JSTQB
Japan Software Testing Qualifications Board

速くなるほど、深く考える

AI時代のQAに学ぶ

JSTQB 技術委員 小林依光

自己紹介：小林依光

- JSTQBの技術委員＆広報担当
- プロダクト開発組織のQAマネージャー
- 書籍の共著者
  Building Quality
  ビジネス成長を加速する1人目QAからの品質戦略
- 日科技連 コミュニティのアドバイザー
  「ソフトウェア品質保証プロフェッショナルの会」



© Japan Software Testing Qualifications Board

AIの活用で、開発は速くなった？

- 2025 DORA State of AI-assisted Software Development report

各個人の生産性に対する影響の認識

80%以上の回答者が、
AIによって生産性が向上したと回答

ソフトウェアデリバリーの状態

スループットは向上
ただしデリバリーの不安定性は
依然として上昇

- Faros AI Engineering Report Q3 2026

タスク処理能力

開発者1人あたり69.0%向上

PRに紐づくタスク完了数（チームあたり）
が127.3%増加

レビュー・QAの時間

PRがレビュー中にとどまる時間(中央値)
チーム平均で301.4%増加

タスクがQA段階に留まる時間
チーム平均で300.6%増加

※いずれも相関を示す調査で、因果関係を示すものではありません。Farosはベンダー調査です。

出典元
https://dora.dev/research/2025/dora-report/
https://www.faros.ai/research/speed-trap

© Japan Software Testing Qualifications Board

JSTQBのシラバスから考えてみる

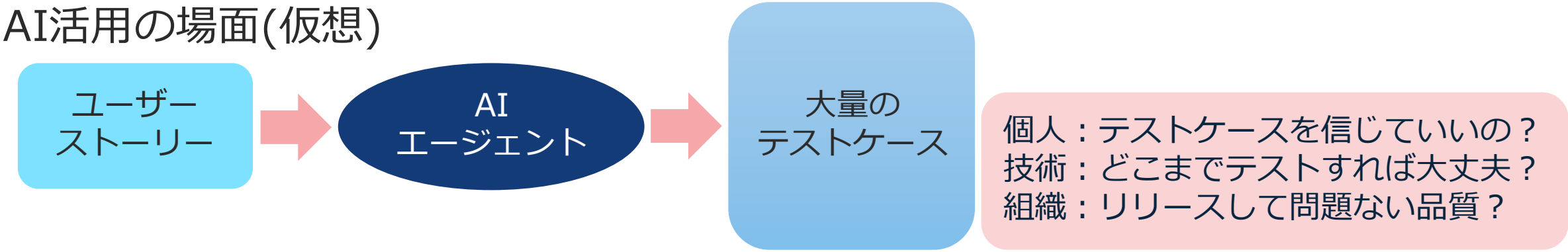
- **テストの原則**　JSTQB Foundation Level シラバス Version 2023V4.0より

1.テストは欠陥があることは示せるが、欠陥がないことは示せない
2.**全数テストは不可能**
3.早期テストで時間とコストを節約
4.欠陥の偏在
5.テストの弱化
6.テストはコンテキスト次第
7.「欠陥ゼロ」の落とし穴

全数テストは不可能！
ごく単純なソフトウェア以外では非現実的

だから、技法・優先順位付け・リスクベースドテストで、
重点を決めている

- AI活用の場面(仮想)



個人：テストケースを信じていいの？
技術：どこまでテストすれば大丈夫？
組織：リリースして問題ない品質？

© Japan Software Testing Qualifications Board

出典元
https://www.jstqb.jp/syllabus/
Foundation Level シラバス Version 2023V4.0

個人：テストケースを信じていいの？

- ECサイトの「注文キャンセルと返金」の例（仮想）

ユーザーストーリー　購入者として、発送前の注文をキャンセルして、返金を受けたい

受け入れ基準

1.発送前の注文だけキャンセルできる
2.キャンセルが完了すると、注文の状態が「キャンセル済み」になる
3.返金は、注文時の支払い方法に対して行われる

AIエージェントが作ったテストケース（4件）

#	操作	期待結果
1	発送前の注文をキャンセルする	注文の状態が「キャンセル済み」になる
2	発送済みの注文をキャンセルしようとする	キャンセルは実行されず、注文の状態は変わらない
3	クレジットカードで支払った発送前の注文をキャンセルする	そのクレジットカードに返金される
4	発送前の注文をキャンセルし、返金状況を確認する	返金は、キャンセルから3営業日以内に完了する

© Japan Software Testing Qualifications Board

AIの出力は、何で評価するか？

- テストタスクの生成AI結果の品質を評価する7つのメトリクス
  JSTQB 生成AIを使用したテスト（CT-GenAI）シラバスより

正確さ	精度	再現率	関連性とコンテキストへの適合性
要件などの基準に対して、全体としてどれだけ合っているか	特定の目的に対して、出力がどれだけ正しいか	必要なテストをもれなく出せているか	この場面・要件に合った内容か(書かれていない条件を持ち込んでいないか)

多様性	実行成功率	時間効率
繰り返しを避けて、幅広くカバーしているか	生成したテストのうち、正常に実行できる割合	人が手で作るより、どれだけ時間を節約できたか

© Japan Software Testing Qualifications Board

出典元
https://www.jstqb.jp/syllabus/
生成AIを使用したテスト（CT-GenAI）シラバス

期待結果を先に書き、AIの外で確かめる

- 根拠を探す仕組みを作る

①期待結果を先に決める（人）

受け入れ基準を
実装前にチームで決める

TDD/BDD/ATDD

②テストケースを作る（AI）

テストケースも期待結果も
AIは作れる

だから、AIの出力だけでは
根拠にならない

③突き合わせる（AIの外の根拠と）

・要件、受け入れ基準
・既知のシステムの動作
・ドメイン専門家

突き合わせる作業はツールを活用してもOK
基準はAIの外にある

出典元
https://www.jstqb.jp/syllabus/
Foundation Level シラバス Version 2023V4.0
アジャイルテクニカルテスト担当者 シラバス
生成AIを使用したテスト（CT-GenAI）シラバス

© Japan Software Testing Qualifications Board

技術：どこまでテストすれば大丈夫？

- AIが作った大量のテストケースを全部、実行しますか？

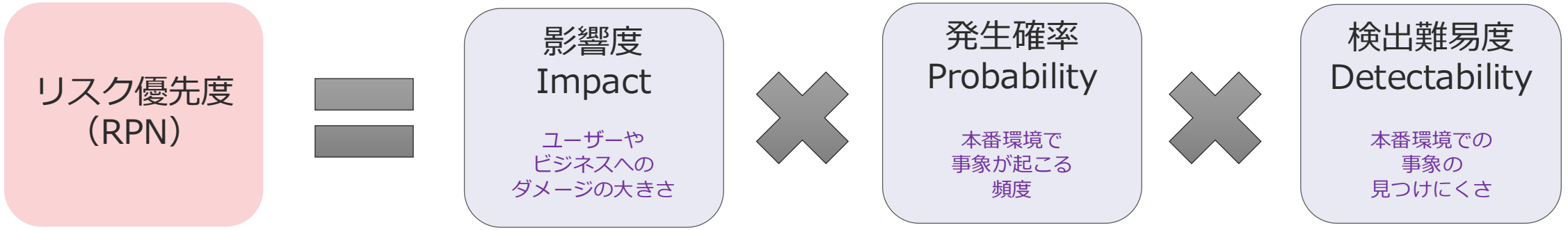
徹底的に確かめるなら、全部実行したい	時間がかかるので、1日に何度も回せない
でも、速く、何度でも確認したい	CIでは確認を増やすほど遅くなる、バランスが必要
「どこまで」「どの深さで」を決める必要がある	例（返金）　浅い：正常な返金を1件 深い：返金が失敗したとき、二重操作、端数の金額まで
AIを活用して順番（優先度）をつけてもらえばいい？	AIの順番付けにも、間違い（推論エラー）が起こりうる 人が確かめる基準が必要

© Japan Software Testing Qualifications Board

出典元
https://www.jstqb.jp/syllabus/
アジャイルテクニカルテスト担当者 シラバス
生成AIを使用したテスト（CT-GenAI）シラバス

リスクに応じて、テストの深さを決める

- リスクの点数のつけ方の例：RPN（Risk Priority Number）、FMEAの考え方



リスク優先度と事例	自動テスト 機械が決まった手順を繰り返す	ブラックボックステスト 人が仕様通りか確かめる	探索的テスト 人が動かしながら探る
高：返金機能	Should	Should	Must
中：注文の状態表示	Could	Could	Must
低：画面の文言	Won't	Won't	Must

AIが作った量ではなく、リスクでテストの深さ（スコープ）を決める

※安全に関わるシステムでは組み合わせが変わる

出典元
https://www.jstqb.jp/syllabus/
アジャイルテクニカルテスト担当者 シラバス
Foundation Level シラバス Version 2023V4.0
Advanced Level シラバス テストマネジメント

© Japan Software Testing Qualifications Board

組織：リリースして問題ない品質？

- AIが作ったテストを実行した。テストは通った　→　リリースしてよい？

テストの終了基準（先に決めておく、テストを終えてよい条件）

例：返金の場面（仮想）
- ☑ 計画したテストを、すべて実行した
- ☑ 直っていない重大な不具合が、残っていない
- ☑ 見つかった不具合は、すべて報告・共有した

アジャイルでは「完了の定義（DoD）」と呼ぶことが多い

リリース基準（出してよい条件）

例：返金の場面（仮想）
- ・問題が起きたとき、すぐ戻せるか
- ・返金の問い合わせに、対応できるか
- ・残るリスクを関係者が理解して受け入れるか（時間切れのときも）

テストの終了基準を満たしても、
リリースしてよいかは、別の判断

テストの結果だけでなく、リリース後のことも含めている

© Japan Software Testing Qualifications Board

出典元
https://www.jstqb.jp/syllabus/
Foundation Level シラバス Version 2023V4.0

AIを使う開発では、基準を見直す

- 基準を見直し、組織で合意する

①テストの終了基準を見直す
・AIが作ったテストについて、AIの外の根拠を使って確かめているか？

②問題が起きたとき、戻せるかを見直す
・AIは同じ指示でも、結果が変わることがある。AIに直させても、同じ誤りが、また出ることがある
・人が確かめた状態（前回のリリース）を残して、そこに戻せるか？

③「リスクが低い」の判断を、見直す
・「リスクが低いから、テストは浅くてよい」の判断を、AIにさせると、AIの推論エラーが混ざりうる
・「リスクが低い」は、人が、根拠をもって決めたか？

出典元
https://www.jstqb.jp/syllabus/
生成AIを使用したテスト（CT-GenAI）シラバス
大規模アジャイルテストのリーダーシップ (CT-ATLaS) シラバス

© Japan Software Testing Qualifications Board

まとめ：速くなるほど、深く考える

- JSTQBのシラバスから、導いてみた事例

個人の問い

テストケースを信じていいの？

AIの外の根拠で、
確かめる

技術の問い

どこまでテストすれば大丈夫？

人が根拠をもって、
リスクで深さを決める

組織の問い

リリースして問題ない品質？

AIを使う開発では
基準を見直して
組織で合意する

3つとも、一人では答えが出ない → 対話で、根拠を一緒につくる
「プロセスやツールよりも個人と対話を」

皆さんのチームでは、どう答えますか

速くなった分を、深く考えることに回す

© Japan Software Testing Qualifications Board



ありがとうございました

https://www.jstqb.jp/