないかと

AIが出さないであろうテスト観点でテストする人たち
「探索テストのエキスパート」が重宝される

エキスパートやマネジメント系をやってきた人にと
っては、「誰が」作成したものなのかが変わる
だけで**やることは実は変わらない**

QA/テストエンジニアの仕事はAIに代替される

● 良い面、● 悪い面
を考えてみましょう

● 良い面 — AI時代のQA/テストエンジニアのチャンス

- **若手の学びの質が向上する**
 - AIが定型作業を担い、「なぜこのテストが必要か」を考える訓練に集中できる
- **QA/テストエンジニアはスペシャリストとして価値向上する**
 - コモディティから高度判断・設計・戦略を担う専門職へ
- **QAが経営・開発上流の戦略的パートナーになる**
 - 品質リスクを可視化する役割として発言力が増す
- **機能安全・規制対応領域で需要増大する**
 - AIが担えない「なぜ安全か」の説明責任で人間のQAが不可欠
- **探索的テスト思考・熟練判断力は高付加価値ポジションになる**
 - 人間の「ひらめき・共感」はAIに代替されにくい
- **テスト品質を評価できるレビュアー需要が急増する**
 - 「AIが作ったプロダクト/テストを批評できる人間」が希少かつ重要なポジションになる

● 悪い面 — QA/テストエンジニアが向き合うべき課題

● 体験なき理解は脆く、大きな失敗が発生しやすくなる

- バグの嗅覚・テスト漏れの感覚は経験で身につく。
- 若手育成の仕組み再設計が急務。

● 新規メンバの参入障壁が高くなる

- 業界への人材供給が細り、中長期的に人材が枯渇するリスクが高くなる

● スキルの再教育なしでは中間層が空洞化する

- テスト実行が主業務だったQA/テストエンジニアが急に戦略・判断を求められても、対応できない

● ベテランのAIへの過信が誤出力を見逃す

- 「自分のやり方で問題ない」という過信が批判的評価を妨げる

● AIが量産する粗製テストがレビュー負荷を集中させる

- 少数の熟練QA/テストエンジニアに負荷が集中し、疲弊・品質低下が同時に起こるリスクが高くなる

一概には言えないが、
海外ではどうなっている？

【2025年12月】Momentic — "Will AI Replace QA Engineers?"

AIはQAエンジニアを「置き換える」が、その根本原因はAIではなく、過去10年の業界トレンドとして、QAエンジニアは少しずつ衰退しつつある。

従来型QAの問題点（2015年からの傾向）

- **コストが高い**
 - ・ 10人チームで年間\$1M超(予算25%以上)
- **リリース速度が低下する**
 - ・ ボトルネックになりやすい
- **組織が分断される**
 - ・ 開発とQAの断絶が品質低下を招く
- **バグ発見の遅延する**
 - ・ 修正コスト増加 & リリース遅延する
- **Microsoft・Yahooは2015年に独立QA廃止**

AIテストツールがもたらす解決策

- **コードレスでのテスト作成（自然言語入力）**
- **自己修復テスト（変更に自動追従）**
- **CI/CDパイプラインとの直接統合**
- **AIエージェントが将来的にPRまで自動修正**
- **64%の組織がQAにAIを導入済みまたは計画中**

【2026年1月】Tricentis Transform 2025 — "QA Trends for 2026"

以下の4つの示唆が示されている。

AI生成コードのリスクが増している

- 昨年のコードの40%以上がAI生成。
- 88%の開発者がAI生成コードのデプロイに自信なし (Stack Overflow調査)
- 29%がAIエラーによるロールバックを経験 (GitLab調査)

QAの役割は根本的に変化している

- QAチームの役割は「テストを書いて実行する」から、「品質目標を定義しAIの出力を監督する」へ。
- AIが失敗するとエンタープライズ規模で失敗する時代に。

Human-in-the-Loopは必須である

- AIパイロットの95%が不適切なガバナンスが原因で失敗。
- そのうち60%はコンプライアンス違反。AIは確率的であり、決定論的ではないため人間の監視が前提となる。

AIをいち早く導入した企業が先行している

- 「AIを知る人と知らない人のギャップは今は最も小さい。しかしこのギャップはこれから急速に広がる」
 - Sophia Velastegui (元Microsoft Chief AI Officer)

AIプロダクト リリースの現実

54.5% がすでにAI機能を
リリース済み

44.1% が運用コスト超過を
理由にAI機能を停止

29.6% でAIによる
テスト予算が
開発予算より速く増加

高品質チームの共通点 & 警告

完全自律型AIを開発しているのは全体の25%未満。
大半は依然として人間監視が前提になっている

継続的評価ループ・ハイブリッド（人間×AI）テスト・ドメイン
専門家の関与が成功の鍵

「AIにテストを書かせると要件の検証が飛ばされる」— Adonis
Celestine (Applause)

レッドチームテスト(悪意ある攻撃のテスト)は必須。内部開発者
のみでは自分のコードの脆弱性を見逃しやすい

QA/テストエンジニアに対する昨今のトレンド

①完全代替は すぐに起きない

定型的QA作業の
40～60%を2030年
までに自動化される

75%の組織が
AIによるテストを戦略の
柱と位置づけ

②新役割への 移行が加速する

AI活用型QAエンジニア
への転換

品質戦略エンジニア/
データ品質保証エンジニア
の誕生

③スキルアップが 急務になっている

72.8%の
QA/テストエンジニアが
AIによるテストを
2026年の最優先に選択

70%の組織が
QA人材の
AIスキル習得を期待

昨今のトレンド① 完全代替はすぐには起きない

- AIは定型的・反復的なテスト実行を自動化するが、下記の検証は人間にしかできない。
 - 探索的テスト
 - リスク評価
 - ユーザー体験の検証
- 75%の組織がAIによるテストを戦略的優先事項とする一方で、実際に導入済みの組織は16%にとどまっており、計画と実装の大きなギャップが存在する。

40～60%

定型的QA作業が
2030年までに自動化

75%

の組織がAIによるテストを
戦略の柱と位置づけ

昨今のトレンド② 新役割への移行が加速する

「テストを書く専門職」から「AIエージェントを指揮する品質責任者」へ変容する。

- テスト戦略の設計/AIの出力監督 / リスクベース優先付け / 開発チームへのコーチングが主要業務となる。
- より高付加価値な職種への昇格と見なされている。

従来のQA/テストエンジニア

- テスト作成
- テスト実行
- バグ報告

現在（過渡期）

- QA業務自動化
- 一部戦略立案
- AIツール活用

AI時代の新役割

AI活用型QAエンジニア
品質戦略エンジニア
データ品質保証エンジニア

昨今のトレンド③ スキルアップが急務になっている

今すぐ習得すべきスキルとして下記が示されている。

- AI駆動テスト生成の理解と活用
- 予測型品質分析
- リスクベース・自律テスト手法
- ノーコード / ローコード自動化ツール
- プロンプトエンジニアリング
- AIの出力を監督するガバナンス設計

72.8%

QA/テストエンジニアが
AIによるテストを
2026年の最優先に選択

70%

の組織が
QA人材の
AIスキル習得を期待

AI活用型QAエンジニアへ
スキルアップしましょう！
でも・・・どうやって？

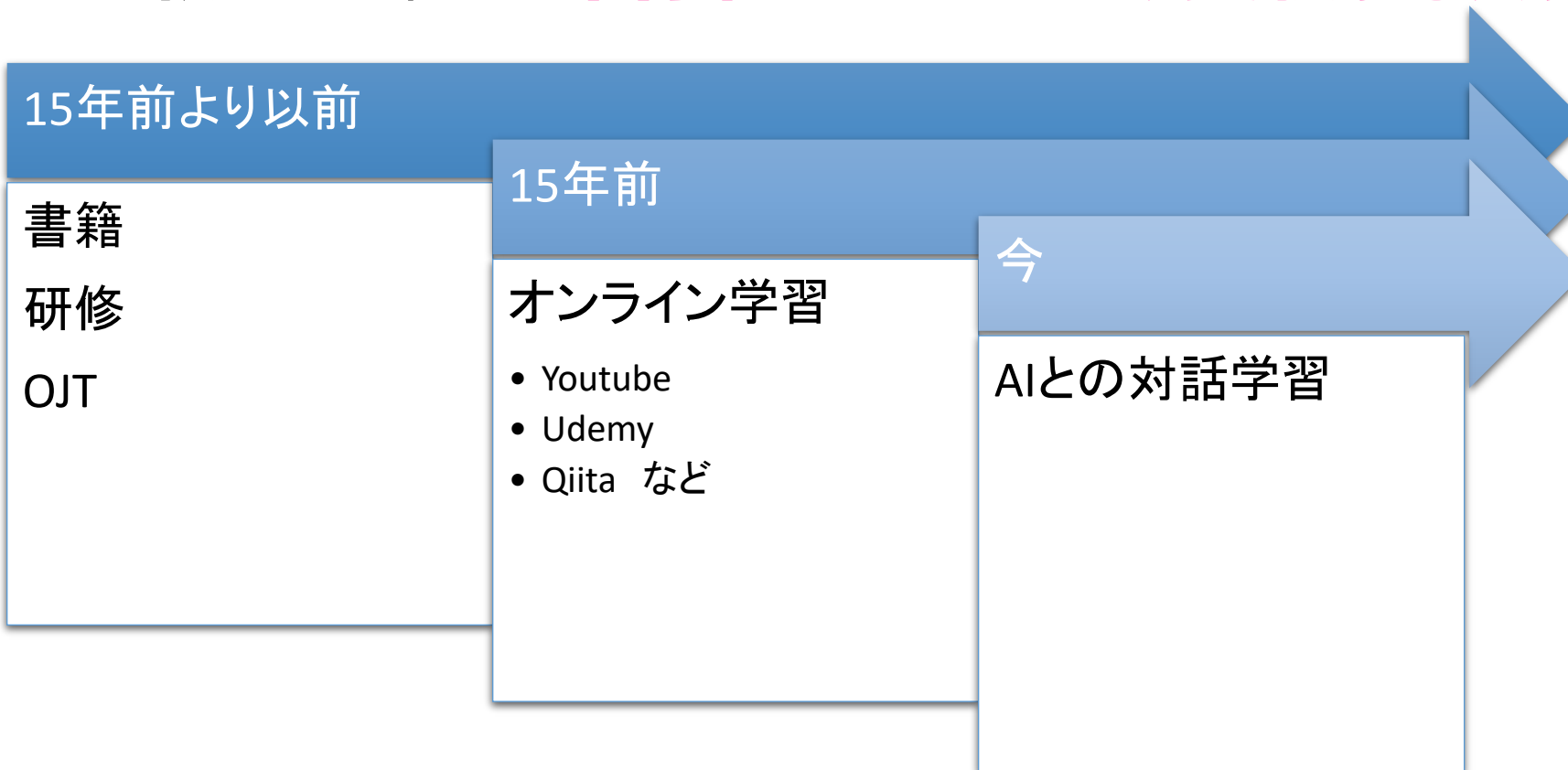


3. いまからの学び方

AI時代が変える「学び方」

変わったのはスピードだけでなく、学びを考えるのが自分以外の誰かや組織ではなく、個人に移っている。

- インプット偏重 → アウトプット重視
- 誰かに教わる時代 → **自ら問いを立てて、実験するスタイル**へ変化



自ら問いを立てて、
実験するとき...

ワクワクしませんか？

あなたが強く感じるQA/テストの「ワクワク」は何ですか？

1. 【発見】のワクワク

例：アプリの誰も触らないような深い階層で、誰も気づかなかった画面の崩れ（バグ）を偶然見つけたとき。

2. 【攻略・達成】のワクワク

例：「再現確率10%」と言われるような、めったに起きない幻のバグを発生させる確実な手順（再現手順）を突き止めたとき。

3. 【創造】のワクワク

例：これまで手動で3時間かかっていたテストを、ボタン一発で5分で終わらせる「自動テストの仕組み」を設計しているとき。

4. 【予測・戦略】のワクワク

例：開発者の修正コードを見て、「この直し方だと、おそらく別のあっちの機能に影響（デグレード）が出ているぞ」と予測してテストに向かうとき。

5. 【共創・繋がり】のワクワク

例：リリース直前のバグをお祭りのようにチーム全員で探し、みんなで協力して品質をハイクオリティに仕上げたとき。

6. 【変身・なりきり】のワクワク

例：「超せっかちでボタンを連打するユーザー」や「システムを壊そうとする意地悪なハッカー」になりきってアプリを操作するとき。

「あそび＝まなび」
という考え方もある

あそびが育む「生きる力」!!

「あそび」とは、「やってみたい」という動機から**自発的に始まるリアルな体験**であり、その結果として人を育てていく過程のこと。(≠娯楽)

想像力

- 自由な発想で、限りなく広がるあそびの世界。
- 夢中になって遊んだ経験は、夢見る力につながっていきます。

発見と理解

- 見て聞いて、触って確かめて、発見が感動とともに学びや理解につながっていきます。

創造性

- 誰もが持っている、その人だけの感性を生かすこと。
- 湧きあがるイメージを自らかたちづくる楽しさや、つくりあげたときの達成感をもたらします。

コミュニケーション能力

- 楽しいこと、おかしいこと、時には秘密…。それらを誰かと共有する楽しさ。こうしたことの積み重ねが、人と人(AI)の間に信頼感を育みます。

AIと遊ぶことで私たち自身がさらに成長できると捉える

- 人の考える力がなくなるといわれるが、それは人を育んでいく過程(あそび)にAIを組み込もうと考えていないからとも思える。
- AIがやってくれること（今までの自分より早く情報をまとめたり、単純な知的作業）を人の成長のきっかけにする。
 - 踏み込んで「**自由な発想**」、「**イメージの具現化**」、「**発見と理解**」、「**共有の楽しみ**」を意識する。



私たちはAIやロボットと
どう付き合うかは知っているはず...



AIと遊ぶことで私たち自身がさらに成長できると捉える

- 人の考える力がなくなるといわれるが、それは人を育んでいく過程(あそび)にAIを組み込もうと考えていないからとも思える。
- AIがやってくれること（今までの自分より早く情報をまとめたり、単純な知的作業）を人の成長のきっかけにする。
 - 踏み込んで「**自由な発想**」、「**イメージの具現化**」、「**発見と理解**」、「**共有の楽しみ**」を意識する。



こんなこといいなできたらいいなを考える

「ワクワク×AIとの遊び」が最強の学習戦略である理由

最初は危機感かもしれないが…

学習継続を決める最大要因は「**楽しいかどうか**」

- 義務感・焦り・恐怖からの学習は短期間しか続かない
- 「AIに負けないために勉強する」は消耗する学び方
- 「AIと遊んでいまよりちょっと面白いことをやりたい」は自然に続く学び方
- 内発的動機づけは外発的動機づけより長期パフォーマンスが高い

✕ 消耗する学び

「AIに負けないために勉強する」
→ 義務感・焦り・恐怖 = 短期間しか続かない

VS

✓ 続く学び

「AIと遊んで面白いことをやりたい」
→ 内発的動機づけ = 長期パフォーマンスが高い

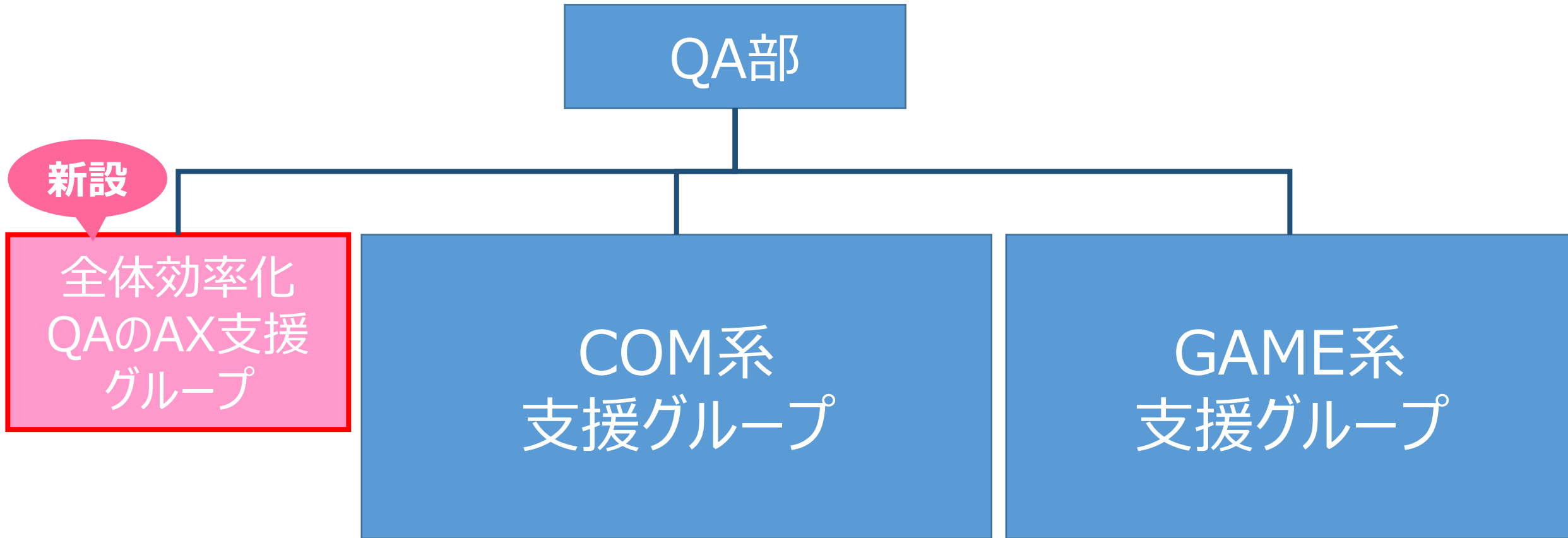
まず「試してみたい」という感情を大切に！

学習アプローチ&戦略 — 試して・つながって・続ける

- **インプット偏重から「試して学ぶ」アウトプット型学習へ**
 - **AIと対話しながら学ぶ**
 - 思考の壁打ち相手・学習パートナーにする
 - **スモールスタート**
 - 明日から1つ試せることを見つける
- **実践の場を積極活用する**
 - 普段の業務でAI活用を小さく試し続ける「実験文化」を自分から作る
- **実験したことをAIだけでなく、他の人にも共有する**
 - 共有することで新たなアイデアや発想が生まれてくる
 - そのアイデアや発想を試す良いサイクルを回し続ける

2025年 春

QA部の構成



よーし、みんなで
AIと対話学習して
業務改善しよう！

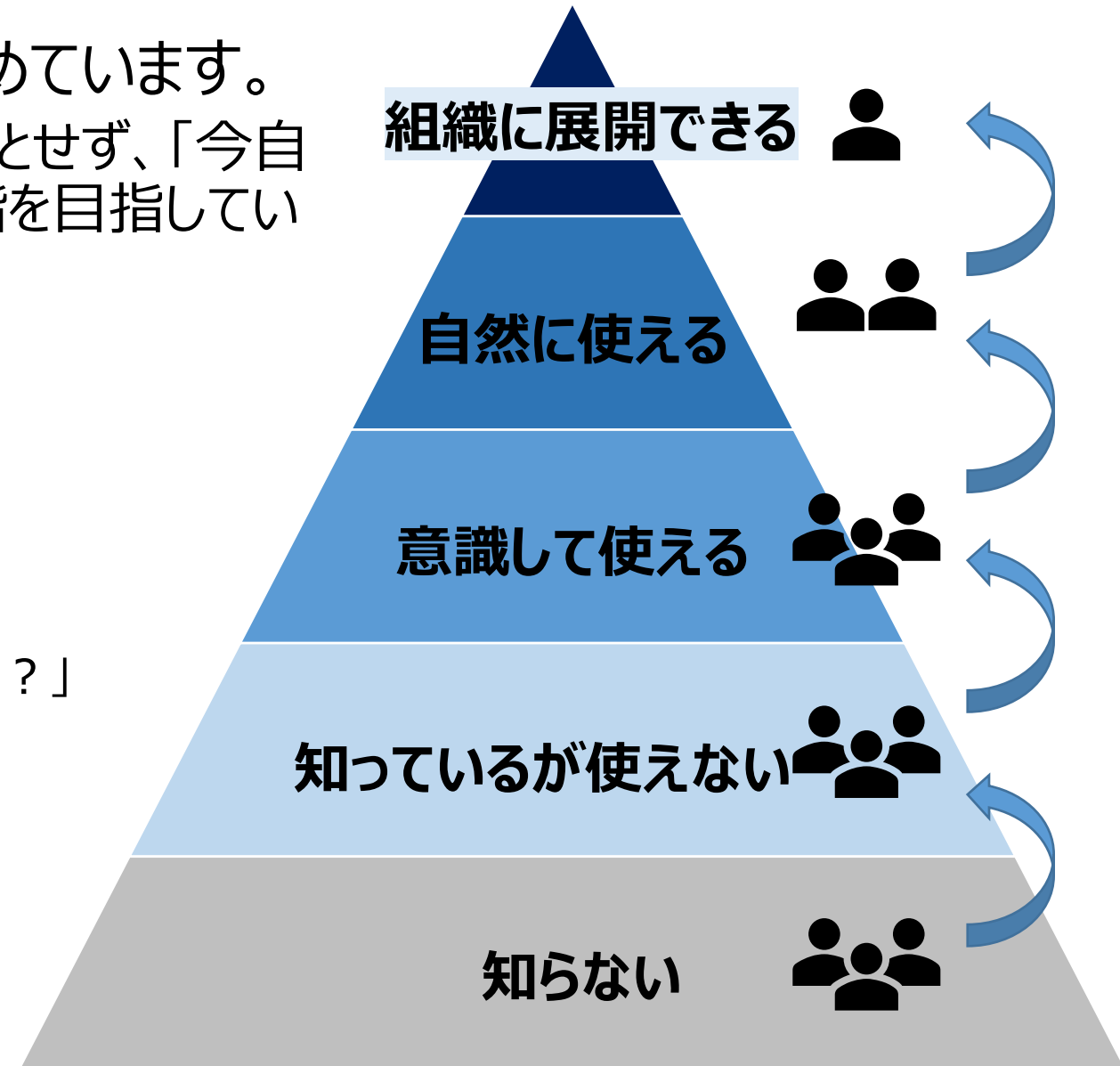
AIツール触ったことないです！

By 現場メンバ

メンバーの状態にあった
学習方法
を考えましょう。

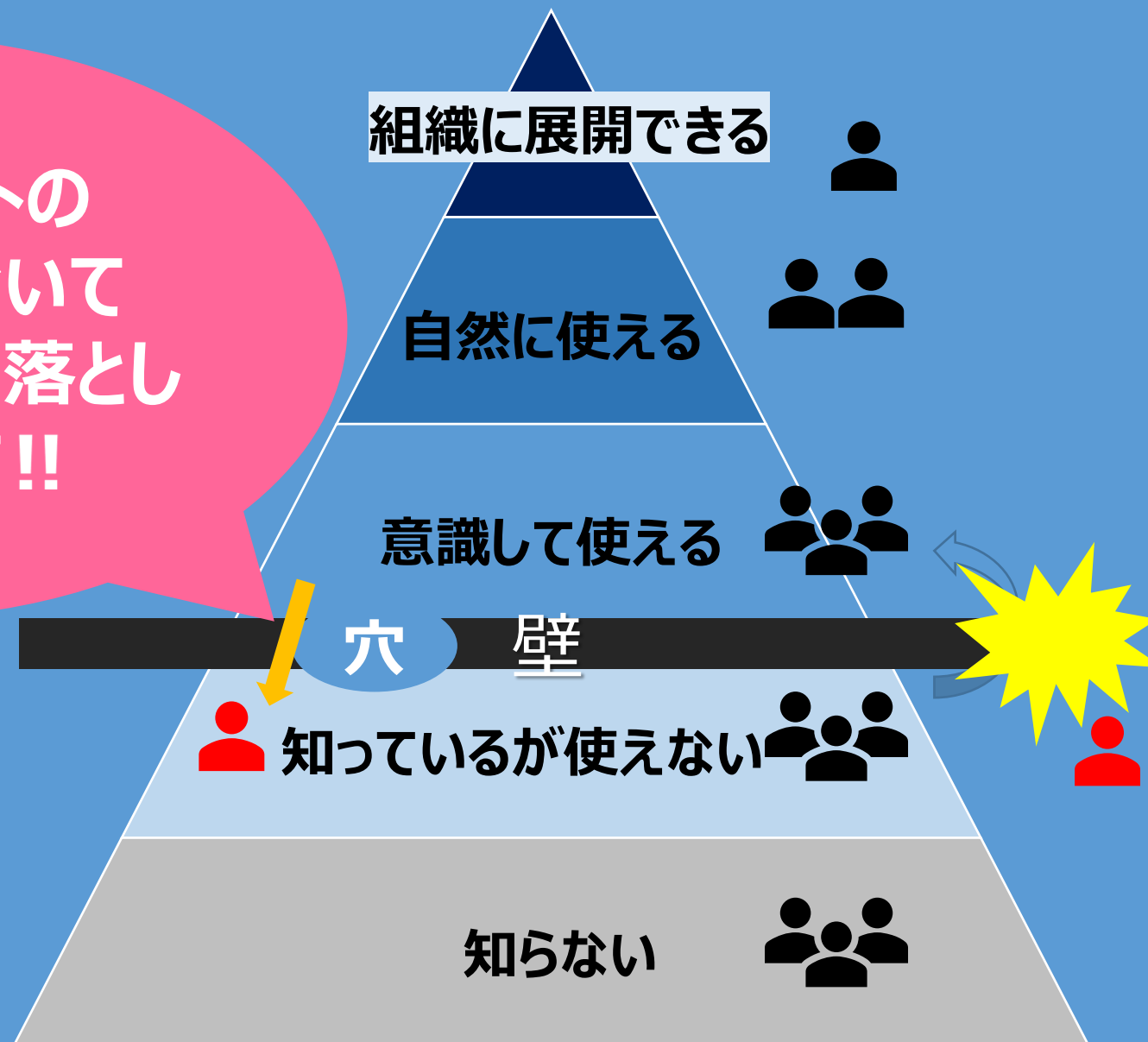
AIとの対話学習の5段階

- 25年から学習の5段階を意識して進めています。
 - メンバに対して、段階を一足飛びにしようとせず、「今自分がどの段階か」を把握して、次の1段階を目指しています。
- メンバに対するAI活用推進活動例
 - 数か月に一度のアンケート実施
 - 定期的な活用ヒアリング・デモの実施
 - 「いつもやっている一番面倒なことはありますか？」
 - 「こうAIに依頼するとこんな風にできますよ！」
 - 困ったときのフォロー
 - バイブコーディングの推奨
 - テスト業務を効率化するツール開発



一年間続けていたら
AI対話学習の壁や落とし穴
が見えてきた

QAやテストの
AI活用において
ここには、壁や落とし
があります!!



AIにどう依頼したらいいか…
AIの成果物を信用できない！

By 現場メンバ

壁や落とし穴を回避して
AIと遊びながら学ぶためには
4つのステップがある

AI時代のQA/テストエンジニア個人の学習ロードマップ°

壁や落とし穴を回避してAIと遊びながら学ぶための4つのステップ

1

基礎を固める

AIの不完全なアウトプット
を見抜く基礎
を身に着ける



2

AIを使いこなす

思い通りにAIを動かす
指示 & レビューをする



3

開発全体を見渡す

ミスを防ぐための
仕様確認と安全な修正



4

組織と社会を守る

安全性と信頼性を担保する
ルールと仕組み





ステップ1：【基礎を固める】AIに振り回されない「不変の基礎」

なぜ最初に必要か？

- AIが出力したテストケースやコードの妥当性を評価する基準は人間にしか持てない。
- 出力が「網羅されているか」「偏りがないか」を、直感ではなくロジックで判断するため。

習得すべき知識やスキル

検証（Verification）と妥当性確認（Validation）の概念を知る

「正しく製品を作っているか」と「正しい製品を作っているか」の違いを明確に説明できる知識

テスト設計技法をマスターする

同値分割法・境界値分析・デシジョンテーブル・状態遷移テストを要求仕様から自力で設計できる思考を身に着ける

共通語彙、品質モデルを理解する(できれば、この段階でソフトウェアエンジニアリング手法も)

JSTQB Foundation Levelの用語理解・ISO/IEC 25010の品質モデル（機能適合性・信頼性・使用性など）の把握
ソフトウェア開発のための手法(要求分析、設計、コーディング)として何があるかを把握



ステップ2：【AIを使いこなす】思い通りにAIを動かす「指示&レビュー力」

壁や落とし穴：

- 日本人は具体的に書くことに慣れていない人が多い
 - 主語・前提を省略 → AIからの外的な回答しか引き出せない
- 綺麗な文章を書こうとして情報が不足し、AIのハルシネーションを誘発する

壁や落とし穴の回避方法：

「3つの要素」テンプレートで超ローコンテキスト指示を徹底する



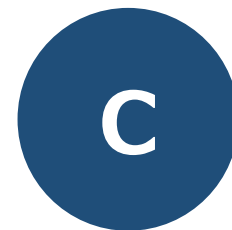
Context（前提）

私は〇〇の機能のテストをしています



Constraint（条件）

異常系だけ、5パターン、表形式で出してください



Criteria（基準）

期待される結果（何が正解か）も書いてください



ステップ3：【開発全体を見渡す】ミスを防ぐための仕様確認と安全な修正

壁や落とし穴：

- ・ 既存システム全体の流れを理解し、仕様変更プロセスを組み立てるのはハードルが高すぎる
- ・ AIに修正コードを丸投げ → AIの「思い込み」で予期しない場所が壊れる（デグレード多発）

壁や落とし穴の回避方法：

コード修正前の3ステップの仕組みを実践する

- ① 要求仕様の具体化：曖昧な要望をAIが誤解しないレベルで要求仕様としてドキュメント化する
- ② 影響範囲の可視化：変更仕様と既存コードの関係をコード修正前にリスト化する
- ③ 修正計画の明確化：変更前後のロジックの違いを自然言語で記述する

チケット管理ツールの運用 & ユーザー視点のテストを設計する

- ・ Jira/Redmineでのバグチケット正確な起票 & AIによる根本原因分析（RCA）
- ・ ペルソナや利用シナリオ（ユースケース）を想定したシナリオテスト設計をする
- ・ Google Analyticsなどの行動ログを分析する

ユーザー視点シミュレーションで仕様の盲点を探る

AIを「多様なユーザー」に変身させ、仕様の使いにくさを一緒に探るパートナーにする。

例：「あなたはこのシステムを明日から使わなければいけない『IT操作が少し苦手で、手順が複雑だと迷ってしまう心配性なユーザー』になりきってください。
『ここが少し分かりにくいかも』と感じるポイントを5つ、ユーザーのセリフ形式でフィードバックしてください」



ステップ4：【組織と社会を守る】安全性と信頼性を担保するルールと仕組み

ステップ3までを経て、ルール、仕組みを構築して、プロダクト品質だけでなく、AI開発がはらむ組織・社会的リスクをコントロールする方法を検討する。

静的解析・品質監査ツールの活用

例：SonarQube / lint系ツール

AIが出力したコードのバグや技術的負債を自動検知する

セキュリティ・ライセンス脆弱性診断

例：Snyk / FOSSA

AI生成コードのOSSライセンス違反や脆弱性を自動スキャン・監査する

CI/CDパイプラインへの組み込み&構成管理

例：GitHub Actions / GitLab CI など

静的解析 → 脆弱性スキャン → 自動テストが自動で走る品質ゲートの運用する

AI品質基準（QA4AI）の理解

例：OWASP Top 10 for LLM

AI特有の脆弱性を定義したセキュリティガイドラインの基本把握する

重要：個人学習と組織実務は切り分けておく

- 個人学習ではステップ4が最後だが、実務でセキュリティ対策を最後に回すのは絶対NG
- 最初から自動的に脆弱性スキャンが走るなど「自動ガードレール」などの仕組みを全員に適用しておく

⚠ 順序を守ることの重要性

特に、基礎(ステップ1、2)を飛ばして、いきなりAI活用や高度な仕組みづくりに取り組むと…

- ✕ ツール(AI)に使われるQA/テストエンジニアになってしまう
- ✕ ツールやAIの要求仕様が変わった瞬間に何もできなくなる
- ✕ AIが作った成果物をただ眺めるだけになる
- ✕ 自身のキャリアの再現性が失われ、長期的な成長が止まる

まず、自分の頭を鍛え、次にAIを武器にし、最後に組織を変えることを目指しましょう！

これらがそろえば、自ら問いを立てて、実験するワクワク状態に！！

この4つに気をつけて健全にAIと遊んでQAとテストについて学びましょう！

1

基礎を固める

AIの不完全なアウトプット
を見抜く基礎理論
を身に着ける



2

AIを使いこなす

思い通りにAIを動かす
指示&レビューをする



3

開発全体を見渡す

ミスを防ぐための
仕様確認と安全な修正



4

組織と社会を守る

安全性と信頼性を担保する
ルールと仕組み





4.AI時代の実践事例紹介

【創造のワクワク】

QA/テスト

AIエージェント開発

に取り組んでいます。

【祝リリース】 qa-ai-workspace v1.1.1.0 爆誕!

Cursor × QA業務が劇的にラクになる!

テスト実装からモバイル (Maestro) 実行まで、まるっと自動化の魔法の空間



まだ手作業でAIに
プロンプト投げて
消耗してるの？

毎週
リリース

【祝リリース】qa-ai-workspace v2.0.0 爆誕!

QA業務の「面倒くさい」を
根こそぎ解決する救世主、ついに登場!



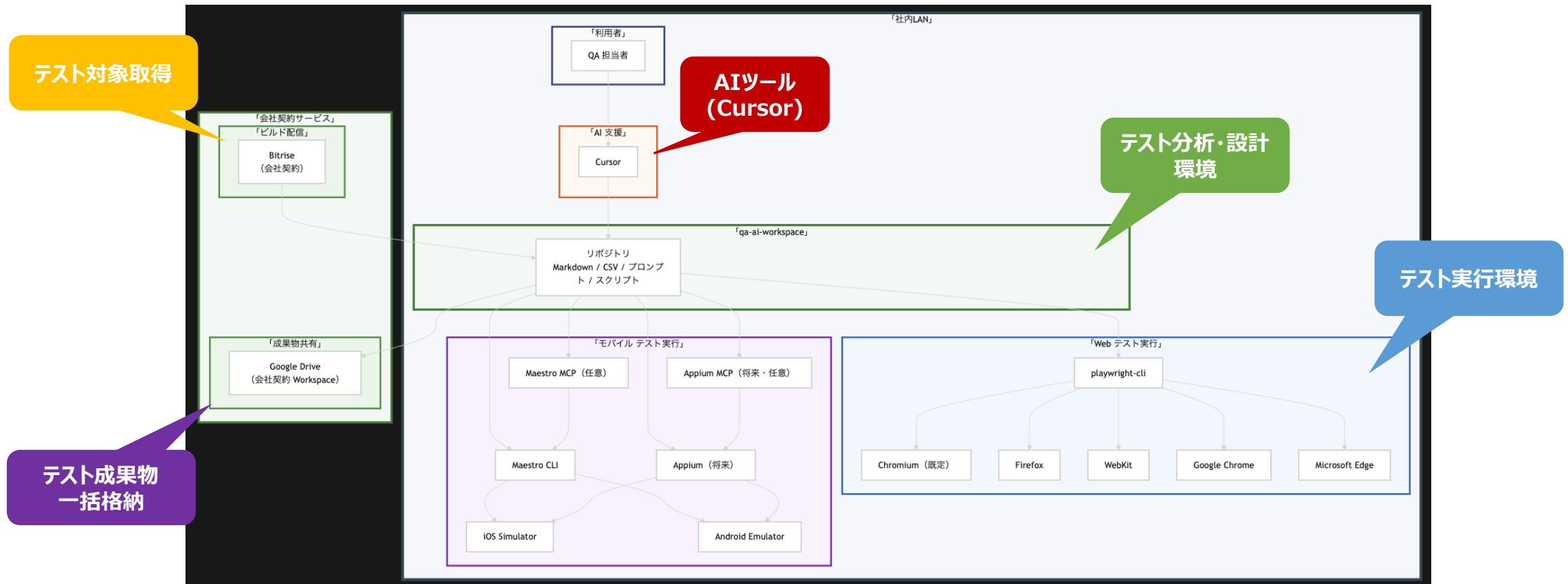
えっ、これ以上
ラクになるん
ですか!?

自動化の限界、
超えてみない?

【創造】のワクワク：AIEージェントによるE2Eテスト開発

AIと創造したこと

- Webブラウザ、モバイルアプリのテスト実行環境を構築して、テストケースがあれば、いつでも、どこでも、誰でもテストをできるようになりました。
 - 環境構築には利用者に対して丁寧なフォローが必要だったが、AIが実施

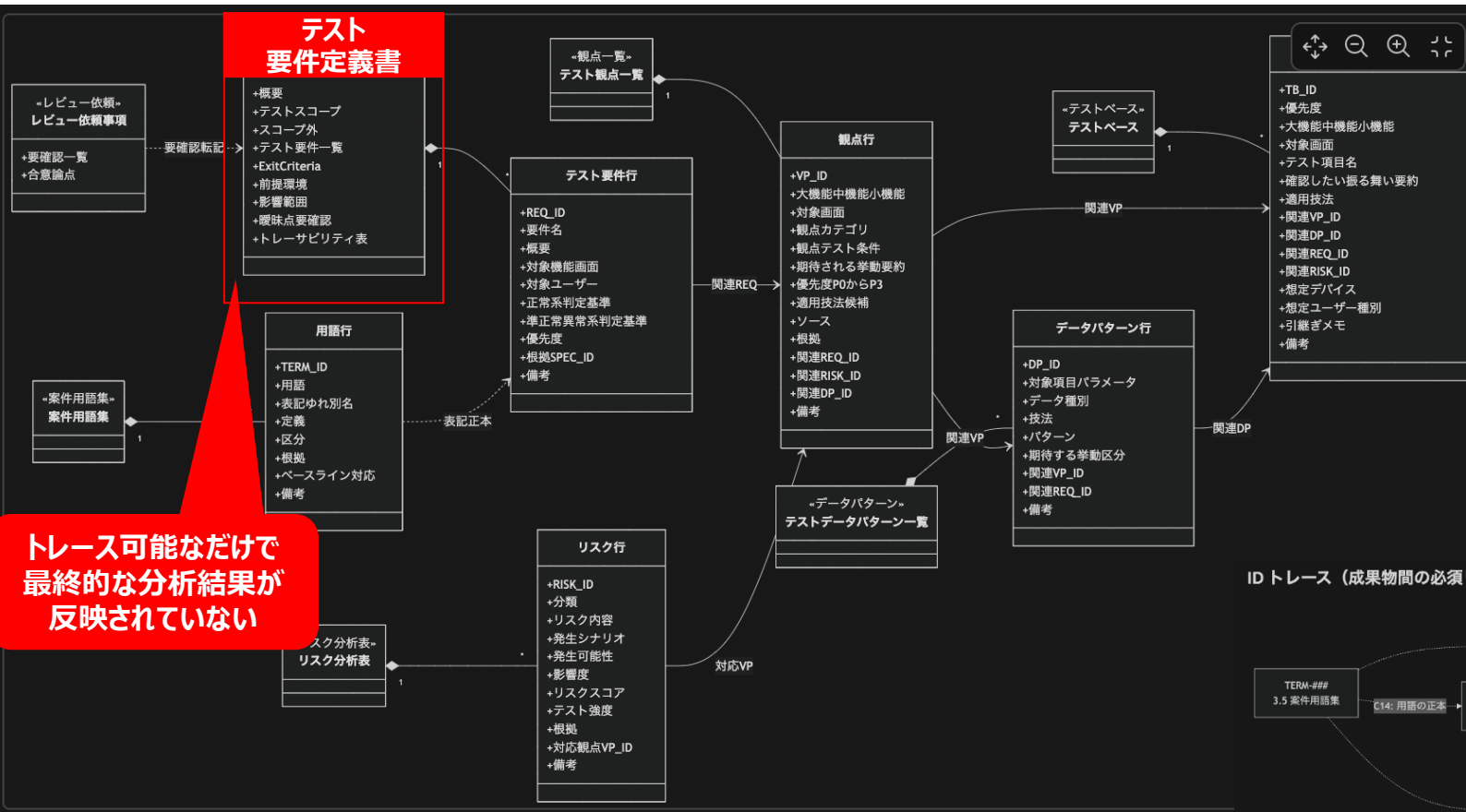


【発見】のワクワク：AIエージェントによるE2Eテスト開発

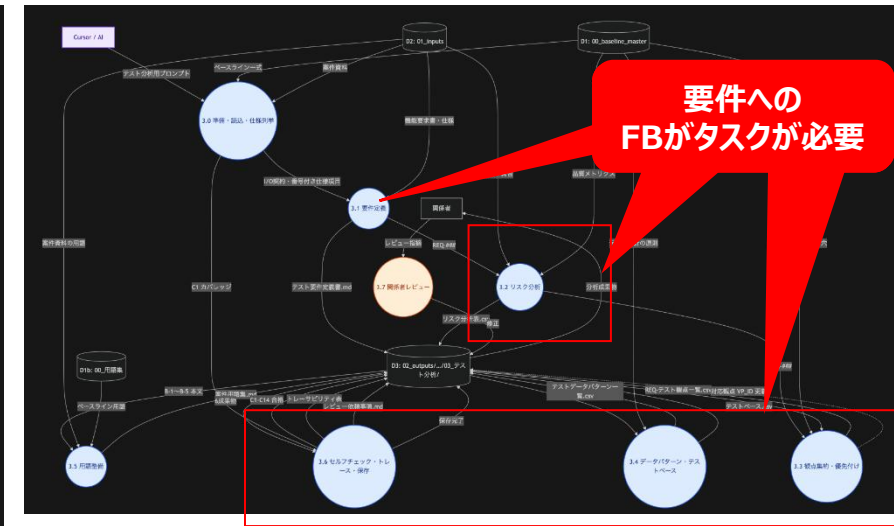
AIと発見したこと

- テストプロセスや成果物の冗長な個所を見つけてチームで共有/議論できた。
 - 今までは、プロセスや成果物の収集・理解に、ものすごく時間かかり、断念していたがAIで可能に。

成果物定義(クラス図)

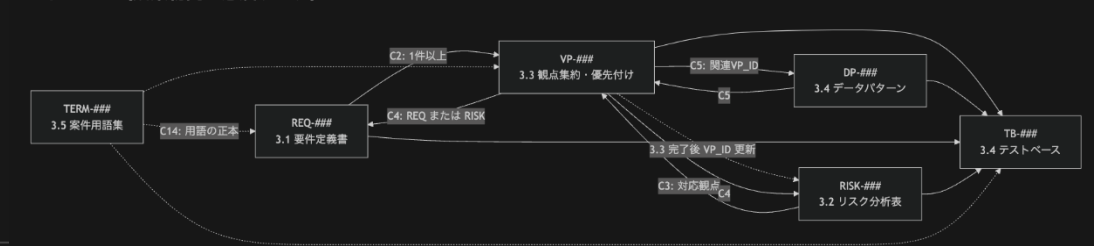


プロセス定義(DFD図)



IDトレース

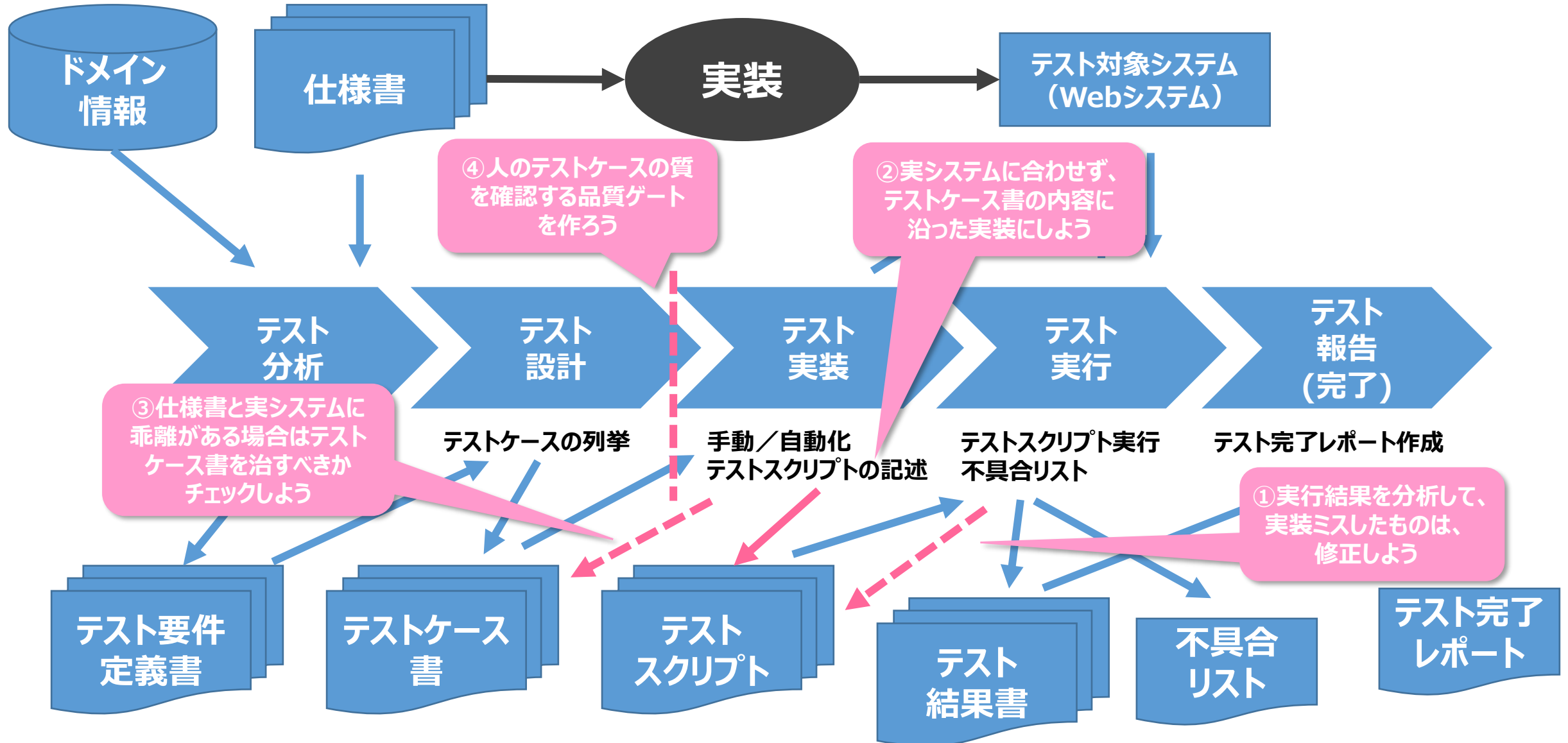
IDトレース (成果物間の必須リンク)



トレース可能なだけで
最終的な分析結果が
反映されていない

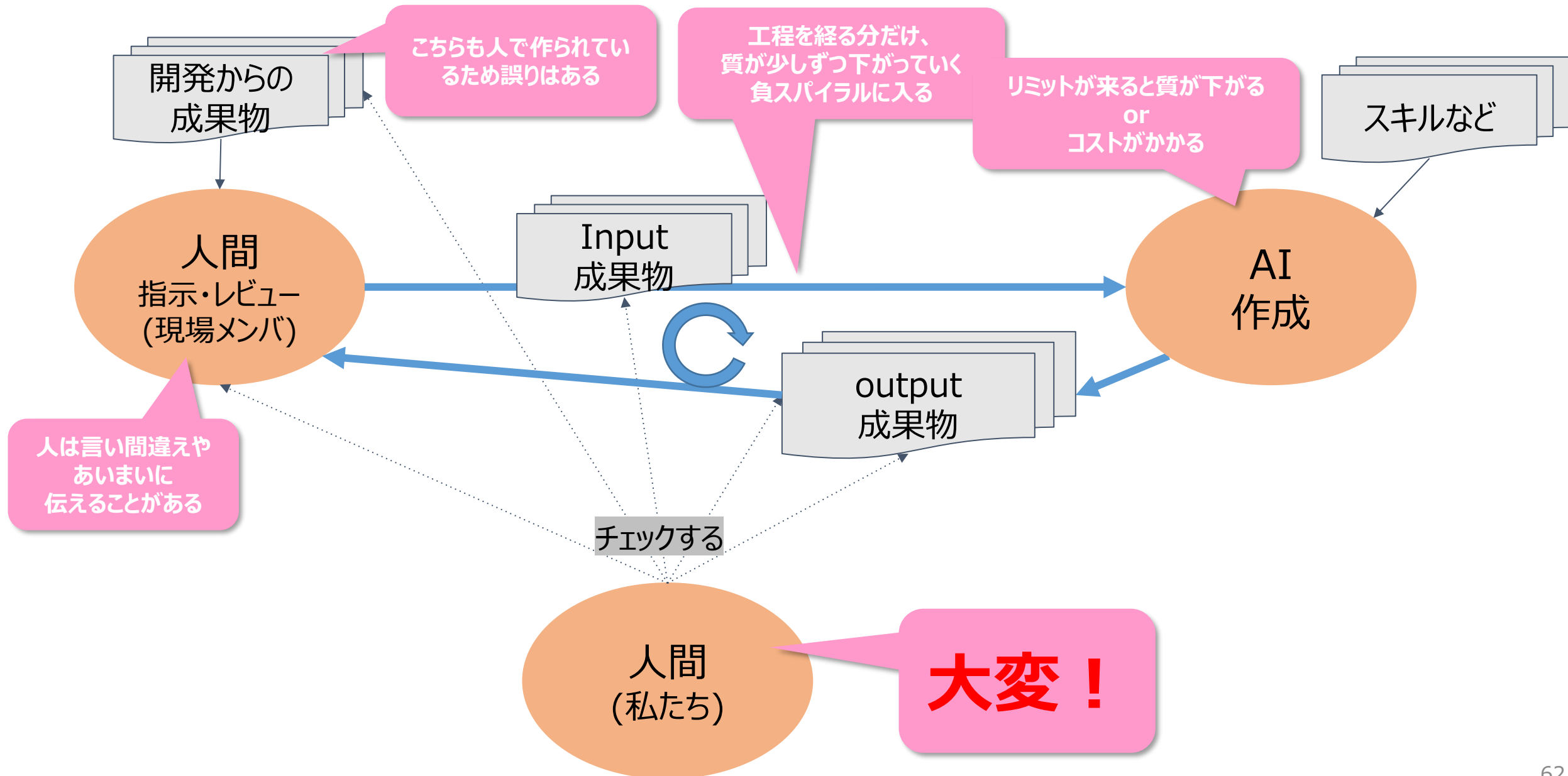
要件への
FBがタスクが必要

AIと学んだこと(発見と理解)①



AIと学んだこと(発見と理解)②

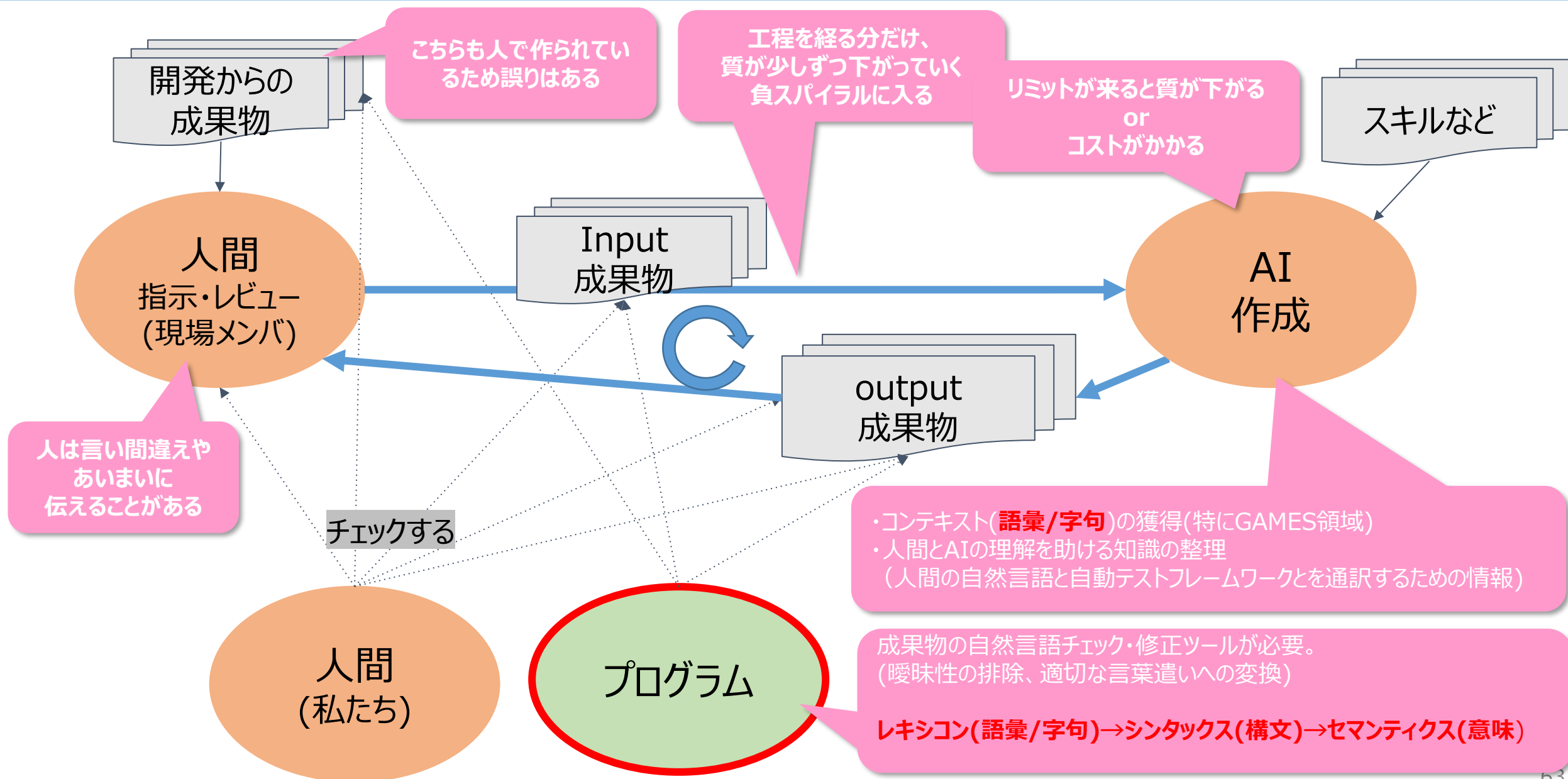
コストに応じて
質が変動



AIと学んだこと(発見と理解)②

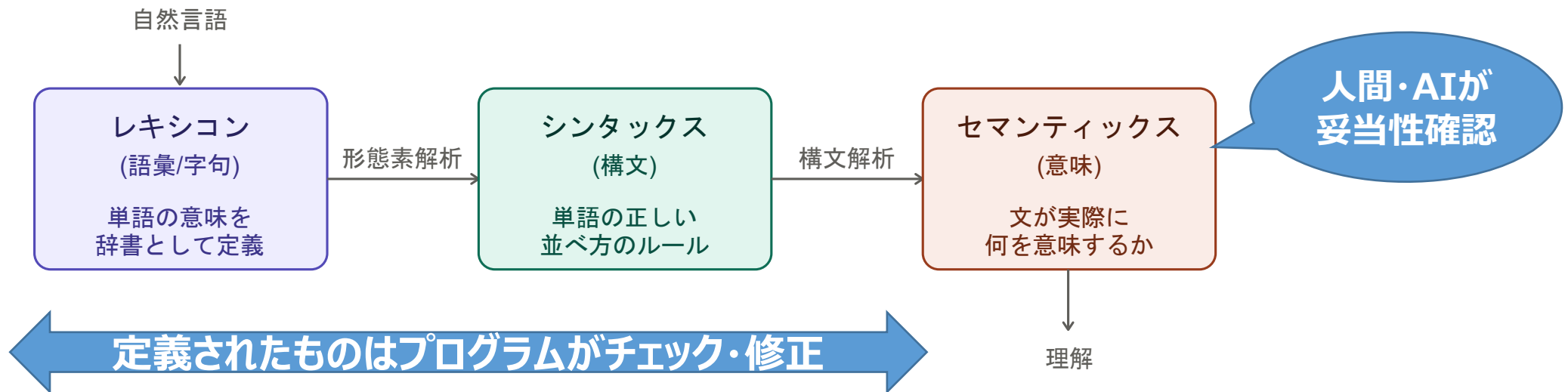
壊れやすいが
質が一定

コストに応じて
質が変動



AIと学んだこと(発見と理解)③

最終的には、すべて自然言語→プログラムに変換されるため、そのための事前知識が必要。



項目	説明
レキシコン (語彙/字句)	if、print、(、)のような「 そもそもどんな単語（トークン）が存在するか 」を定義する段階。実装では**字句解析器（lexer / tokenizer）**が担当し、文字の並びを意味のある最小単位（トークン）に分割する。
シンタックス (構文)	定義されたトークンを どの順番でどう組み合わせれば正しい文（構文）になるかを定めるルール 。実装では**構文解析器（parser）**が担当。
セマンティックス (意味)	正しく構文解析されたプログラムが実際に何をするか（ 実行時の振る舞い ）を定める。

【創造のワクワク】

いままでやったことない
テスト作ってみよう
に取り組んでいます。

試せば試すほど、「ワクワク」はどんどん増えています。

1. 【発見】のワクワク

- 「全チームのテストプロセス、成果物を分析して統合するためのルールを見つけよう！」

2. 【攻略・達成】のワクワク

- 「FBを受けたその日に、自動テストの実行スピードを50秒→11秒に改善した。」

3. 【創造】のワクワク

- 「Webブラウザ、モバイルアプリだけでなく、ゲームアプリのテスト実行環境を構築しよう！」

4. 【予測・戦略】のワクワク

- 「テストケースとテストスクリプトが一致していないのであれば、プロセス毎にチェックすることで防ぐようにしよう！」

5. 【共創・繋がり】のワクワク

- 「開発チームが作っている仕様駆動ツールと連携できるようにしよう！」

6. 【変身・なりきり】のワクワク

- 「AIEージェントの成果物チェック機構にペルソナを利用しよう！」 etc,...,



5.まとめ

AI時代のQA/テストエンジニア個人の学習ロードマップ°

壁や落とし穴を回避して、**AI活用型QAエンジニア**になりましょう！

1

基礎を固める

AIの不完全なアウトプット
を見抜く基礎理論
を身に着ける



2

AIを使いこなす

思い通りにAIを動かす
指示 & レビューをする



3

開発全体を見渡す

ミスを防ぐための
仕様確認と安全な修正



4

組織と社会を守る

安全性と信頼性を担保する
ルールと仕組み



まとめ

- **QA/テストエンジニアの役割はAI時代に大きく変容しつつある**
 - **役割の消滅ではなく「高度化」**
 - 繰り返し作業はAIへ移譲し、人間は探索的テスト、リスク評価、品質戦略設計に集中する。
 - 「テストを書いて実行する人」から「AIを監督する品質責任者」へ。
 - **Human-in-the-Loopは不可欠**
 - AIにテストを任せても要求や仕様検証は人間が担う必要がある。
- **AIは最強の学習 & 遊び相手**
 - 「負けない」より「遊ぶ」マインドでまずは活用してみませんか。
- **ワクワクが最強の学習戦略 — 今日から1つ、試してみよう！**
 - この後のセッション、講演で「**ワクワクすること**」があれば、試せるものは試しちゃいましょう！
 - とにかく試す、考えて試す、考えて試し続けるのが大事！

