SQiP ソフトウェア品質シンポジウム 2026
2026年9月10日

AIと品質: AIシステムの品質保証とAIによるソフトウェア品質保証

鷲崎 弘宜

ソフトウェア品質管理研究会運営小委員会委員長

IEEE Computer Society, 2025 President

早稲田大学 研究推進部副部長・教授

国立情報学研究所 客員教授

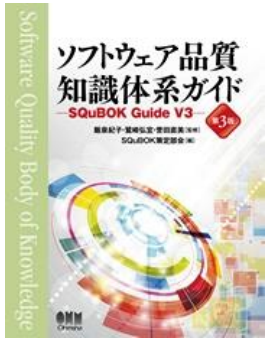
(株)エクスモーション 取締役, (株) SI&C 顧問

http://www.washi.cs.waseda.ac.jp/

目次

AIシステムの品質保証: 品質特性とマネジメント

AIベースシステムの品質



AIシステムにおける品質の概念

- 訓練・テストデータの品質
- モデルの品質
- 機械学習システムの品質

AIシステムの品質マネジメント

（PoC、監視、オンライン学習、更新など）

AIシステムの品質保証技術

- 疑似オラクル
- メタモルフィックテスティング
- 頑健性検査
- ニューロンカバレッジ
- 説明生成

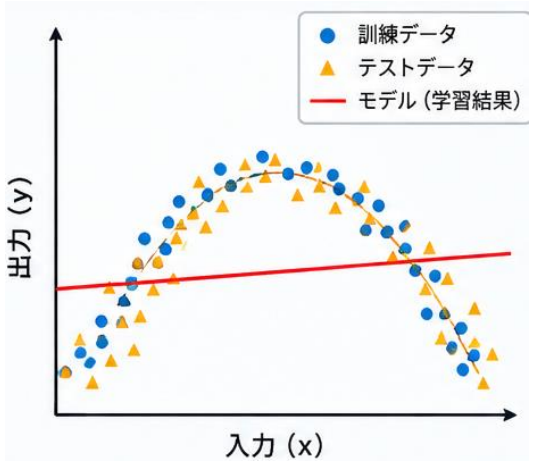
飯泉紀子、鷲崎弘宜、誉田直美 監修、SQuBOK策定部会 編、“ソフトウェア品質知識体系ガイド（第3版）－SQuBOK Guide V3－”、ISBN 978-4-274-22631-1、2020/11/21、オーム社

AIシステムの品質の概念

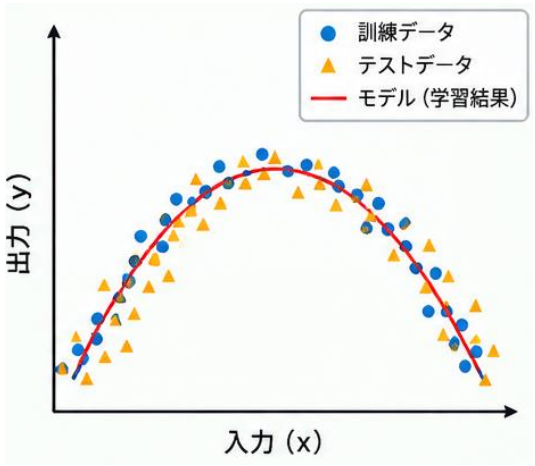
- データの品質
  - コンセプト・データドリフト
- モデルの性能
  - 正解率、混同行列
  - ROC、AUC
  - 未学習、過学習
- モデルの頑健性
- モデルの説明可能性
- システムの品質
  - KPI
  - 仮説検定

		予測	
		陽性 Positive	陰性 Negative
正解	陽性	真陽性 True positive	偽陰性 False negative
	陰性	偽陽性 False positive	真陰性 True negative

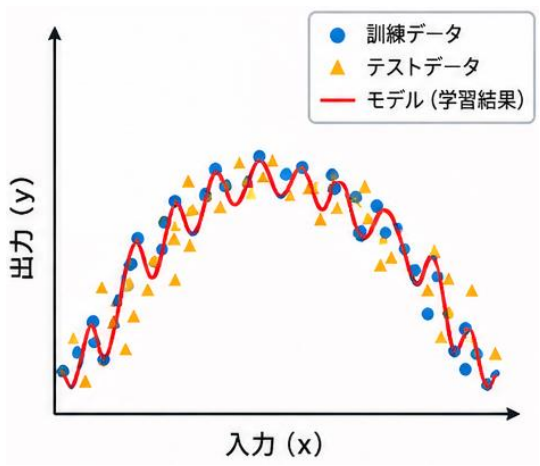
未学習



適切



過学習



飯泉紀子、鷲崎弘宜、誉田直美 監修、SQuBOK策定部会 編、“ソフトウェア品質知識体系ガイド（第3版）－SQuBOK Guide V3－”、2020/11/21、オーム社

ISO/IEC 5259-2分析と機械学習のデータ品質 第2部：データ品質の測定

ISO/IEC 25012 データ品質モデル

- 正確性（Accuracy）
- 完全性（Completeness）
- 一貫性（Consistency）
- 信憑性（Credibility）
- 最新性（Currentness）
- アクセシビリティ（Accessibility）
- 標準適合性（Compliance）
- 機密性（Confidentiality）
- 効率性（Efficiency）
- 精度（Precision）
- 追跡可能性（Traceability）
- 理解性（Understandability）
- 可用性（Availability）
- 移植性（Portability）
- 回復性（Recoverability）

追加 データ特性

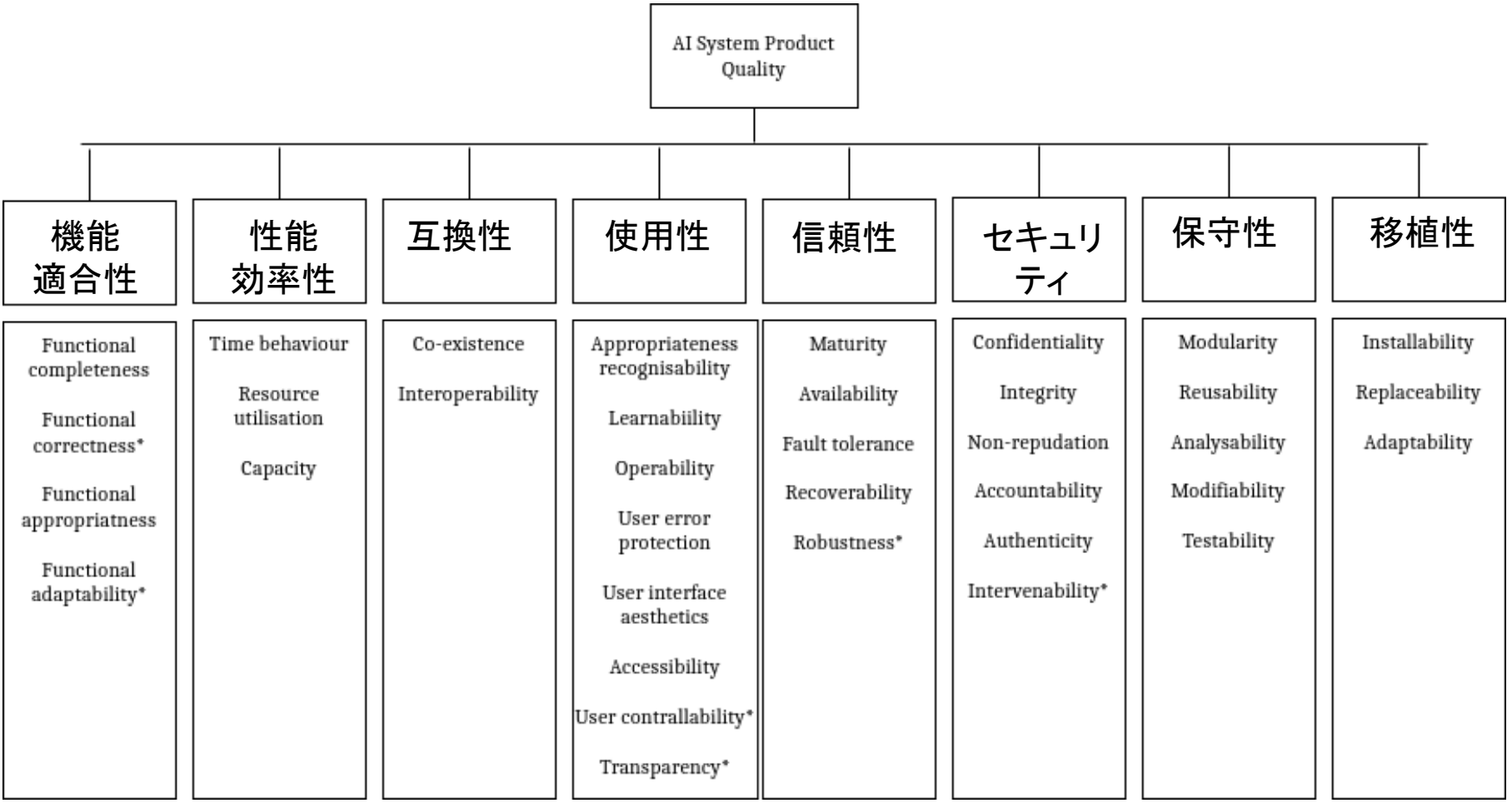
- 関連性: データ値が対象変数の優れた予測因子であること
- データスケーラビリティ: データ量と処理速度が増加してもデータ品質が維持されること
- 状況網羅性
- 適時性: データの処理速度
- 識別可能性: 個人データ保護のため（25012では「機密性」の配下）
- 監査可能性: 関連ステークホルダーがデータを利用可能

追加 データセット特性

- バランス: サンプルの分布を機械学習モデルで学習可能、許容可能な性能を達成可能、訓練データセットと実世界の間に差異が無い
- データセットの有効性: 特定の機械学習プロセス使用のための要件群を満たしていること
- 類似性: 関心のある特徴量におけるサンプル間の類似性
- 代表性: サンプルが母集団を反映していること

D. Natale, Liaison JTC1/SC7 to SC42 Artificial Intelligence, SC7, 2022/6/9

ISO/IEC 25059:2023
AIシステムのプロダクト品質モデル



機能適合性
- 機能正確性
- 機能適合性

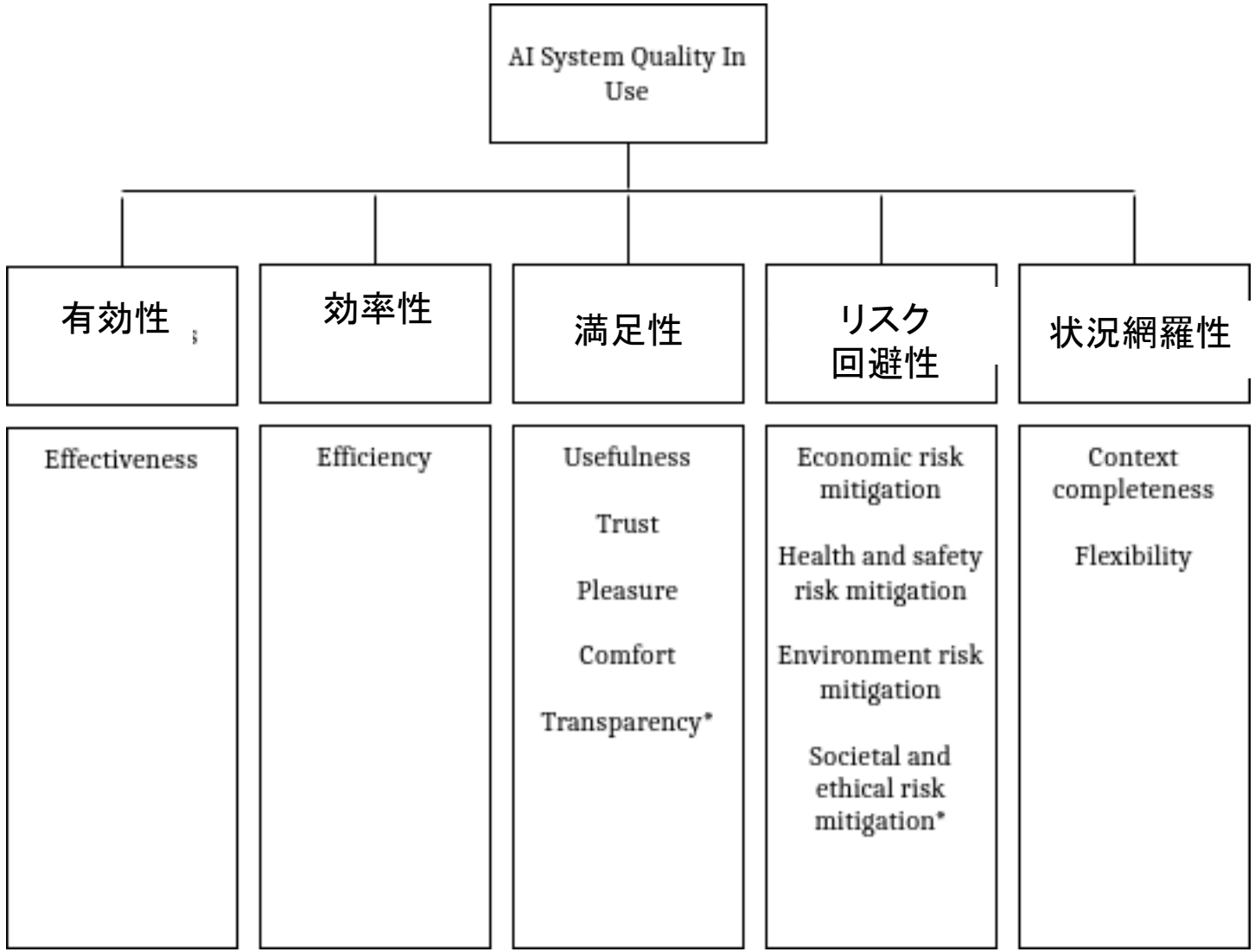
使用性
- ユーザ制御性
- 透明性

信頼性
- 堅牢性

セキュリティ
- 介入性

D. Natale, Liaison JTC1/SC7 to SC42 Artificial Intelligence, SC7, 2022/6/9

ISO/IEC 25059:2023
AIシステムの利用時の品質モデル



D. Natale, Liaison JTC1/SC7 to SC42 Artificial Intelligence, SC7, 2022/6/9

満足性

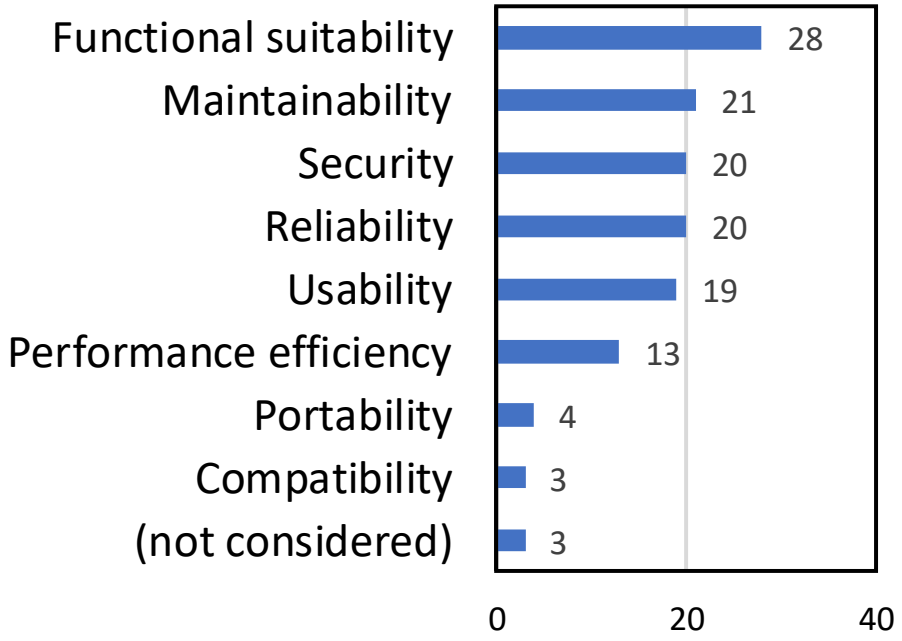
- 透明性

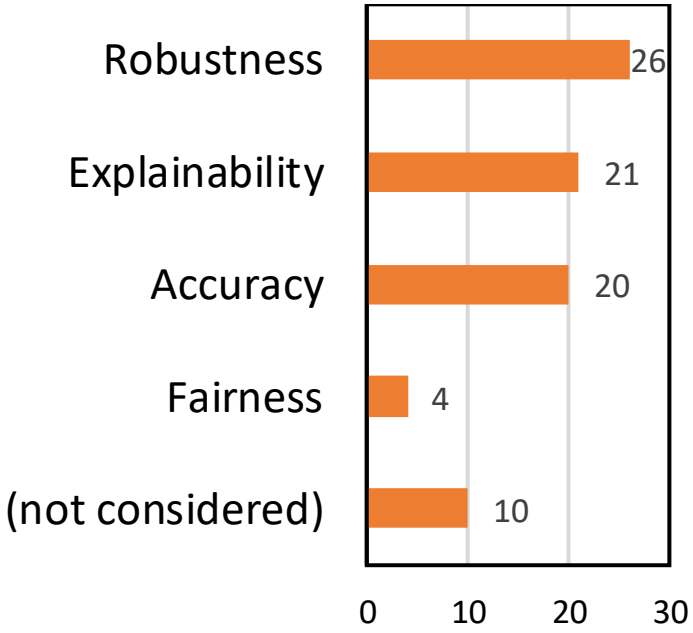
リスク回避性

- 社会的・倫理的リスク緩和性

実務家への調査 [ICSME'20]

- 300名以上を対象に調査、46名が機械学習開発について回答
- どのような製品の品質属性が考慮されているか？
  - 保守性、信頼性、セキュリティ、および使いやすさ
- どのようなモデルの品質属性や予測の品質属性が考慮されているか？
  - 頑健性、精度、および説明可能性



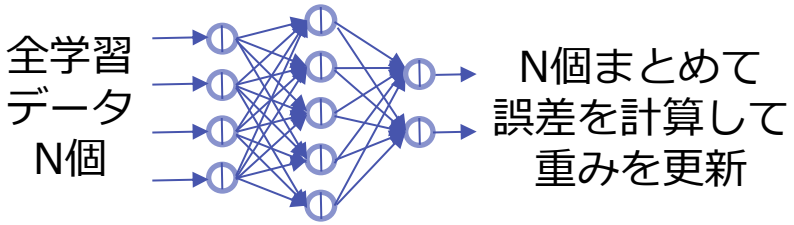


H. Washizaki, et al., Practitioners' insights on machine-learning software engineering design patterns: a preliminary study, ICSME 2020

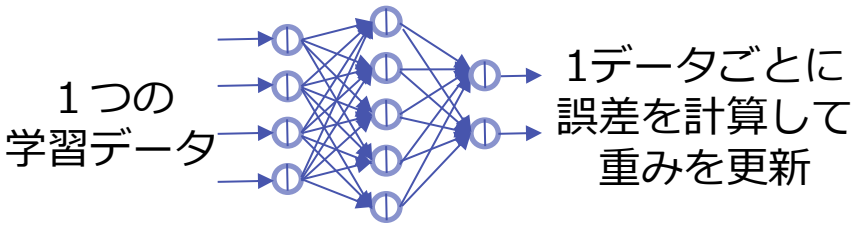
AIシステムの品質保証マネジメント

- 試験的・探索的と顧客協力
- 監視と性能劣化検出
- オンライン学習と検証
- 成果物・版管理
  - データ、プログラム、ハイパーパラメータ

バッチ学習



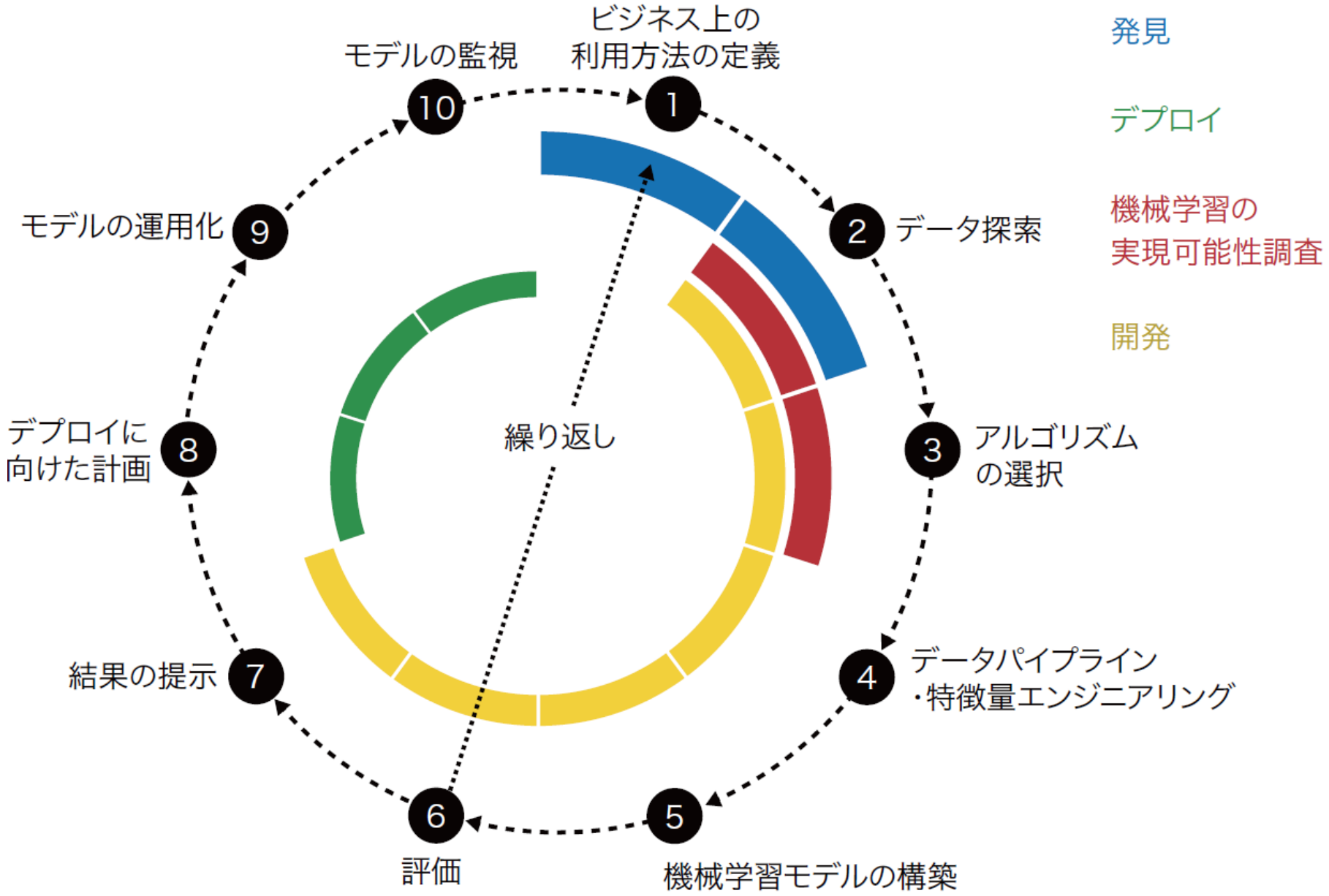
オンライン学習



増倉孝一,スマートIoTシステムビジネス入門, スマートエスイー講義資料, 2018

飯泉紀子、鷲崎弘宜、誉田直美 監修、SQuBOK策定部会 編、“ソフトウェア品質知識体系ガイド（第3版）－SQuBOK Guide V3－”、2020/11/21、オーム社

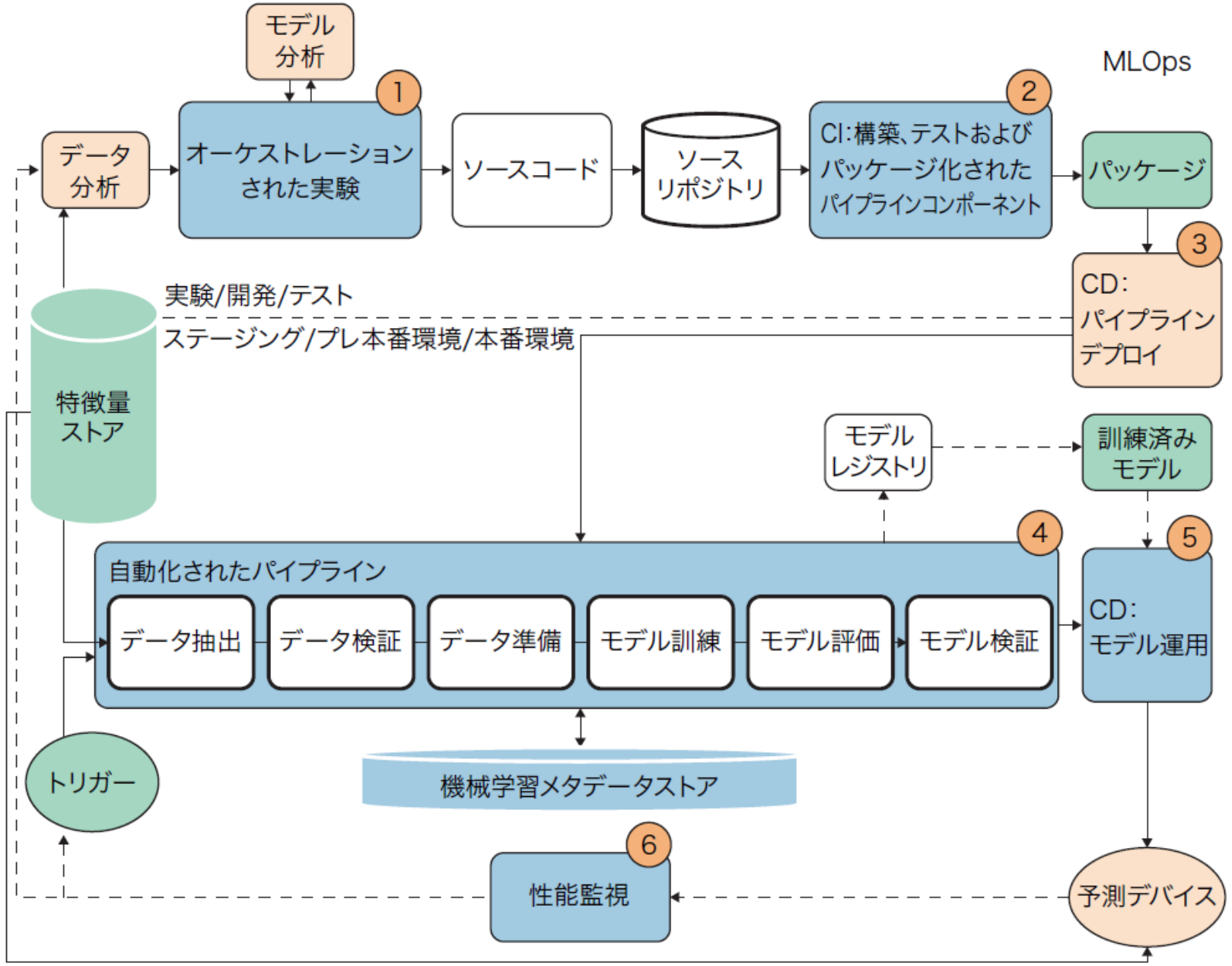
機械学習モデルのライフサイクル



『機械学習デザインパターン ―データ準備、モデル構築、MLOpsの実践上の問題と解決』Lakshmananら著, 鷲崎ら訳, オライリージャパン, 2021

機械学習モデルの扱い：MLOpsと自動化

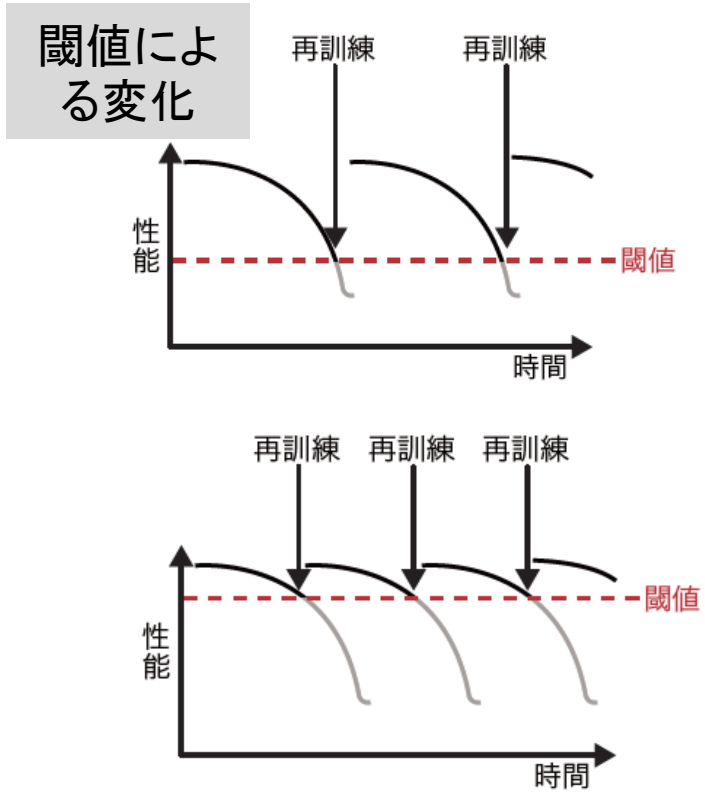
1. 戦術的（手動）
2. 戦略的（パイプライン）
3. **変革的（自動化）**

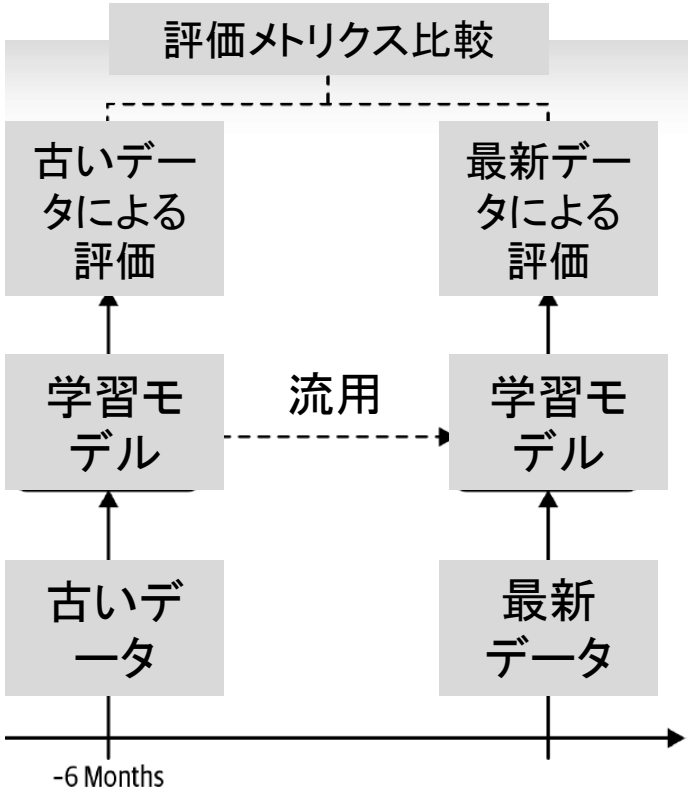


『機械学習デザインパターン ―データ準備、モデル構築、MLOpsの実践上の問題と解決』Lakshmananら著, 鷲崎ら訳, オライリージャパン, 2021

継続的モデル評価 Continued Model Evaluation

- 問題: モデルはデプロイした瞬間から劣化していく。背景にある前提の変化に伴う入力と目標の関係の変化（コンセプトドリフト）、入力データからの予測時データの変化（データドリフト）。
- 解決: 継続的なモデル評価・モニタリングと再訓練
- 考慮
  - 閾値による性能と訓練コストのトレードオフ
  - オフラインでの古いデータによる学習モデルの新データへの適用評価

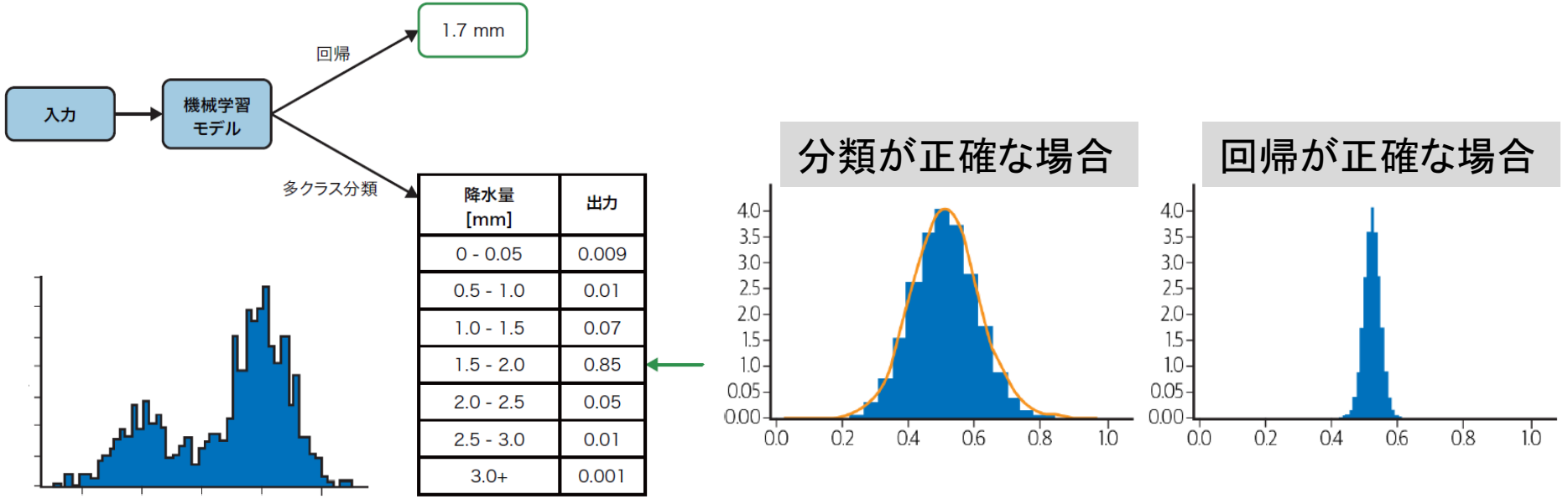




『機械学習デザインパターン ―データ準備、モデル構築、MLOpsの実践上の問題と解決』Lakshmananら著, 鷲崎ら訳, オライリージャパン, 2021

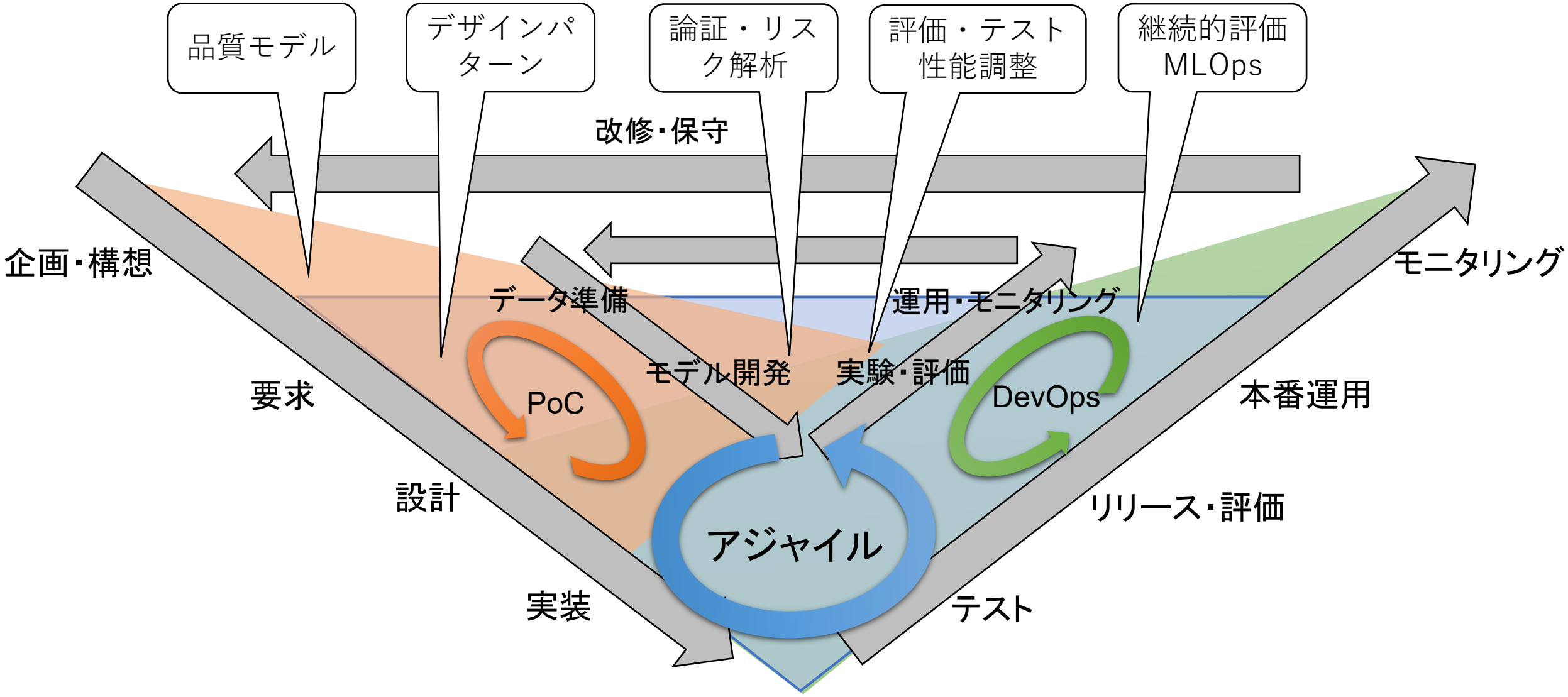
問題再設定 Reframing

- 問題: 確率的であるため、特に精緻な実数の予測に固執すると当初の目的変数や出力形式では限界を生じる場合が有る（特に分布上のピークが複数の場合）。
- 解決: 問題を捉えなおす。特に回帰から分類へ変更する。
- 考慮:
  - 数値が重要な場合は逆に回帰へ（例: 都市そのものより緯度・経度）
  - 分類への移行方法: 回帰結果による分類、マルチタスク学習
  - 真に目的にかなうものか注意（例:おすすめ映画の種類からユーザのクリック数へと予測問題を変更すると、過剰にクリック数の増大方向）



『機械学習デザインパターン ―データ準備、モデル構築、MLOpsの実践上の問題と解決』Lakshmananら著, 鷲崎ら訳, オライリージャパン, 2021

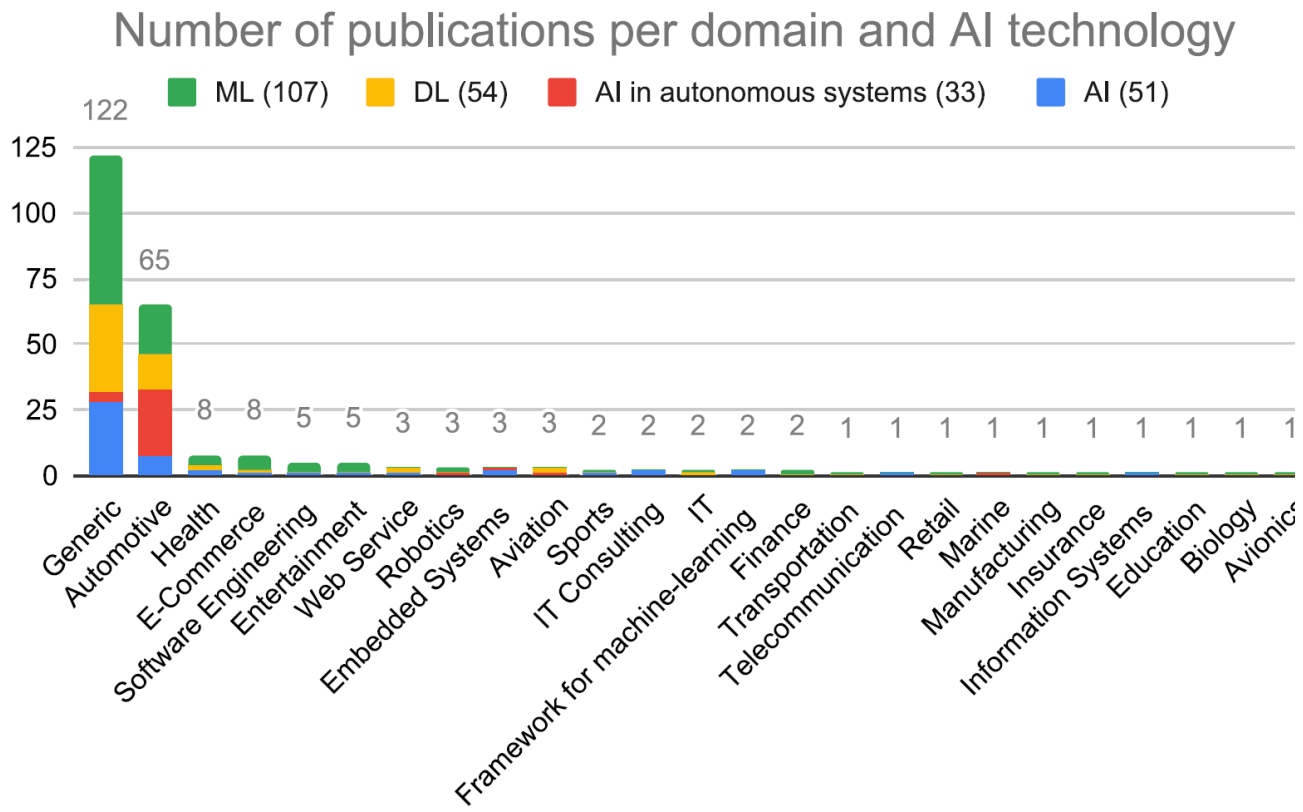
AIシステム開発運用プロセスの例と対応技術

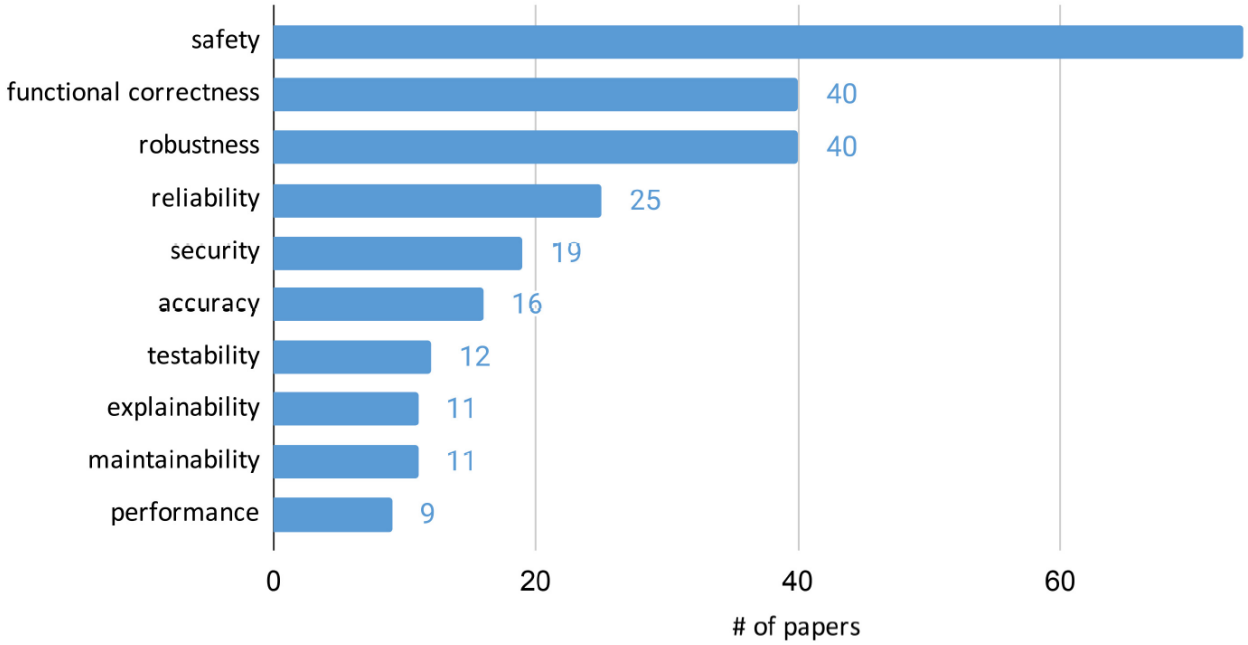


AIシステムの品質保証: 設計による品質

AIシステムのソフトウェアエンジニアリング動向

- 調査: Software Engineering for AI-Based Systems: A Survey (TOSEM'22)
  - 2010-2020に248編の論文、うち2/3以上は2018年以降
  - 対象: 一般、ならびに自動車が大部分
  - 品質特性: ディペンダビリティ、セーフティの扱いが多い
  - 領域: テスト、品質が多い。保守などは軽視。データ関連が頻出研究課題。





Silverio Martínez-Fernández, Justus Bogner, Xavier Franch, Marc Oriol, Julien Siebert, Adam Trendowicz, Anna Maria Vollmer, Stefan Wagner: Software Engineering for AI-Based Systems: A Survey. ACM Trans. Softw. Eng. Methodol. 31(2): 37e:1-37e:59 (2022)

領域別の動向

領域	数	傾向
要求	17	• AI特化の新品質特性への注目、確率的結果や曖昧さを扱う仕様（例：部分仕様） • AIのための要求工学プロセスの取り組みは限定的
設計	34	• 安全性や信頼性などの設計戦略、マイクロサービスとしてのモデル共有など、具体的なAI基盤 • システムレベルではパターン、設計標準、リファレンスアーキテクチャの提案は少ない
構築	23	• 構築上の課題とガイドライン、プラットフォームの提案、成熟度や根拠など不十分
テスト	115	• テストケース（55件）、うちテストケースの生成（40）と選択（12）。カバレッジ提案（14） • 手法: メタモルフィックテスト最多（16件）、ファジング（6）、ミューテーションテスト（5）
保守	6	• 少数。AIシステムのバグやデバッグ、リペア関連。
プロセス	31	• 学際的チーム形成、モデルレベルでのAIパイプラインに焦点を当てた研究あり（6件） • プロセス関連のサポートとしてツール（6）やフレームワーク（3）
手法	38	• AIコンポーネントを用いたCPSの検証と妥当性確認（18件） • セーフティ、自動運転領域の取り組みが多い
品質	59	• ISO 25000やISO 26262などの国際規格の更新必要性の提起 • 多くはMLの品質特性、フレームワーク、品質保証、認証に焦点
他	4	• マネジメント、経済、構成管理、プロフェッショナルプラクティス

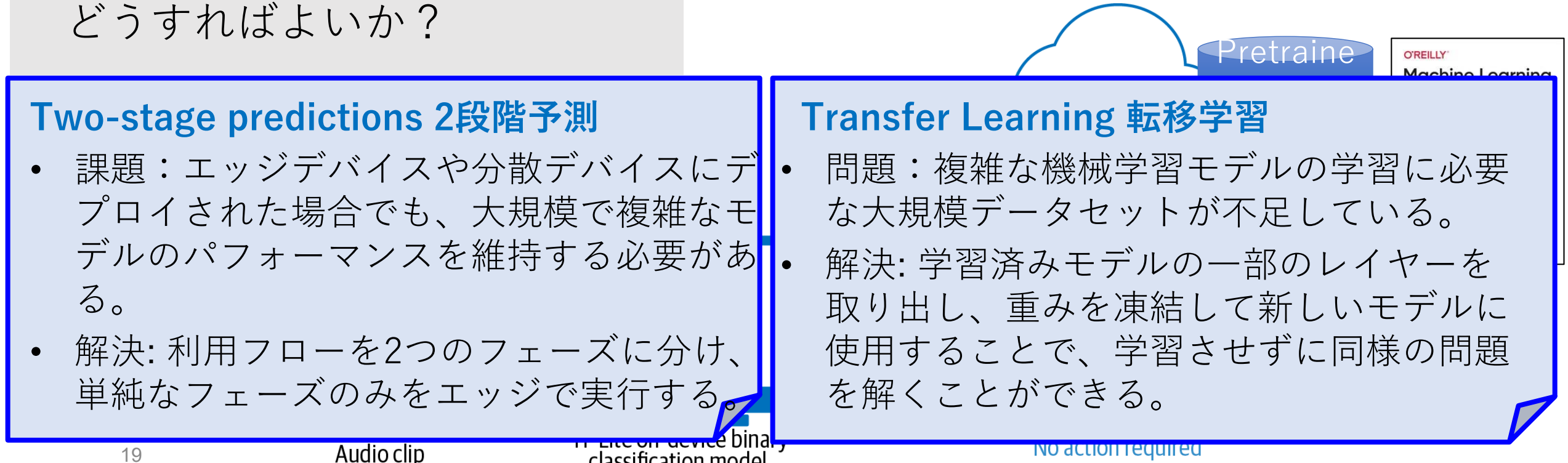
Silverio Martínez-Fernández, et al., Software Engineering for AI-Based Systems: A Survey. ACM Trans. Softw. Eng. Methodol. 31(2): 37e:1-37e:59 (2022)

機械学習システムの設計例

- スマホが拾った音から楽器の種類を特定し、その種類に応じた録音・応答を実現したい。
- しかし、スマホのメモリや性能には限りがあり、大規模なディープラーニング・モデルは搭載できそうにない。

どうすればよいか？

- 携帯電話上の小さなモデルが音が楽器かどうかを判断し、クラウド上の大きなモデルが楽器である場合のみ音の種類を分類するという2段階予測を使用しよう。
- 大規模モデルには、正確な分類を実現するために転移学習を採用しよう。



Two-stage predictions 2段階予測

- 課題：エッジデバイスや分散デバイスにデプロイされた場合でも、大規模で複雑なモデルのパフォーマンスを維持する必要がある。
- 解決: 利用フローを2つのフェーズに分け、単純なフェーズのみをエッジで実行する。

Transfer Learning 転移学習

- 問題：複雑な機械学習モデルの学習に必要な大規模データセットが不足している。
- 解決: 学習済みモデルの一部のレイヤーを取り出し、重みを凍結して新しいモデルに使用することで、学習させずに同様の問題を解くことができる。

Machine Learning Design Patterns (V. Lakshmanan, et al. 2020)

AI・機械学習ソフトウェアエンジニアリングパターン

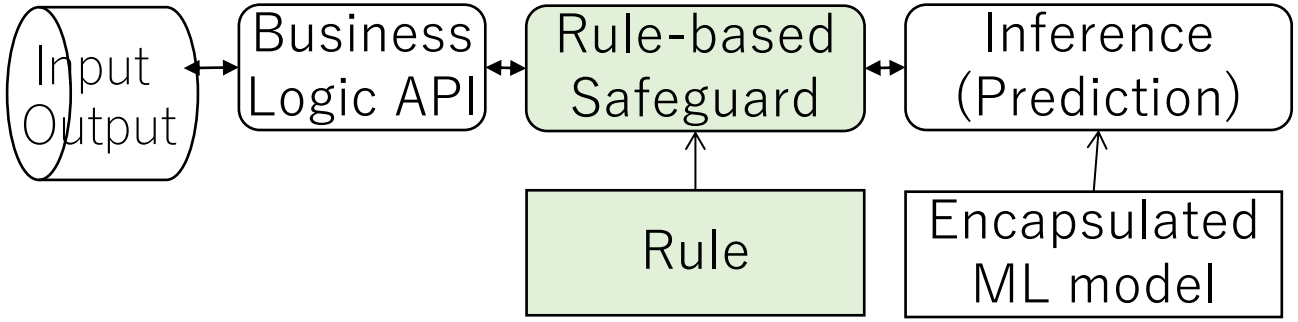
- アーキテクチャとデザインパターン
  - MLアプリケーションのためのソフトウェア工学パターン[SEP4MLA]
  - 機械学習デザインパターン[MLDP]
- 論証パターン
  - MLシステムのための安全性ケースパターン[Safety]
  - DNNのためのセキュリティ論証パターン[Security][Security2]
- レスポンシブル（責任ある）AIパターン
  - 責任あるAIシステムのデザインパターン [Responsible]
- 開発と管理のプラクティス
  - ライフサイクルフェーズのプラクティス[Practice][Practice2]
- プロンプトパターン
  - プロンプト・パターン・カタログと分類法 [Prompt][Prompt2]

[MLDP] V. Lakshmanan, et al., “Machine Learning Design Patterns,” O’Reilly, 2020
[SEP4MLA] H. Washizaki, et al. “Software Engineering Design Patterns for Machine Learning Applications,” IEEE Computer 55(3) 2022
[Safety] E. Wozniak, et al., “A Safety Case Pattern for Systems with Machine Learning Components,” SAFECOMP 2020 Workshop
[Security] M. Zeroual, et al., “Security Argument patterns for Deep Neural Network Development,” PLoP 2023
[Security2] M. Mutsche, et al. “Robustness-based Security Case Verification for Deep Neural Networks,” AsianPLoP 2024
[Responsible] Q. Lu, et al., “Responsible-AI-by-Design: a Pattern Collection for Designing Responsible AI Systems,” IEEE Software, 2023
[Practice] M. S. Rahman, et al., “Machine Learning Application Development: Practitioners’ Insights,” Software Quality Journal, 31, 2023
[Practice2] Y. Watanabe, et al., “Preliminary Literature Review of Machine Learning System Development Practices,” COMPSAC 2021
[Prompt] J, White, et al., “A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT,” arXiv 2302.11382, 2023
[Prompt2] Y. Sasaki, et al., “Landscape and Taxonomy of Prompt Engineering Patterns in Software Engineering,” IEEE IT Pro 27, 2025

責任あるMLシステム設計のためのパターン [Computer'22]

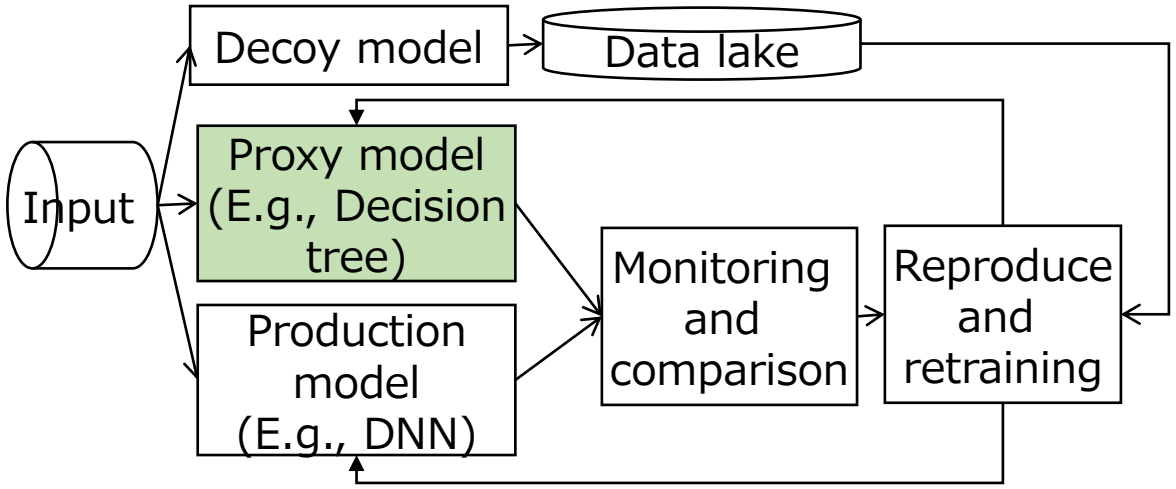
Encapsulate ML Models within Rule-based Safeguards ルールベースの安全ガードによるMLモデルカプセル化

- 問題：MLモデルは不安定で、敵対的攻撃、ノイズ、データドリフトに弱いことが知られている。
- 解決: MLモデルによって提供される機能をカプセル化し、決定論的で検証可能なルールを用いて、システムに内在する不確実性に対処する。



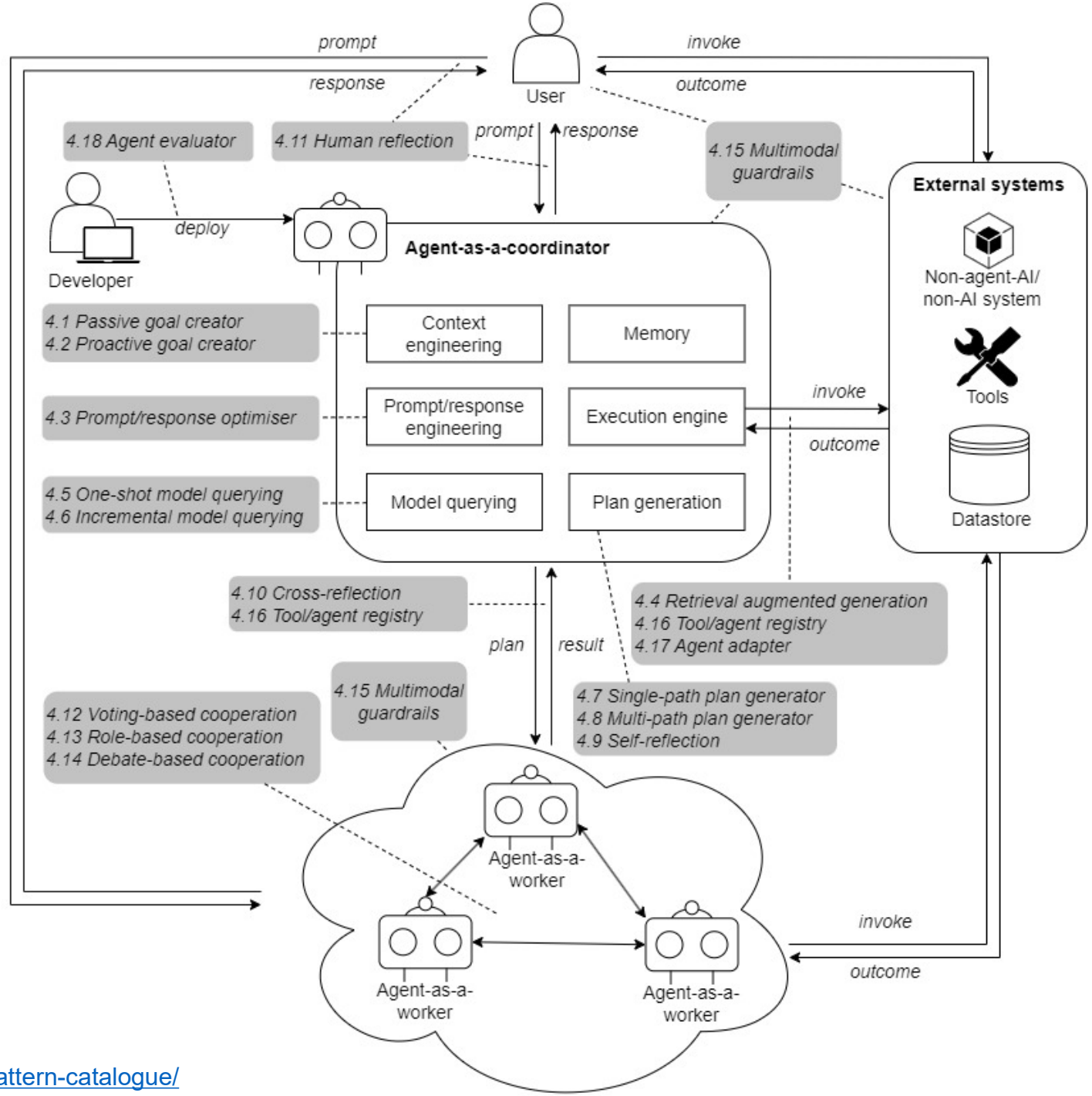
Explainable Proxy Model 説明可能な代理モデル

- 問題：説明可能な代理MLモデルを構築しなければならない。
- 解決 説明可能な推論パイプラインを主推論パイプラインと並行して実行し、予測の違いを監視する。



Hironori Washizaki, Foutse Khomh, Yann-Gael Gueheneuc, Hironori Takeuchi, Naotake Natori, Takuo Doi, Satoshi Okuda,
"Software Engineering Design Patterns for Machine Learning Applications," IEEE Computer, Vol. 55, No. 3, pp. 30-39, 2022. **(Best Paper Award)**

LLMエージェント デザインパターン



CSIRO, Agent Design Pattern Catalogue
https://research.csiro.au/ss/science/projects/agent-design-pattern-catalogue/

各エージェントデザインパターンの概要

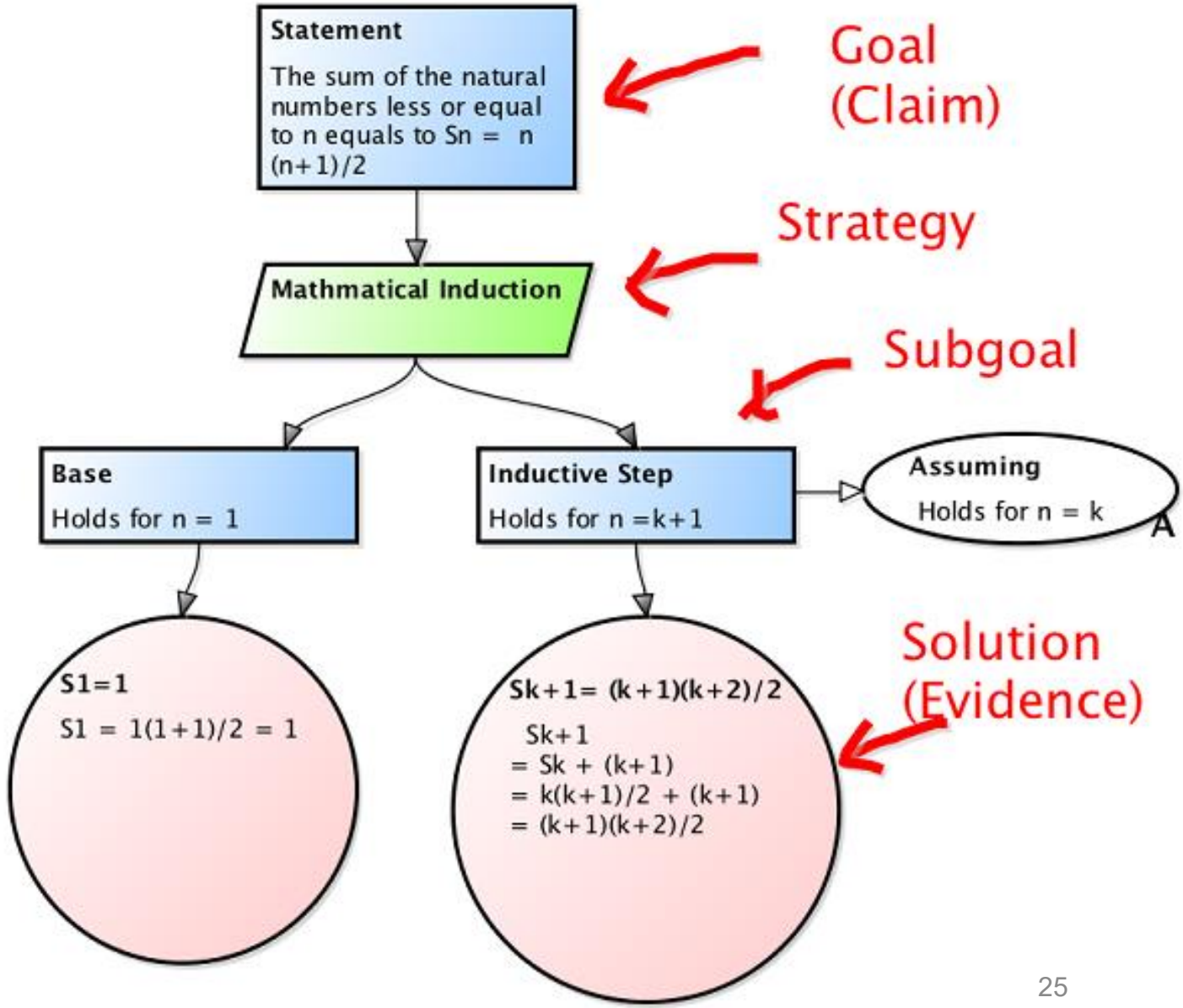
ID	パターン名	概要
4.1	受動的目標生成	対話インターフェースを通じてユーザが表明した目標を分析
4.2	能動的標生成	人の相互作用を理解し、関連ツールを通じて文脈を把握することで、ユーザの目標を予測
4.3	プロンプト/レスポンス最適化	希望する入力または出力の内容と形式に基づいてプロンプト/レスポンスを調整
4.4	RAG	目標達成に向けたエージェントの知識更新可能性を高め、オンプレミスLLMエージェント／システム実装におけるデータプライバシーを維持
4.5	ワンショット・モデルクエリ	LLMは単一のインスタンスでアクセスされ、計画に必要な全ステップを生成
4.6	漸増的モデルクエリ	計画生成プロセスの各ステップでLLMにアクセス
4.7	単一経路計画生成	ユーザの目標達成に至る中間ステップの生成を調整
4.8	マルチパス計画生成	ユーザの目標達成に至る各中間ステップにおいて複数の選択肢を作成することを可能に
4.9	自己省察（リフレクション）	自ら計画と推論プロセスに関するフィードバックを生成し、自ら改善の指針を提供
4.10	相互省察	異なるエージェントやLLMを用いてフィードバックを提供し、生成された計画と対応する推論手順を精緻化
4.11	人による省察	エージェントは人からのフィードバックを収集し、計画を洗練させることで、人の選好に効果的に適合
4.12	投票ベース協調	エージェントは自由に意見を表明し、投票に基づく協調を通じて合意を形成
4.13	役割ベース協調	エージェントには様々な役割が割り当てられ、その役割に基づいて決定が確定
4.14	議論ベース協調	エージェントは他のエージェントからフィードバックを受け取り、合意が形成されるまで他のエージェントとの議論の中で思考と行動を調整
4.15	マルチモーダルガードレール	LLMの入力と出力を制御し、ユーザ要求、倫理基準、法令などの特定の要求を満足
4.16	ツール／エージェント・レジストリ	多様なエージェントやツールを選択するための統一された便利なレジストリを維持
4.17	エージェントアダプタ	タスク完了のためにエージェントと外部ツールを接続するためのインターフェースを提供
4.18	エージェント評価者	多様な要求と指標に基づいてエージェントを評価するためのテストを実施

CSIRO, Agent Design Pattern Catalogue https://research.csiro.au/ss/science/projects/agent-design-pattern-catalogue/

AIシステムの品質保証: 論証とリスク解析

論証ベース・アシュアランスケース

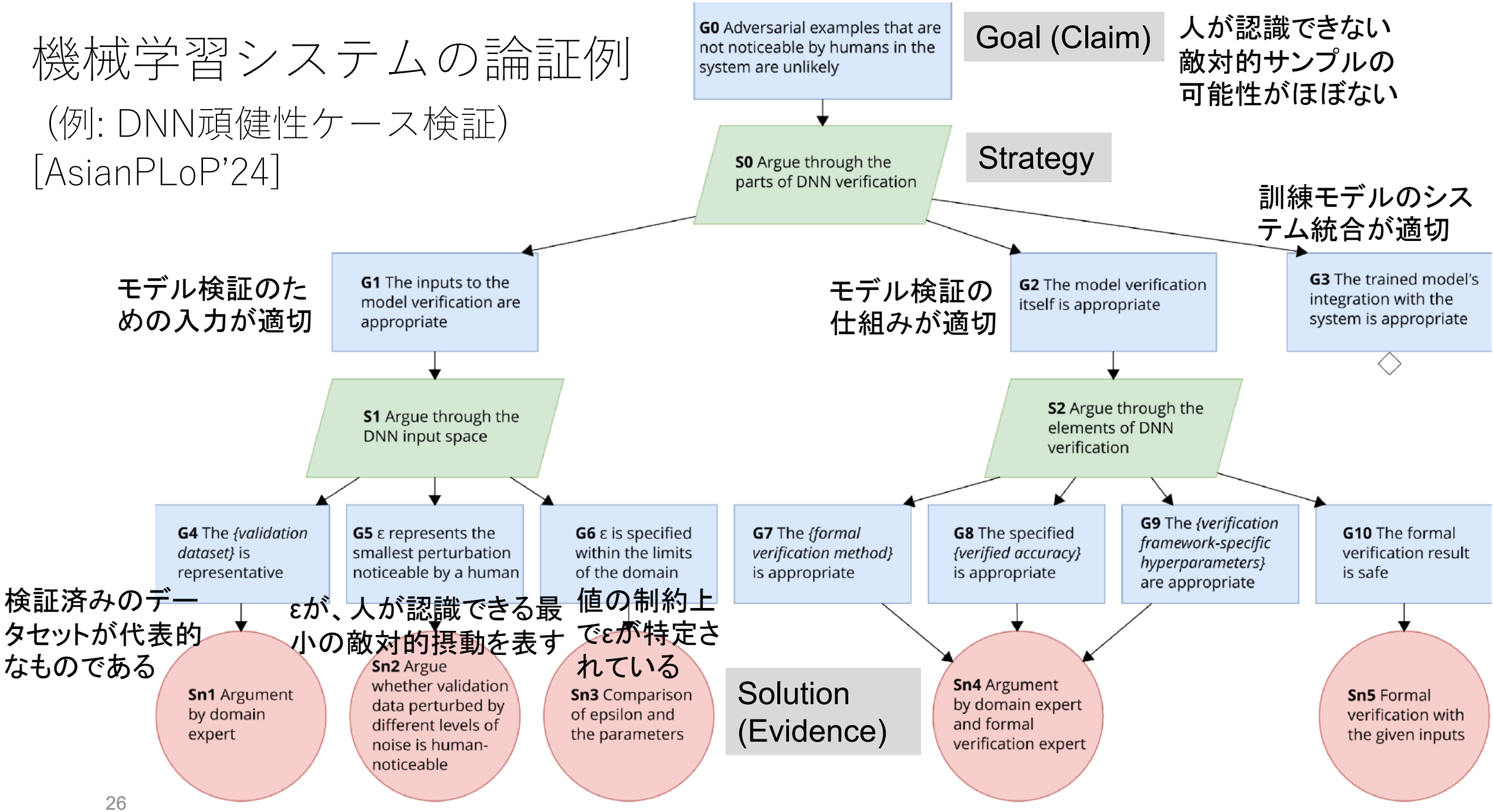
- アシュアランスケース
  - 構造化された主張、議論、証拠により、あるシステムが、保証される必要のある特定の品質や特性を有していることを確信させるもの
  - 安全性、セキュリティ、レジリエンス、…
- ゴール構造表記法（GSN）
  - 重要な特性を保証する論拠を可視化するためのグラフィカルな表記法
  - AIアプリケーションのリスクベース評価採用例：UK ICO AI説明可能性ガイド



Jordi Cabot, Goal Structuring Notation – a short introduction
https://modeling-languages.com/goal-structuring-notation-introduction/

機械学習システムの論証例

（例: DNN頑健性ケース検証）
[AsianPLoP'24]

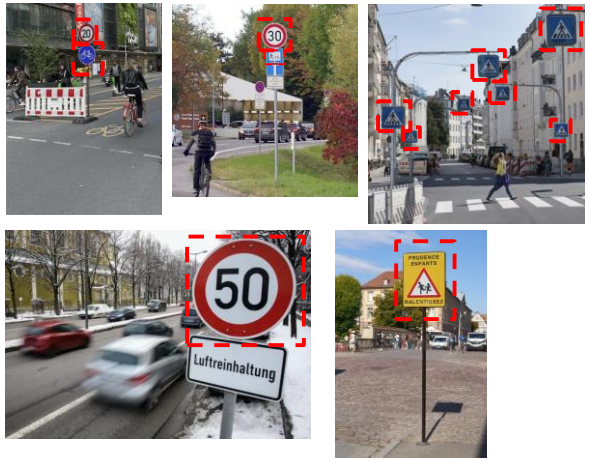


M. Mutsche, et al. "Robustness-based Security Case Verification for Deep Neural Networks," AsianPLoP 2024

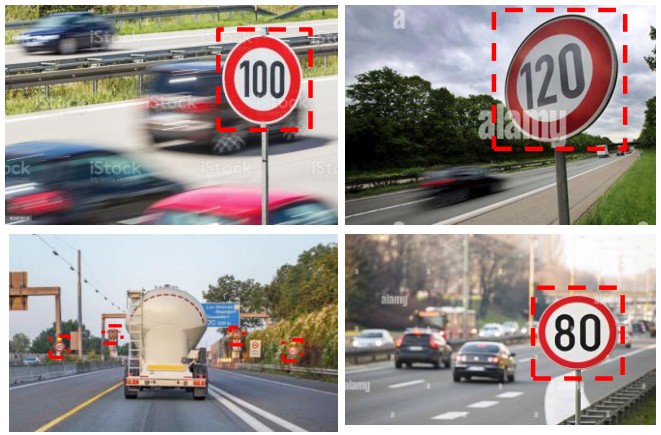
交通標識の認識システムの例

街中と高速道路とで信頼性・安全性を考慮して許容可能な認識精度の機械学に基づくシステムをどのように開発、改訂できるか？

街中



高速道路



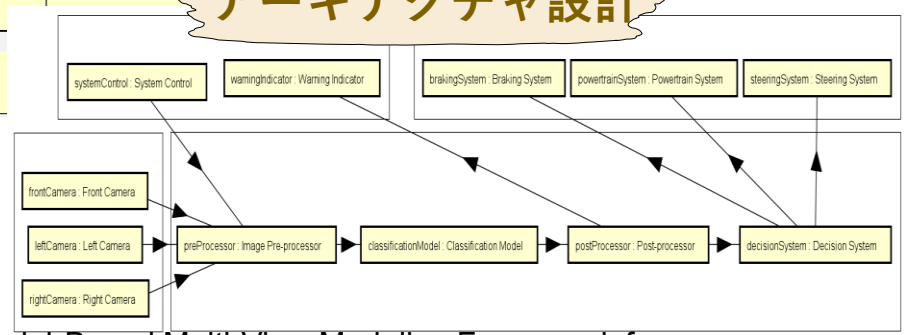
AIプロジェクトキャンバス

Data	Skills	Value Proposition	Integration	Customer
AI.D1 - Color images of traffic signs in operation domain データ	AI.Sk1 - Computer vision スキル	AI.VP1 - Provide a reliable traffic sign classification system for level 3 ADV system 価値提案	AI.I1 - Car sensor system 統合 AI.I2 - Car self driving control system AI.I3 - Car user interface	AI.Cu1 - Self-driving car drivers 顧客 AI.Cu2 - Self-driving car passengers
	Output AI.O1 - Accuracy of overall classification 出力 AI.O2 - Accuracy, precision, and misclassification rate in highway domain AI.O3 - Accuracy, precision, and misclassification rate in suburbian domain		Stakeholders AI.St1 - Traffic authorities ステーク AI.St2 - Car makers ホルダ AI.St3 - Other road users	
Cost AI.Co1 - Data labeling fee AI.Co2 - System development cost コスト		Revenue AI.R1 - SaaS subscription fee 収益		

MLキャンバス

Prediction Task	Decision	Value Proposition	Data Collection	Data Source
MLPT1 - Classify traffic signs from camera color images	MLDe1 - Provide the classification result for decision-making ML model	MLVP1 - Provide a reliable traffic sign classification system for level 3 ADV system	MLDC1 - Use publicly available data (GTSRB) for initial development MLDC2 - Utilize data collection during operation for continuous update	MLDS1 - Datalake from operata
Impact Simulation	**Making Prediction**		**Building Models**	**Features**
MLIS2 - Achieve reliablity on highway area by prioritizing important classes MLIS1 - Achieve very high accuracy of overall classification MLIS3 - Achieve reliability on suburban area MLIS4 - Achieve very high precision of overall classification	MLMP1 - Classification will be made in real time		MLBM1 - New version of ML model will be trained and deployed in a monthly cycle	MLF1 - Bouding box for traffic sign
	Live Monitoring			
	MLLM1 - Monitoring of potential data drift			

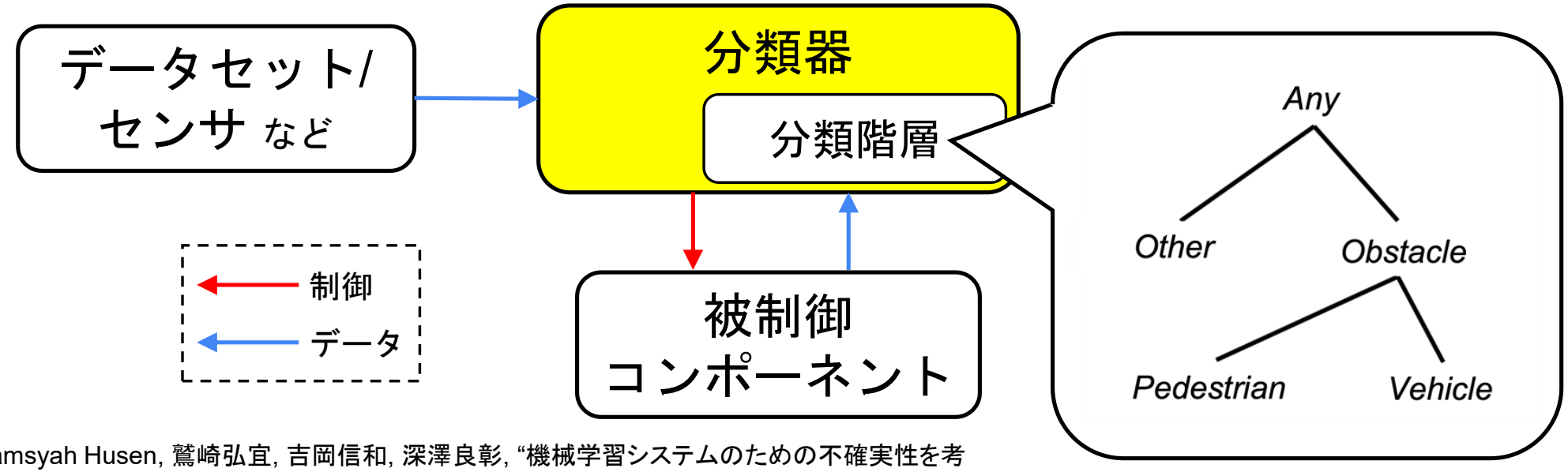
アーキテクチャ設計



Jati Husen, Hironori Washizaki, Nobukazu Yoshioka, Hnin Tun, Yoshiaki Fukazawa and Hironori Takeuchi, “Metamodel-Based Multi-View Modeling Framework for Machine Learning Systems,” 11th International Conference on Model-Based Software and Systems Engineering (MODELSWARD 2023)

安全性リスク解析とリスクマネジメントに向けて

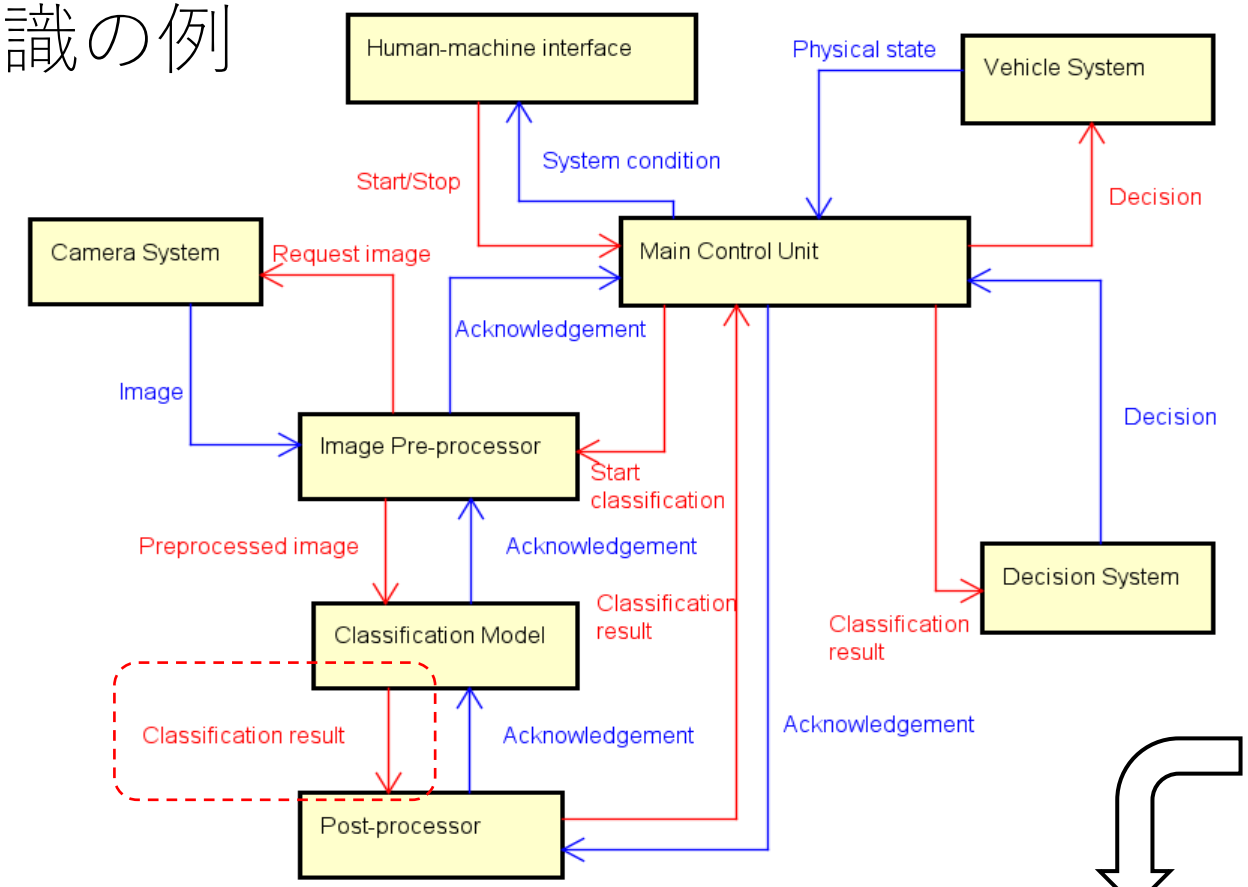
- STAMP/STPA: コンポーネント間の相互作用に着目したリスクの特定
- CFMEA（Classification Failure Mode Effects Analysis）：分類階層を通じた予測の不確実性の扱い＆複数側面を束ねた拡張混同行列
  - “A Safety Analysis Method for Perceptual Components in Automated Driving”, https://ieeexplore.ieee.org/document/8987559
- 組み合わせによる不確実性および多面性考慮のリスク解析とマネジメント



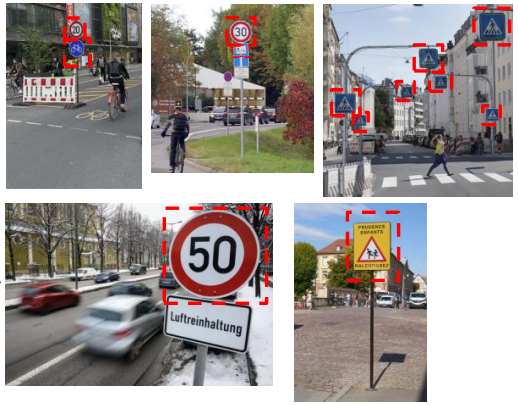
大久保利哉, Jati Hiliamsyah Husen, 鷲崎弘宜, 吉岡信和, 深澤良彰, “機械学習システムのための不確実性を考慮した安全性解析およびリスク定量化手法”, 情報処理学会 第216回ソフトウェア工学研究発表会, 2024

リスク解析 (1/3): 交通標識の例

1. リスク特定
   1. ML分類器や周辺の相互作用着目による非安全制御動作特定
2. リスク分析
   1. 複数側面に基づく拡張混同行列の構築
   2. 一定規模データセットによる分類器の誤分類の頻度測定
   3. 非安全制御動作のリスクレベル決定
3. リスク評価
   1. 非安全制御動作の優先順位付けと対応是非・方針検討
4. リスク対応
   1. 対策の実施（例えばDNNリペア、データ拡張など）
   2. リスク再評価



Providing causes hazard
(UCA2-P-1) Camera providing blurry image
(UCA4-P-1) Classified model misclassified traffic signs for dangerous object [SC1][SC2][SC3]
(UCA4-P-2) Classified model misclassified speed limit signs as higher speed limit [SC4]
(UCA4-P-3) Classified model



UCA4-P-2 (Specific UCAs)
Classification model misclassified “30 km/h” as “80 km/h”
Classification model misclassified “60 km/h” as “20 km/h”

大久保利哉, Jati Hiliamsyah Husen, 鷲崎弘宜, 吉岡信和, 深澤良彰, “機械学習システムのための不確実性を考慮した安全性解析およびリスク定量化手法”, 情報処理学会 第216回ソフトウェア工学研究発表会, 2024

リスク解析 (2/3)

1. リスク特定
   1. ML分類器や周辺の相互作用着目による非安全制御動作特定
2. リスク分析
   1. 複数側面に基づく拡張混同行列の構築
   2. データセットによる誤分類の可能性評価
   3. 非安全制御動作のリスクレベル決定
3. リスク評価
   1. 非安全制御動作の優先順位付けと対応検討
4. リスク対応
   1. 対策の実施（DNNリペア、データ拡張など）
   2. リスク再評価

重大度

Safety Severity	20 km/h	30 km/h	50 km/h	60 km/h	70 km/h	80 km/h	100 km/h	120 km/h
Low speed (~30km/h)	0	0	0	0	0	0	0	0
Middle speed (30~70km	2	1	0	0	0	0	0	0
High speed (80km/h~)	3	3	2	2	1	0	0	0

Progress Severity	20 km/h	30 km/h	50 km/h	60 km/h	70 km/h	80 km/h	100 km/h	120 km/h
Low speed (~30km/h)	0	0	1	1	2	2	3	3
Middle speed (30~70km	0	0	0	0	0	1	2	2
High speed (80km/h~)	0	0	0	0	0	0	0	0

発生可能性

Misclassification Rate	20	30
20 km/h	0.00%	0.00%
30 km/h	6.67%	0.00%
50 km/h	0.00%	1.53%
60 km/h	0.00%	0.00%
70 km/h	1.67%	0.00%
80 km/h	0.00%	0.28%
100 km/h	0.00%	0.00%
120 km/h	10.00%	0.00%
Low Speed Limit	6.67%	0.00%
Middle Speed Limit	1.67%	1.53%
High Speed Limit	10.00%	0.28%
Max speed limit	18.34%	1.81%
Other	0.00%	0.14%

Risk Value	20 km/h	30 km/h
20 km/h	0.00	0.00
30 km/h	0.00	0.00
50 km/h	0.18	1.79
60 km/h	0.18	0.09
70 km/h	3.59	0.09
80 km/h	0.27	4.90
100 km/h	0.27	0.27
120 km/h	5.87	0.27
Other	0.00	0.00
Low Speed Limit	0.00	0.00
Middle Speed Limit	3.95	1.97
High Speed Limit	6.41	5.44
Max speed limit	0.00	0.00

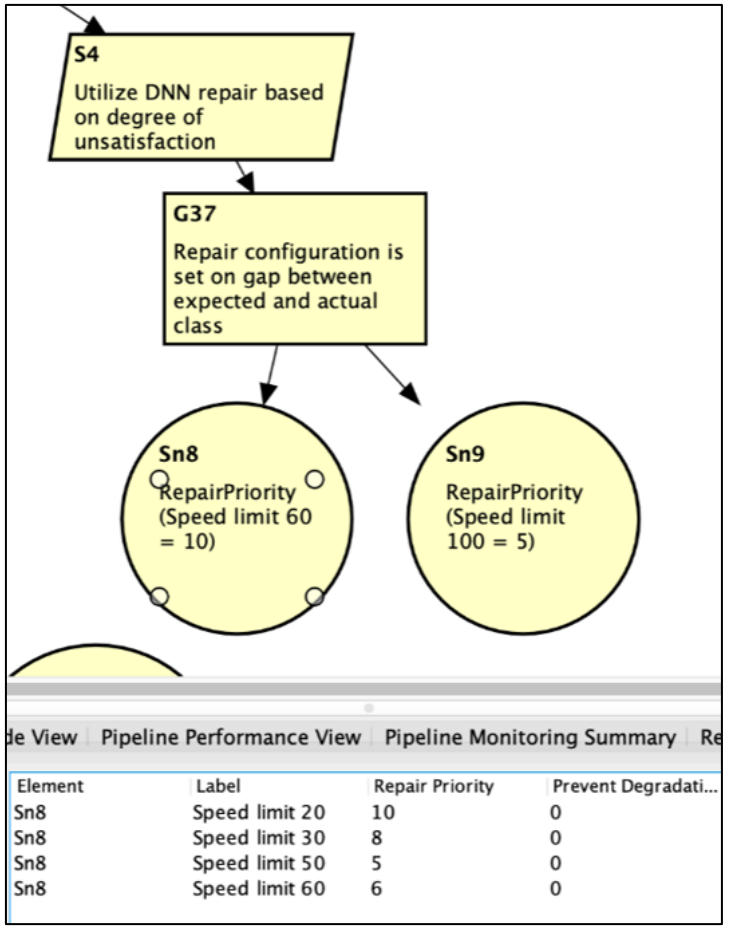
非安全制御動作のリスク

UCAs (Specific-Level)		posure	Controllability	ASIL
Classification model misclassified 20 km/h as		2	3	B
Classification model misclassified 30 km/h as 80 km/h	4.90	3	1	3	A
Classification model misclassified 60 km/h as 80 km/h	3.84	2	2	3	A

リスク解析 (3/3)

1. リスク特定
   1. ML分類器や周辺の相互作用着目による非安全制御動作特定
2. リスク分析
   1. 複数側面に基づく拡張混同行列の構築
   2. データセットによる誤分類の可能性評価
   3. 非安全制御動作のリスクレベル決定
3. リスク評価
   1. 非安全制御動作の優先順位付けと対応検討
4. リスク対応
   1. 対策の実施（DNNリペア、データ拡張など）
   2. リスク再評価

セーフティケースにおけるリスク対策の検討
（例: 20km, 30km, 50km, 60kmの分類精度に重点を置いたDNNリペア）



対策前

UCAs (Specific-Level)	Risk Value	Severity	Exposure	Controllability	ASIL
Classification model misclassified 20 km/h as 120 km/h	5.87	3	2	3	B
Classification model misclassified 30 km/h as 80 km/h	4.90	3	1	3	A
Classification model misclassified 60 km/h as 80 km/h	3.84	2	2	3	A

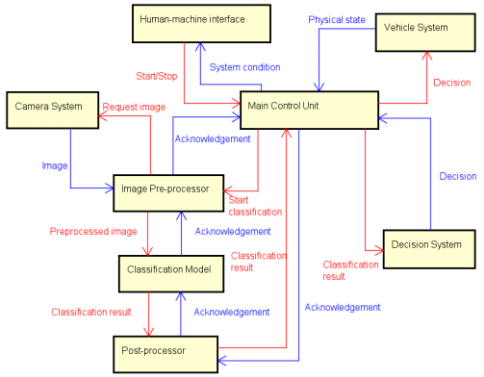
対策後

UCAs (Specific-Level)	Risk Value	Severity	Exposure	Controllability	ASIL
Classification model misclassified 20 km/h as 120 km/h	0.27(-5.60)	3	0	3	QM
Classification model misclassified 30 km/h as 80 km/h	0.27(-4.63)	3	0	3	QM
Classification model misclassified 60 km/h as 80 km/h	3.71(-0.13)	2	1	3	QM

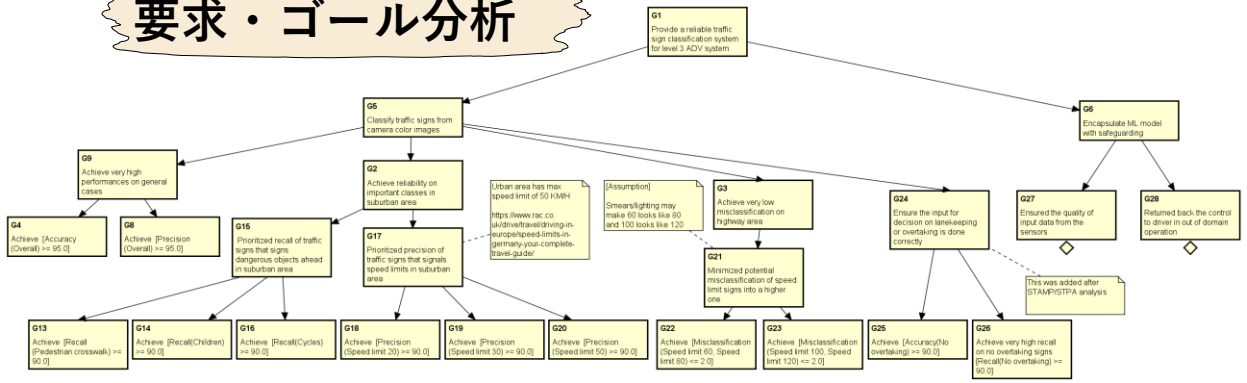
安全性解析

No	CA					
1	Request image	Image Pre-processor	Camera System			
2	Preprocessed image	Image Pre-processor	Classification Model		(UCA2-N-1) One or more camera went offline	(UCA2-P-1) Camera providing blurry image
3	Classification result	Post-processor	Main Control Unit			
4	Classification result	Classification Model	Post-processor			(UCA4-P-1) Classified model misclassified traffic signs for dangerous object [SC1][SC2][SC3] (UCA4-P-2) Classified model misclassified speed limit signs as higher speed limit [SC4] (UCA4-P-3) Classified model misclassified no overtaking signs as other signs [SC5]
5	Start/Stop	Human-machine interface	Main Control Unit			
6	Decision	Main Control Unit	Vehicle System			
7	Classification result	Main Control Unit	Decision System			
8	Start classification	Main Control Unit	Image Pre-processor			

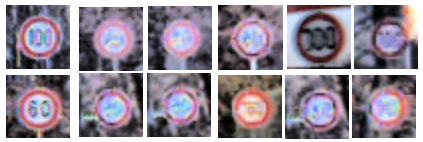
HCFID	HCF	CountermeasureID	Countermeasure	UCA
HCF4-P-1-1	ML model does not have high enough recall for dangerous object traffic signs classes	M1	Ensure the training dataset is representative of operation domain	(UCA4-P-1) Classified model misclassified traffic signs for dangerous object [SC1][SC2][SC3]
HCF4-P-1-1	ML model does not have high enough recall for dangerous object traffic signs classes	M2	Utilize data augmentation to increase the variation of the dataset	(UCA4-P-1) Classified model misclassified traffic signs for dangerous object [SC1][SC2][SC3]
HCF4-P-1-2	Misclassification rate of ML model for speed limit signs is too high	M3	Test ML model with specific test dataset of adversary cases	(UCA4-P-1) Classified model misclassified traffic signs for dangerous object [SC1][SC2][SC3]
HCF4-P-1-2	Misclassification rate of ML model for speed limit signs is too high	M2	Utilize data augmentation to increase the variation of the dataset	(UCA4-P-1) Classified model misclassified traffic signs for dangerous object [SC1][SC2][SC3]
HCF4-P-1-3	ML model does not have high enough recall for no overtaking traffic signs classes	M1	Ensure the training dataset is representative of operation domain	(UCA4-P-1) Classified model misclassified traffic signs for dangerous object [SC1][SC2][SC3]
HCF4-P-1-3	ML model does not have high enough recall for no overtaking traffic signs classes	M2	Utilize data augmentation to increase the variation of the dataset	(UCA4-P-1) Classified model misclassified traffic signs for dangerous object [SC1][SC2][SC3]



要求・ゴール分析



誤分類データ



分類の性能評価

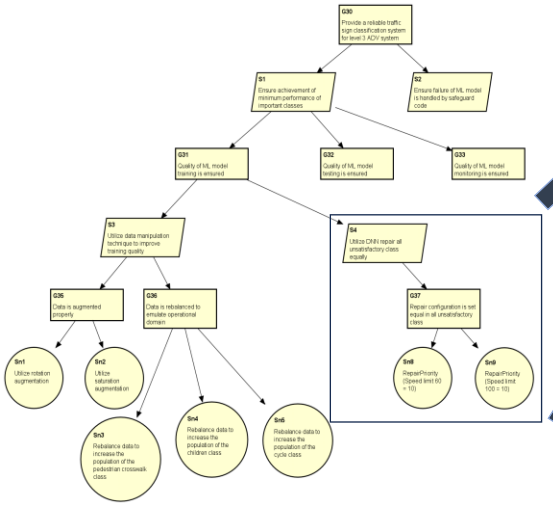
修正に向けた選択

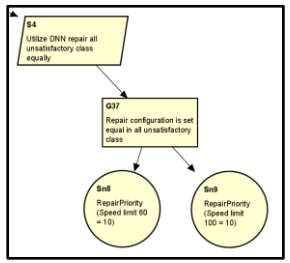
機械学習モデルA

モデルB

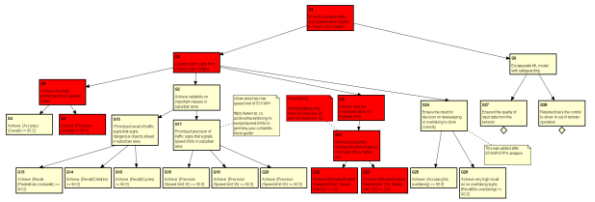
モデルC

安全性・信頼性論証

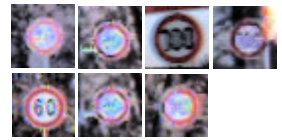


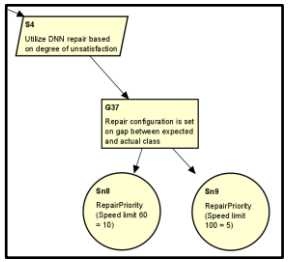


バランスの取れた修正

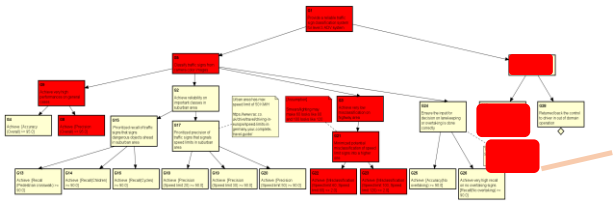


修正結果

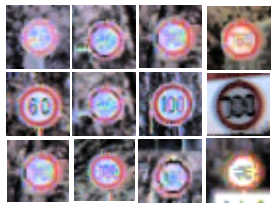




積極的な修正

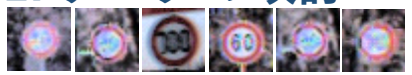


誤分類データ



さらなる改訂へ

1. データの改訂

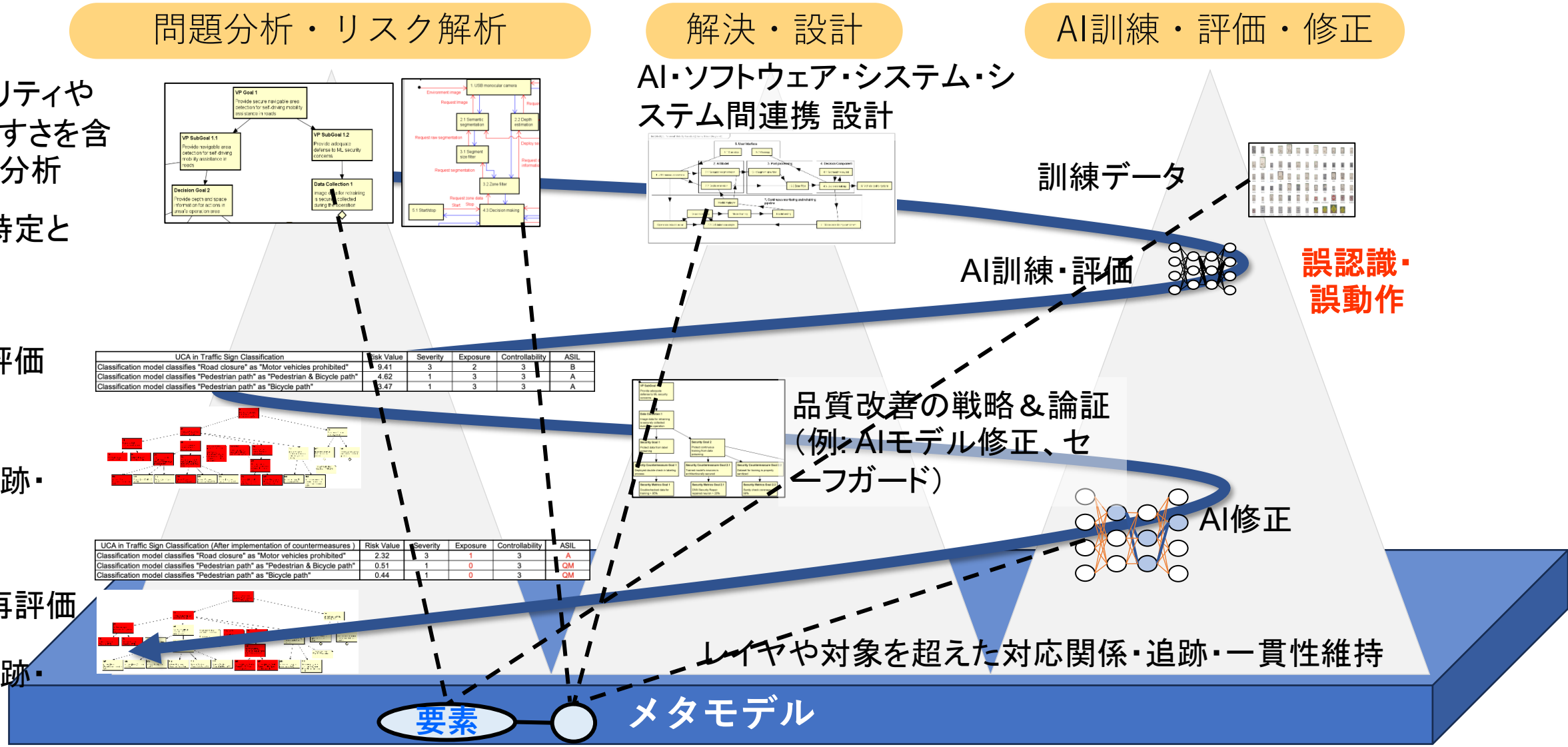


2. 画像の質向上に向けたアーキテクチャ改訂
3. ビジネスゴールの見直し

Jati Husen, Hironori Washizaki, Nobukazu Yoshioka, Hnin Tun, Yoshiaki Fukazawa and Hironori Takeuchi, “Metamodel-Based Multi-View Modeling Framework for Machine Learning Systems,” 11th International Conference on Model-Based Software and Systems Engineering (MODELSWARD 2023)

In this video we will demonstrate our integrated modeling environment

高信頼AIシステム開発のフレームワークの例 (M3S) [SQJ'24]



UCA in Traffic Sign Classification	Risk Value	Severity	Exposure	Controllability	ASIL
Classification model classifies "Road closure" as "Motor vehicles prohibited"	9.41	3	2	3	B
Classification model classifies "Pedestrian path" as "Pedestrian & Bicycle path"	4.62	1	3	3	A
Classification model classifies "Pedestrian path" as "Bicycle path"	3.47	1	3	3	A

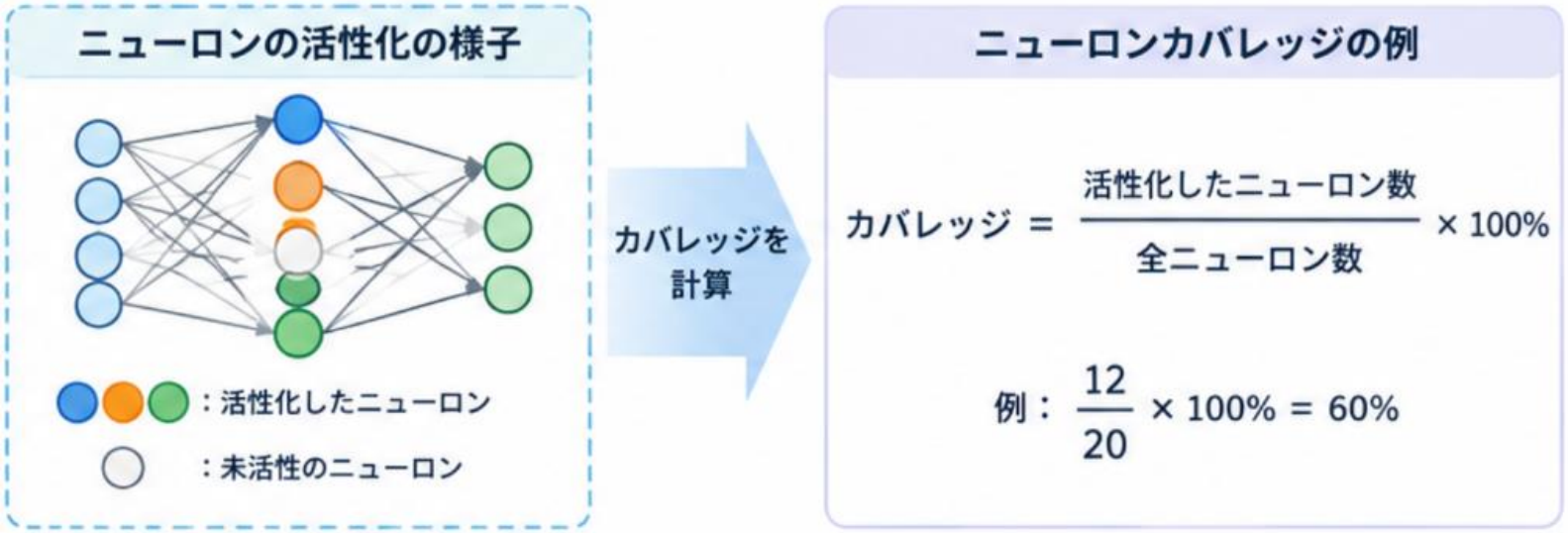
UCA in Traffic Sign Classification (After implementation of countermeasures)	Risk Value	Severity	Exposure	Controllability	ASIL
Classification model classifies "Road closure" as "Motor vehicles prohibited"	2.32	3	1	3	A
Classification model classifies "Pedestrian path" as "Pedestrian & Bicycle path"	0.51	1	0	3	QM
Classification model classifies "Pedestrian path" as "Bicycle path"	0.44	1	0	3	QM

Jati H. Husen, Hironori Washizaki, Jomphon Runpakprakun, Nobukazu Yoshioka, Hnin Thandar Tun, Yoshiaki Fukazawa, Hironori Takeuchi, "Integrated Multi-view Modeling for Reliable Machine Learning-Intensive Software Engineering," Software Quality Journal, Vol. 32, pp. 1239–1285, Springer-Nature, 2024.
https://github.com/jst-qaml/m3s-astah-plugin

AIシステムの品質保証: テスト＆性能調整

AIシステムの品質保証技術

- 疑似オラクル
  - 別の実装や古いバージョンとの比較
  - サーチベースドテスティング
- メタモルフィックテスティング
  - 入力の変化により出力を予想可能な関係によるテスト
  - 例: ノイズを追加しても判定が変わってはならない
- 頑健性検査
- ニューロンカバレッジ
- 説明生成



飯泉紀子、鷲崎弘宜、誉田直美 監修、SQuBOK策定部会 編、“ソフトウェア品質知識体系ガイド（第3版）－SQuBOK Guide V3－”、2020/11/21、オーム社

テストオラクル問題

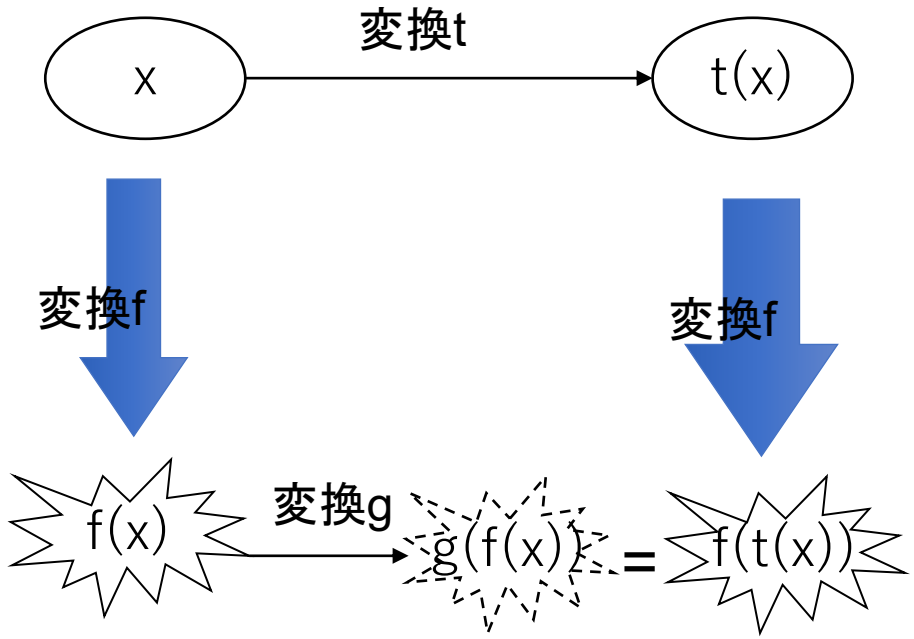
- テストオラクル：テストが合格か不合格かを判定するための仕組み
  - 例：表明（アサーション）、期待される出力、期待される出力を生成するプロセス
  - 問題: 正しい動作と、誤っている可能性のある動作を区別することの難しさ [Barr15]、大規模または複雑なソフトウェアでは特に困難
- 解決策 1：クラウドソーシング
  - Amazon Mechanical Turk を活用し、クラウドワーカーに正しい出力を評価させる [Pastore13]
- 解決策 2：不変条件をテストオラクルとして生成する
  - 出力を監視して不変条件を推論する：例：「結果は常に 0 より大きい」
  - Daikon（MIT開発、C言語向け）http://groups.csail.mit.edu/pag/daikon/
- 解決策3：メタモルフィック・テスティング
  - 異なる入力による複数回の実行に焦点を当てる疑似オラクル・テスト手法

一部ベース: 坂本一憲, “ソフトウェアテスト”, ソフトウェアテスト研修

Earl T. Barr, et al., The Oracle Problem in Software Testing: A Survey, TSE, 41(5), 2015.
Fabrizio Pastore, Leonardo Mariani and Gordon Fraser; , "CrowdOracles: Can the Crowd Solve the Oracle Problem?," Software Testing, IEEE Sixth International Conference on Verification and Validation (ICST), March 2013.

メタモルフィック・テスティング

- メタモルフィック関係に基づき大量に試験
- 入力への変化により、出力の変化を予想できる関係
  - 例: A の検索結果数 ≧ A かつ B の検索結果数
  - 例: sin(x) = sin(x + 360度)



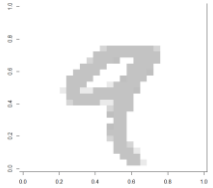
入力変化	出力変化
並び変え	無し
ノイズの追加	
意味的に同じ	
統計的に同じ	
経験的に近い	僅か
定数加乗算	定数の加算、乗算
狭める	部分集合
全く異なる	互いに素

参考: S. Segura et al., "Metamorphic Testing of RESTful Web APIs," IEEE Transactions on Software Engineering, 2017
参考: C. Murphy, “Applications of Metamorphic Testing”, http://www.cis.upenn.edu/~cdmurphy/pubs/MetamorphicTesting-Columbia-17Nov2011.ppt

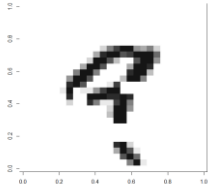
AIシステムのメタモルフィックテスト

文字認識の場合

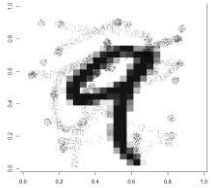
明度



レンズの汚れ



障害物



自動運転の場合 [Tian+ 18]

LLMの場合 [Hyun+ 24]

Original Input #2

Input Text	"Awesome! Then I'll simply disregard your notice."
Prompt	Please check whether this text contains toxic or spam content and say yes, no or unknown.

Perturbed Input #2

Input Text	"Great! In that case, I'll just put aside your notice."
Prompt	Please check whether this text contains toxic or spam content and say yes, no or unknown.

Yes

≠

No

Y Tian, et al., DeepTest: Automated Testing of Deep-Neural-Network-driven Autonomous Cars, ICSE 2018
S. Hyun, et al., "METAL: Metamorphic Testing Framework for Analyzing Large-Language Model Qualities," ICST 2024
参考: 本田澄,アーキテクチャ・品質エンジニアリング, スマートエスイー講義資料, 2018

AIモデルの性能改善・調整（リペア）

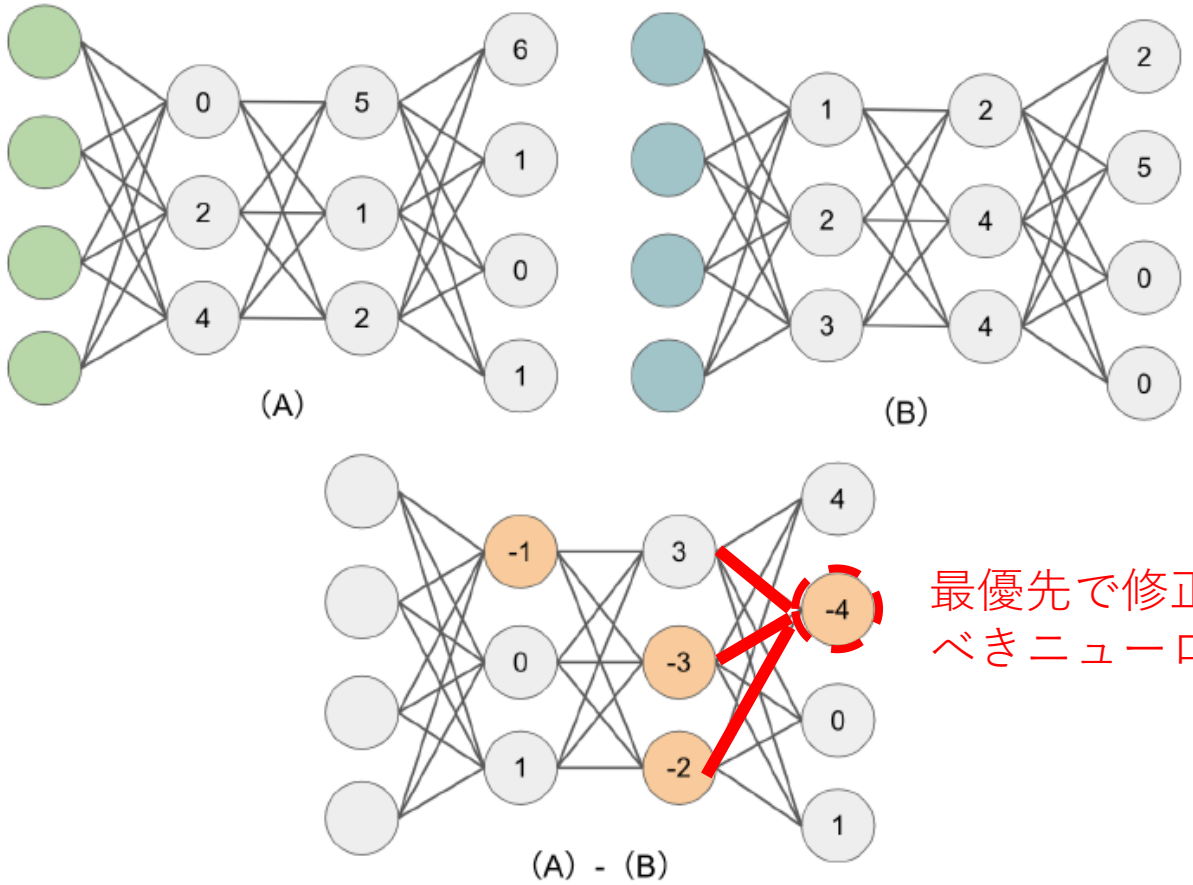
- 再訓練
  - 正攻法であるが、時間・コスト高
  - 追加訓練データの検討の難しさ、性能悪化の可能性
- オンライン学習
  - 特定データによる逐次学習を通じた修正
  - 性能悪化の副作用の可能性
- データ拡充: 生成 [a]、選択 [b]、拡張 [c]など
  - モデルのパラメータを直接には修正せず試行錯誤
  - 敵対的サンプルに対して脆弱な可能性
- 特定サンプルに応じたパラメータ直接変更
  - 要因分析と修正
  - 敵対的サンプルに対する特定ラベルに関する修正 [d]
  - 失敗データにおける影響個所の探索と修正 [e]
- 参考: 修正仕方全般の実証研究 [f]

[a] Generative adversarial nets, NIPS 2014
[b] MODE: automated neural network model debugging via state differential analysis and input selection, ESEC/FSE 2018
[c] Autoaugment: Learning augmentation policies from data, arXiv:1805.09501, 2019
[d] 敵対的サンプルに対するニューラルネットワークモデルの学習無し修正とその評価, JSSST大会 2019
[e] Search Based Repair of Deep Neural Networks, arXiv:1912.12463, 2019
[f] Repairing Deep Neural Networks: Fix Patterns and Challenges, ICSE 2020

敵対的サンプルに対する調整 [e]

- 要因分析: 特定のラベルに対する検証データと誤認識データとのニューロン出力の平均値の差分による局所化
- 修正: 局所化した個所の重みパラメータの特定修正率による更新 → 検証データによるデグレ有無の確認

検証データによる平均値



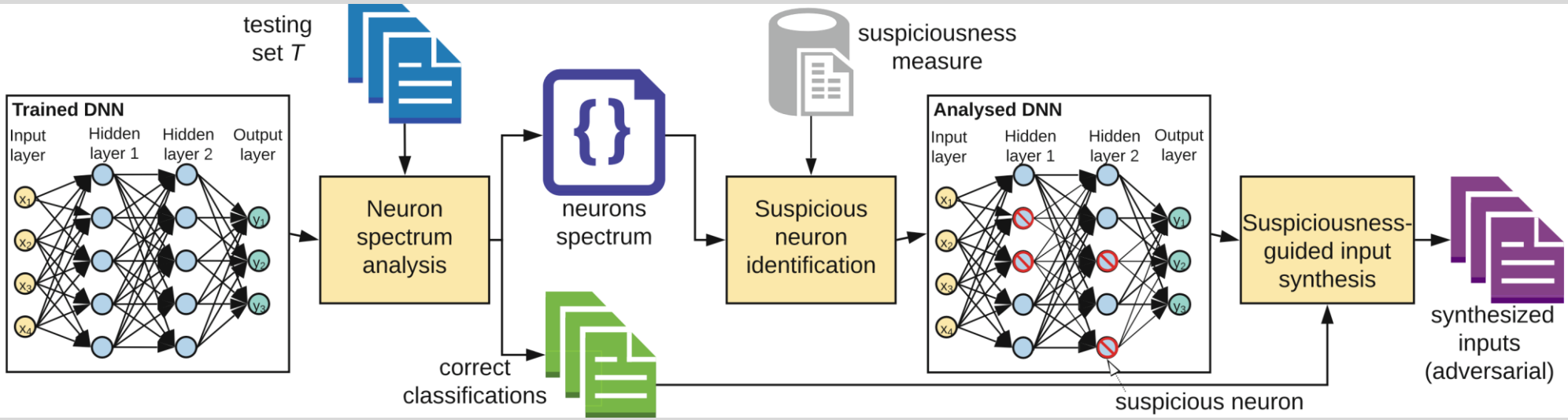
誤認識データによる平均値

最優先で修正すべきニューロン

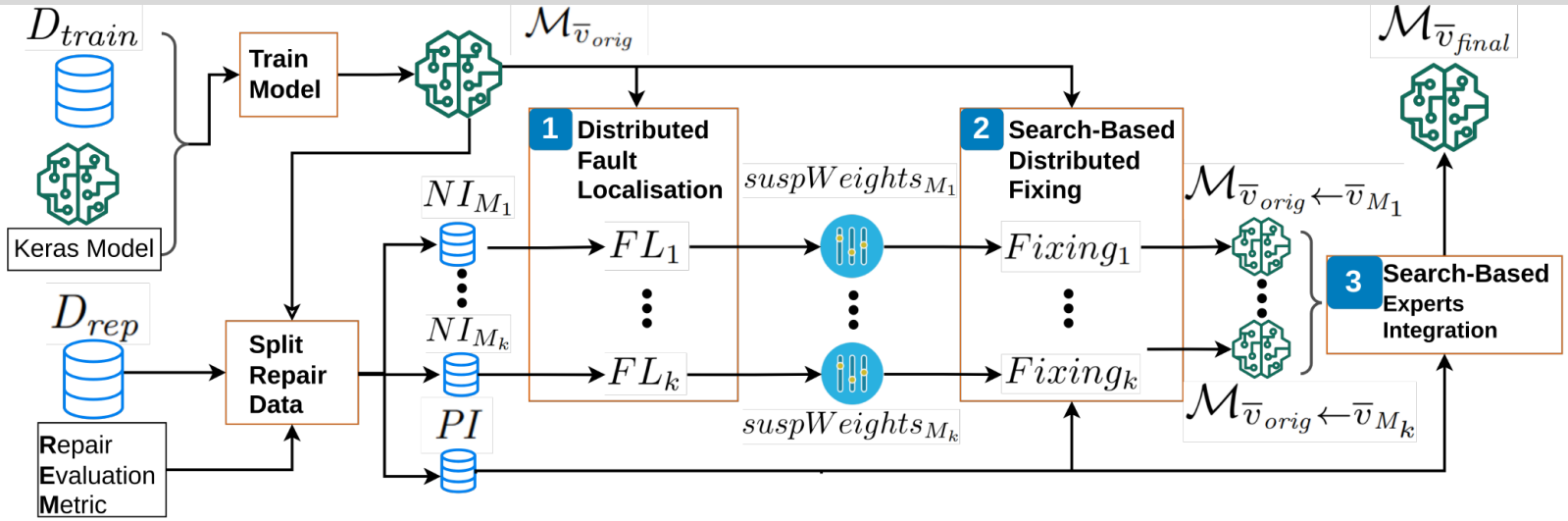
[d] 敵対的サンプルに対するニューラルネットワークモデルの学習無し修正とその評価, JSSST大会 2019

深層学習モデル 性能調整（リペア）の進展

- DeepFault: ニューロン疑惑値計算に誤分類(および正分類)時の活性化回数活用 [Eniser+19]



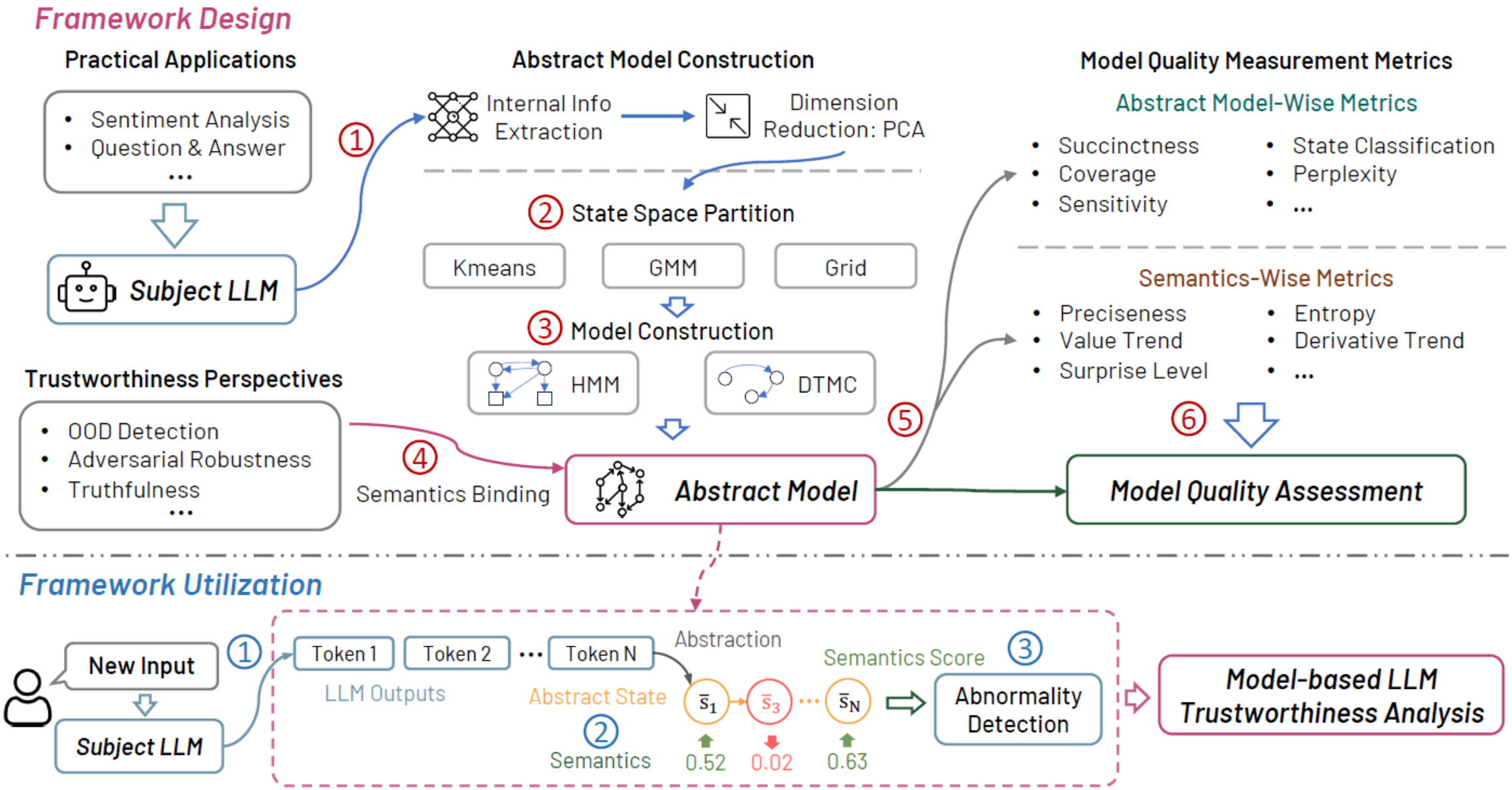
- Distributed Repair: 各誤分類に対して粒子群最適化に基づく重みのリペア、統合 [Calsi+23]



H.F. Eniser, et al., “DeepFault: Fault Localization for Deep Neural Networks,” FASE 2019
D. Li Calsi, et al. “Distributed Repair of Deep Neural Networks,” ICST 2023

https://github.com/jst-qaml/eAI-Repair-Toolkit/

LLMの性能解析に向けて [Song+24]



D. Song, et al, “LUNA: A Model-Based Universal Analysis Framework for Large Language,” IEEE Transactions on Software Engineering, 50(7), 2024

AIシステムの品質保証: 生成AIの評価

生成AIの評価観点

- 評価観点
  - 品質標準 ISO/IEC 25059:2023
  - 生成AIQMガイドライン
  - QA4AIガイドライン
- ベンチマーク
  - テキスト関連の一般ベンチマーク
  - 開発タスクの一般ベンチマーク: SWEBenchなど
- そのうえでタスクやシステム構成に応じた評価を設計

石川冬樹, 生成AIの評価, スマートエスイーセミナー「生成AIによるソフトウェア開発」, 2025
情報処理学会 監修, 鷲崎弘宜 編著, 家村康佑, 鵜林尚靖, 蝦名拓也, 近藤将成, 徳本晋, 中川尊雄, 増田航太, 石川冬樹, 竹之内啓太, 小川秀人, 川上真澄, “生成AIによるソフトウェア開発: 設計からテスト、マネジメントまでを全て変革するLLM活用の実践体系”, オーム社, 2025

タスク別評価: 検索・分類・検出

- 見落とし、誤検出の少なさの評価
  - 混同行列
- 上位KPI・リスクとの接続した評価
  - 例: 医療診断であれば見落としの重視
  - GQM+Strategiesといった枠組み活用も

		予測	
		陽性 Positive	陰性 Negative
正解	陽性	真陽性 True positive	**偽陰性 （見落とし）**
	陰性	**偽陽性 （誤検出）**	真陰性 True negative

石川冬樹, 生成AIの評価, スマートエスイーセミナー「生成AIによるソフトウェア開発」, 2025

情報処理学会 監修, 鷲崎弘宜 編著, 家村康佑, 鵜林尚靖, 蝦名拓也, 近藤将成, 徳本晋, 中川尊雄, 増田航太, 石川冬樹, 竹之内啓太, 小川秀人, 川上真澄, “生成AIによるソフトウェア開発: 設計からテスト、マネジメントまでを全て変革するLLM活用の実践体系”, オーム社, 2025

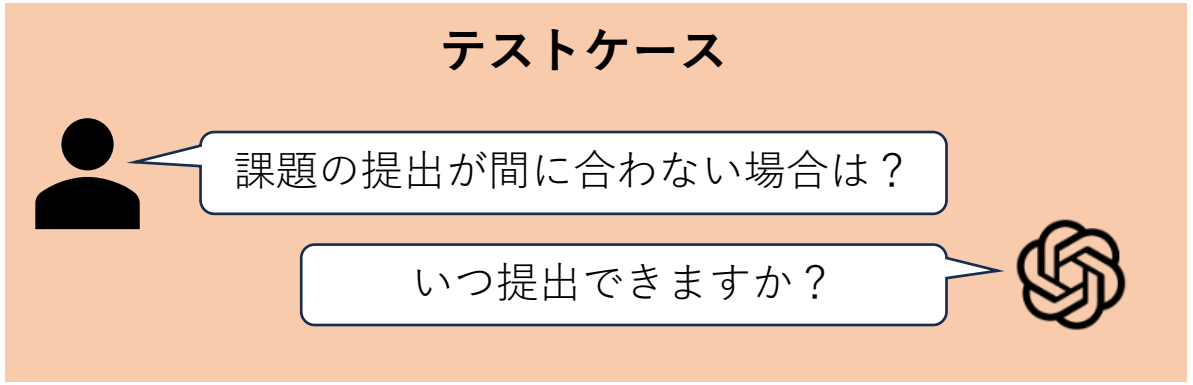
Victor Basili, Adam Trendowicz, Martin Kowalczyk, Jens Heidrich, Carolyn Seaman, Jurgen Munch, Dieter Rombach 著, 鷲崎弘宜, 小堀貴信, 新谷勝利, 松岡秀樹 監訳, 早稲田大学グローバルソフトウェアエンジニアリング研究所ゴール指向経営研究会 訳, “ゴール&ストラテジ入門: 残念なシステムの無くし方 (GQM+Strategies)”, オーム社, 2015

経験的ベンチマーク Heuristic Benchmark

- 問題: 結果の良し悪しの程度を意思決定者へ説明困難
- 解決: 過去との比較や経験則に基づく判断
- 考慮
  - （生成AIや機械学習に限らず）業界標準やプラクティスがあればそれに従う
  - ときとしてエキスパートの知見
  - モデルの改善の影響を利用価値（採用判断上の指標、例えば金銭的価値、良い検索結果など）に翻訳して説明できることが望ましい

シナリオ例	経験上のベンチマーク例	タスク例
時系列の将来値を予測する回帰問題	• 永続的か、線形トレンドか。 • 季節性の考慮 • 年間データの場合、前年の同じ日/週/四半期と比較	週間の販売量を予測
現在、人間の専門家が解決している分類問題	人間の専門家のパフォーマンス	網膜スキャンから目の病気を検出
・・・	・・・	・・・

タスク別評価: 非定型出力



正解無しの人による判定

正解無しのLLM判定:「提出可能日を尋ねている点は有用ですが・・・」

出力と正解の合致度のLLM判定（LLM-as-a-Judge）:「合致度は4です。受け付けられる可能性に言及していませんが・・・」

出力と正解との類似度測定: 48%（コサイン類似度、文字列編集距離など）

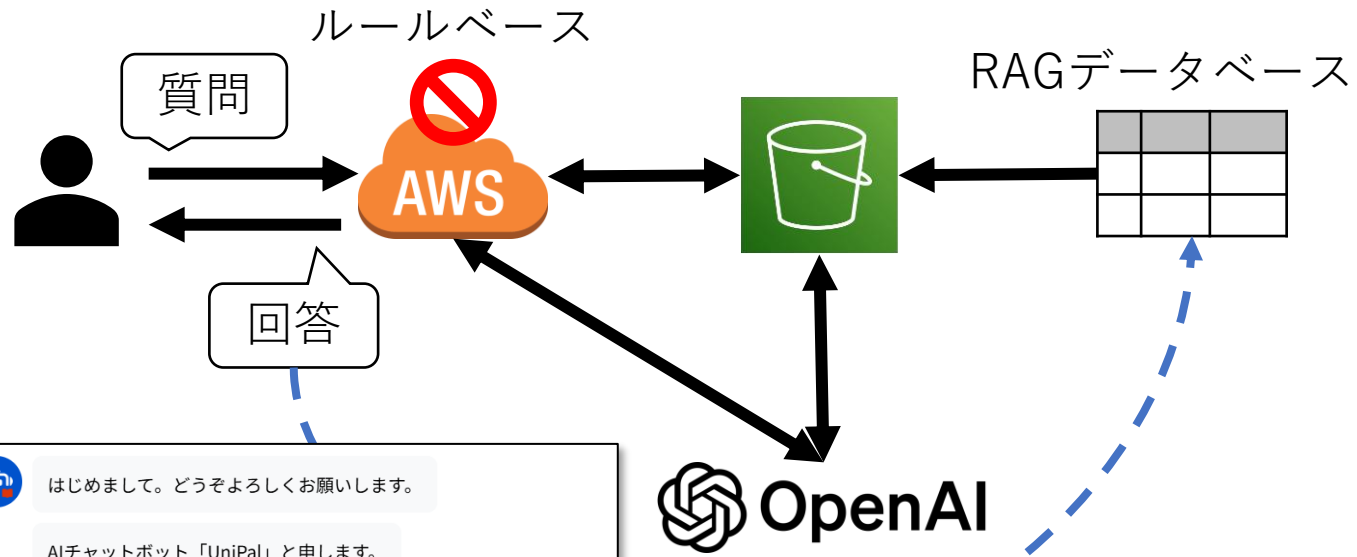
正解（模範解答）

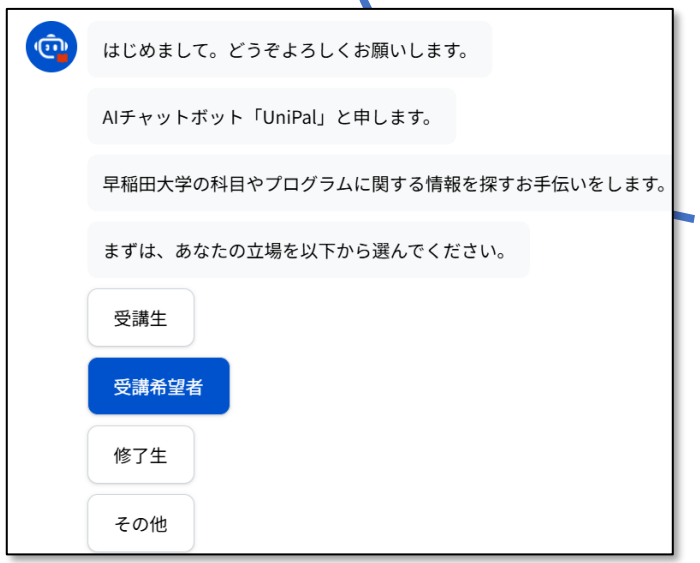
期限外提出を受け付けられる可能性があります。提出可能日を知らせてください。

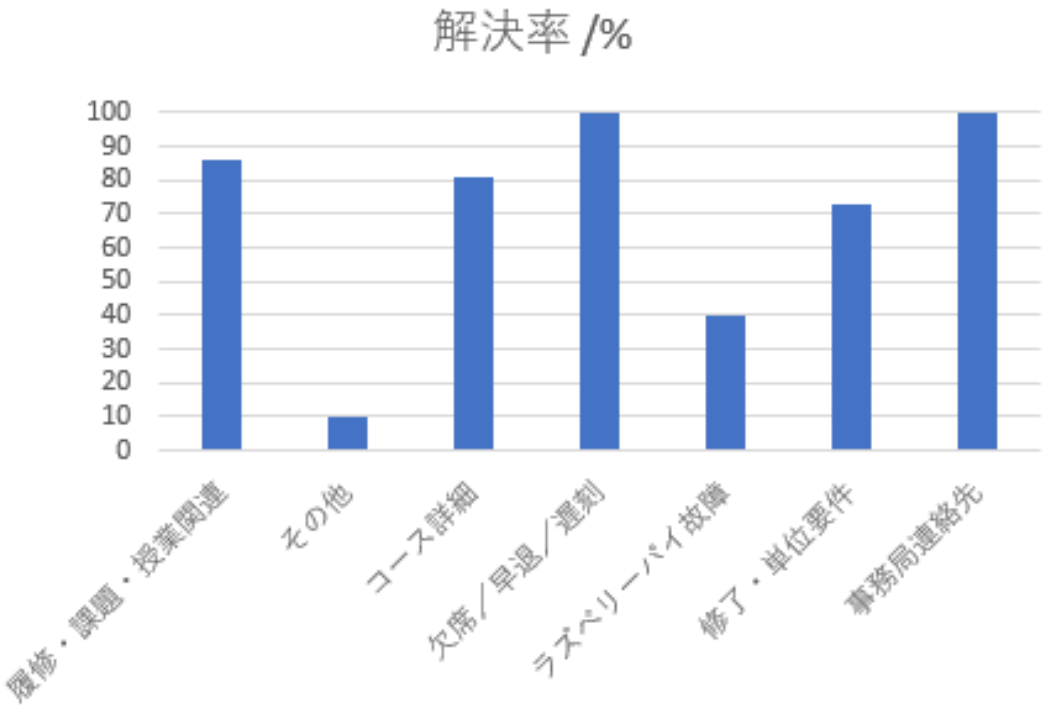
石川冬樹, 生成AIの評価, スマートエスイーセミナー「生成AIによるソフトウェア開発」, 2025
情報処理学会 監修, 鷲崎弘宜 編著, 家村康佑, 鵜林尚靖, 蝦名拓也, 近藤将成, 徳本晋, 中川尊雄, 増田航太, 石川冬樹, 竹之内啓太, 小川秀人, 川上真澄, “生成AIによるソフトウェア開発: 設計からテスト、マネジメントまでを全て変革するLLM活用の実践体系”, オーム社, 2025

評価例: 質問応答ボット事例 [COMPSAC'26]

- 受講者や検討者からの質問や依頼への自動回答・メール連絡
- 過去の対応履歴・FAQを追加情報源とするRAG
- ルールベースによる事務局誘導 ⇒ 非誘導の解決率60%強



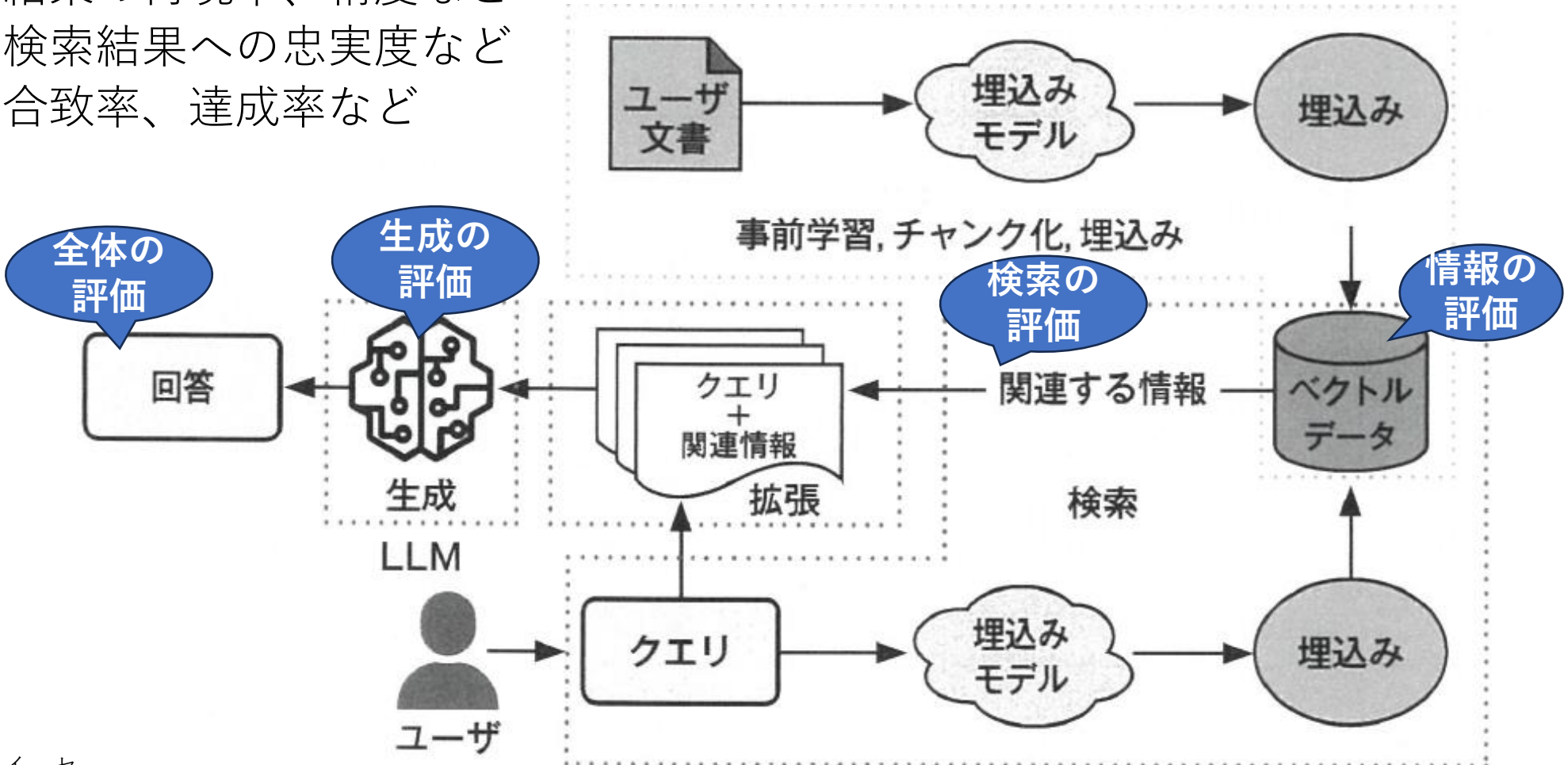




Hironori Washizaki, Shoichi Okazaki, Kazunori Sakamoto and Satoshi Okuda, "GenAI-Driven Transformation of ICT Continuing Education Program: A Case Study of "Smart SE"," 50th IEEE International Conference on Computers, Software, and Applications (COMPSAC 2026)

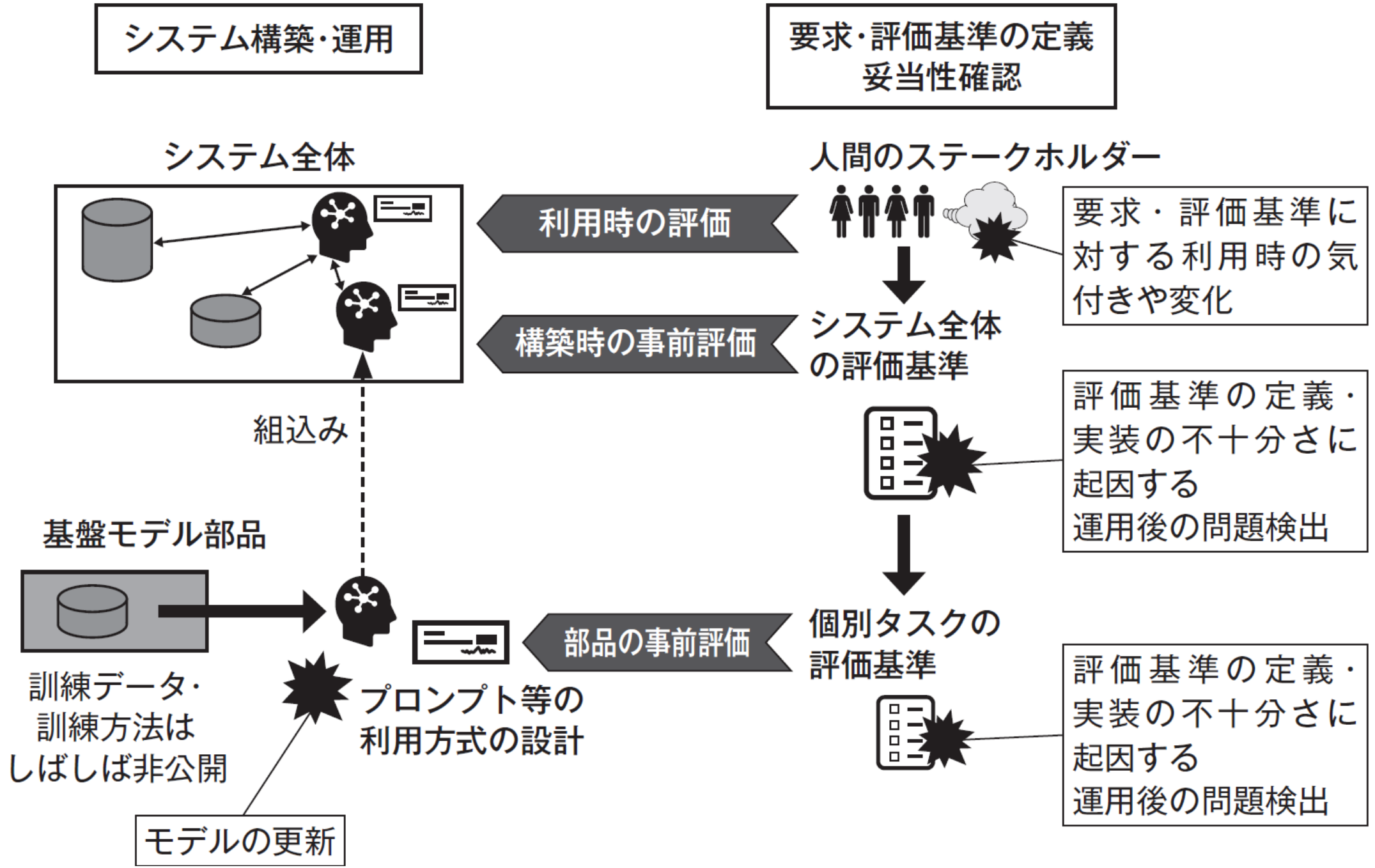
タスク別評価: RAGを用いた出力

- システム構造に応じたパート別の評価
  - データベース: 必要な情報の有無など
  - 検索部分: 検索結果の再現率、精度など
  - 回答生成部分: 検索結果への忠実度など
  - 全体: 正解との合致率、達成率など
- ツール活用
  - RAGAS, RAGCheckerなど



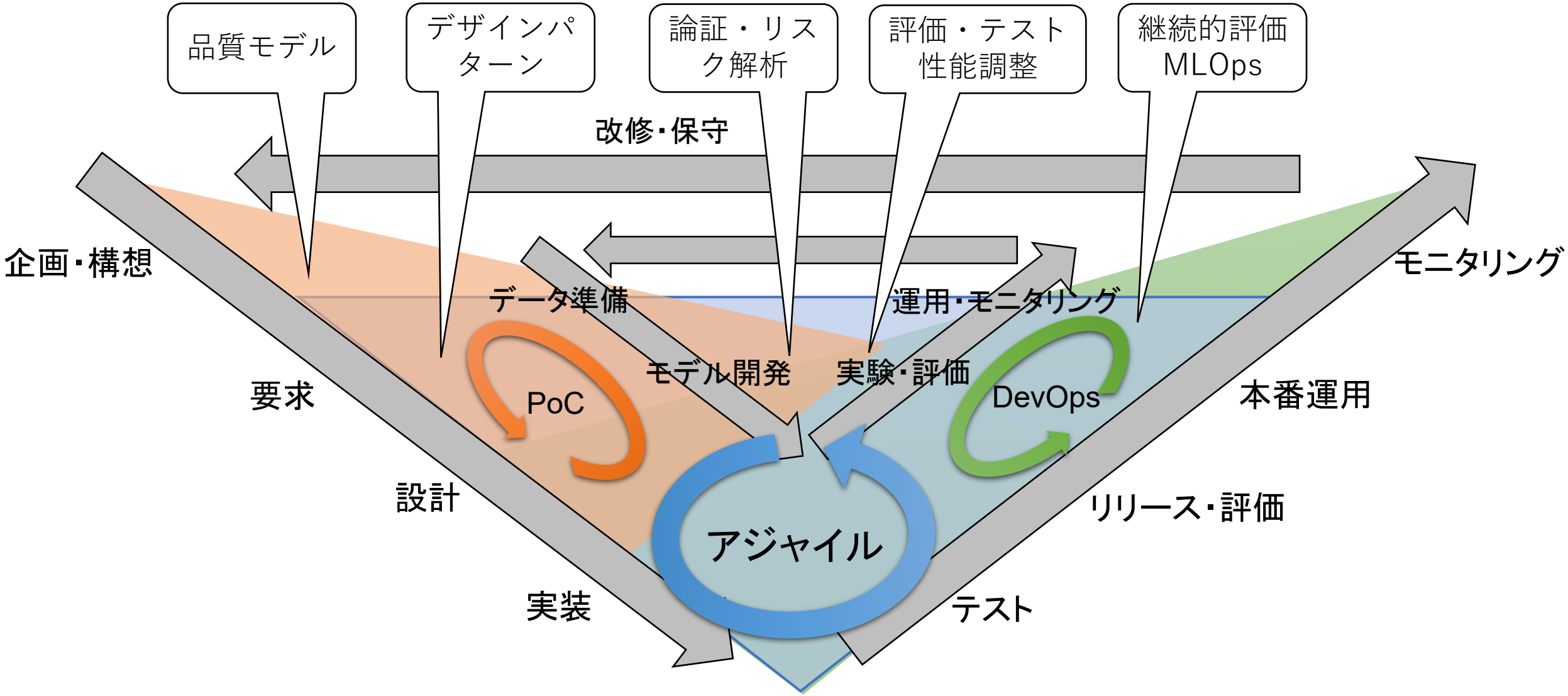
石川冬樹, 生成AIの評価, スマートエスイーセ

情報処理学会 監修, 鷲崎弘宜 編著, 家村康佑, 鵜林尚靖, 蝦名拓也, 近藤将成, 徳本晋, 中川尊雄, 増田航太, 石川冬樹, 竹之内啓太, 小川秀人, 川上真澄, “生成AIによるソフトウェア開発: 設計からテスト、マネジメントまでを全て変革するLLM活用の実践体系”, オーム社, 2025



情報処理学会 監修, 鷲崎 編著, 家村ほか著, ”生成AIによるソフトウェア開発”, オーム社, 2025年11月予定

再掲: AIシステム開発運用プロセスの例と対応技術

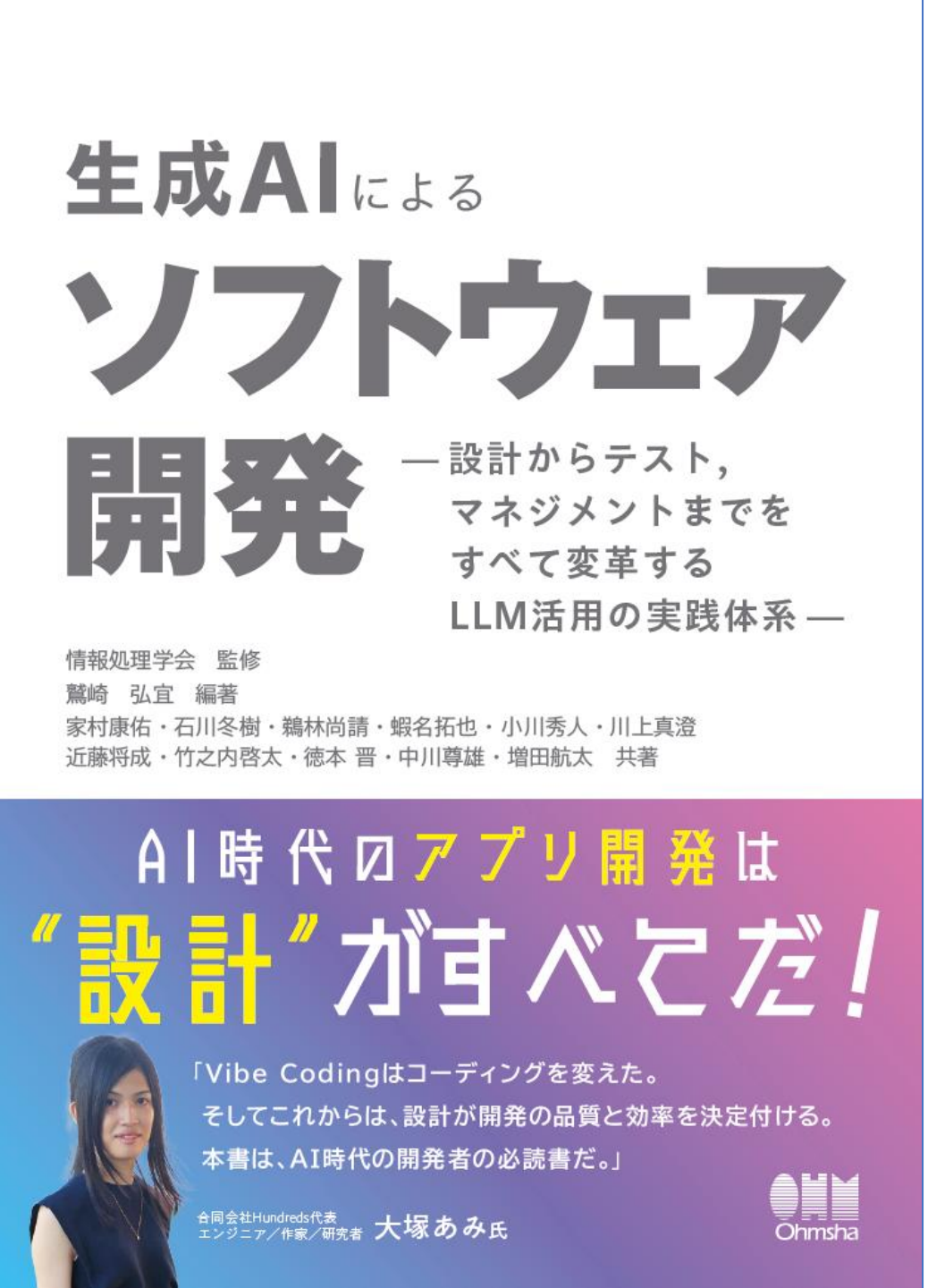


目次

- AIシステムの品質保証
  - 品質特性とマネジメント
  - 設計による品質
  - 論証とリスク解析
  - テスト&性能調整
  - 生成AIの評価

- **生成AIによる品質保証**
  - **生成AIによる品質考慮の要求・設計**
  - **Vibe Codingと品質**
  - **生成AIによるテスト**

生成AIによる品質保証



生成AIによるソフトウェア開発

設計からテスト、マネジメントまでを全て変革するLLM活用の実践体系

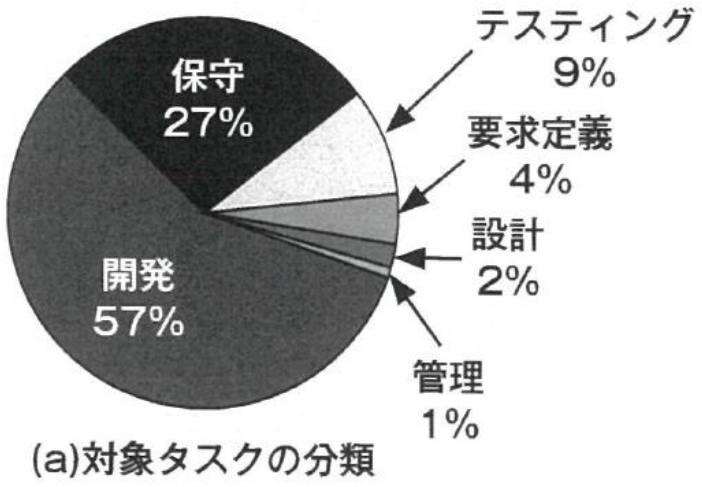
情報処理学会 監修（ソフトウェア工学研究会レビュー）
鷲崎 編著, 家村・石川・鵜林・蛯名・小川・川上・近藤・竹之内・徳本・中川・増田 著, オーム社, 2025年11月発刊
https://www.ohmsha.co.jp/book/9784274234156/
https://www.amazon.co.jp/dp/4274234150/

謝辞: 伊原氏、小林氏、斎藤氏、竹内氏、中才氏、名倉氏、藤原氏、細野氏、山下氏ほかレビューア各位

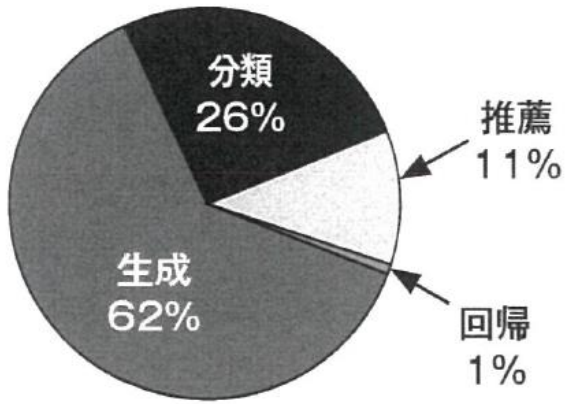
1. 生成AIの仕組み
2. 生成AIによるソフトウェアの要求
3. 生成AIによるソフトウェアの設計
4. 生成AIによるプログラムの実装
5. 生成AIによるソフトウェアのテスト
6. AIエージェントによるソフトウェア開発の自動化
7. 生成AIの評価
8. 生成AIを活用したプロセスとマネジメント
9. 生成AIによるソフトウェア産業の将来

ソフトウェアエンジニアリングにおけるLLM活用の広がり

- コード周辺中心 [Hou+24]
  - ソフトウェアエンジニアリングを抑えたライフサイクル全体へ
- LLMおよびデータの扱い留意
  - ハルシネーション
  - 計算コストとリソース効率
  - モデルの適応と知識の取り込み
  - プロプライエタリ、著作権



(a)対象タスクの分類

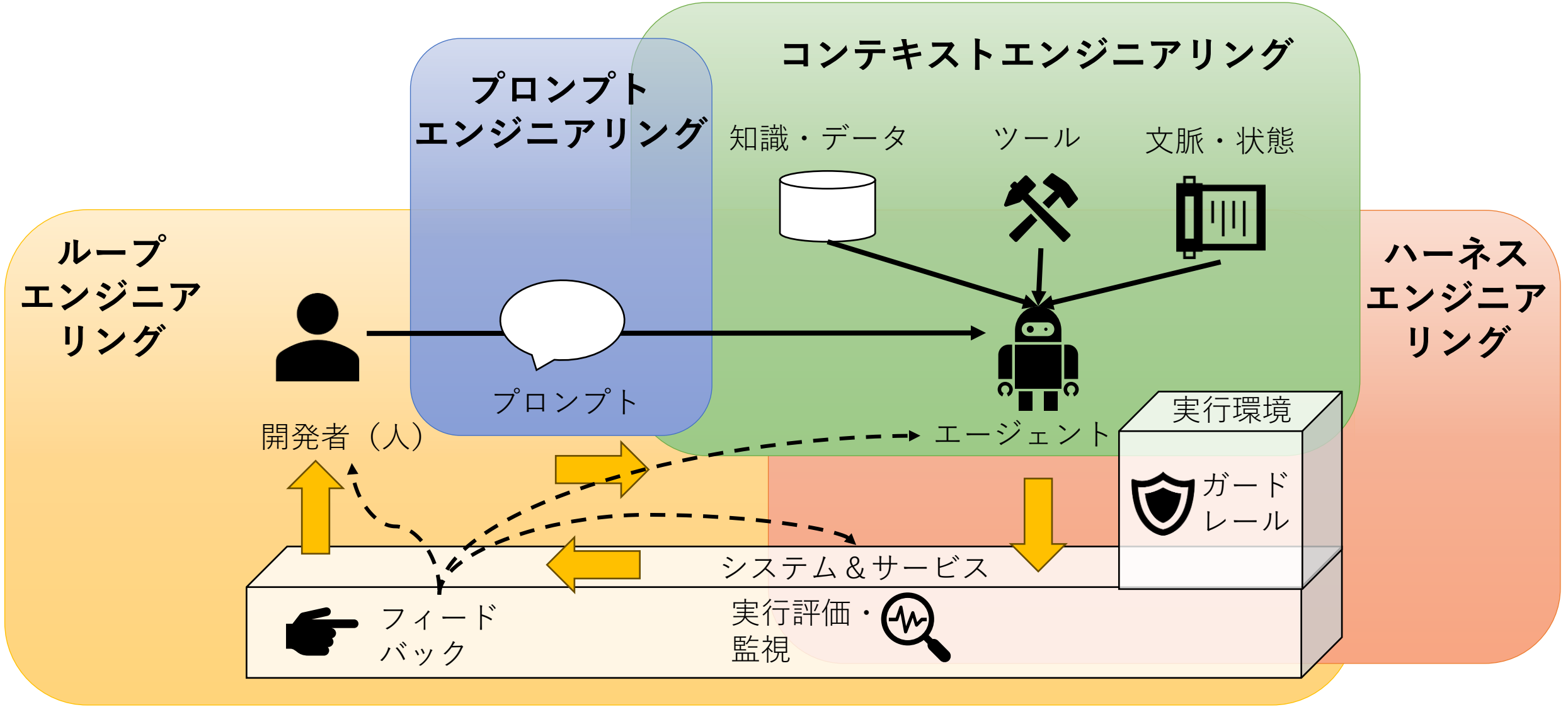


(b)LLMの使い方

- エンジニア＆開発者
  - 探索: 要求定義や設計といった不確実性の高い上流工程
  - 自動・効率化: コーディングやテストといった「正解」が明快な下流工程
  - 役割シフト: コードから糸中心へ、全体の設計と検証評価、コンテキスト管理、エージェント統制
- 組織的
  - 知識創造・蓄積循環の仕組みづくり
  - 組織能力を良くも悪くも増幅 ⇒ 基礎的な組織能力＆レディネスがカギ（参考 [Google+25]）

X. Hou, et al., “Large Language Models for Software Engineering: A Systematic Literature Review,” ACM TOSEM, 33(8), 2024
Google Cloud, “DORA State of AI-assisted Software Development report,” 2025

エンジニアリングの進展



エンジニアリングの進展（つづき）

プロンプトエンジニアリング

AIへの指示の最適化による単発の出力品質向上

- 指示の構造化
- 例示
- 制約や出力形式

コンテキストエンジニアリング

情報やツールの設計による文脈上の高品質実行

- RAG設計
- ツール連携
- 履歴・データ
- 手順 Task Skills

ハーネスエンジニアリング

エージェントの制御・観測・評価による安全な運用

- ガードレール（ポリシー・制約安全策など）
- 制御 Safety Skills
- 観測
- 自動テスト・評価
- 権限管理・統合

[OpenAI 26]ほか

ループエンジニアリング

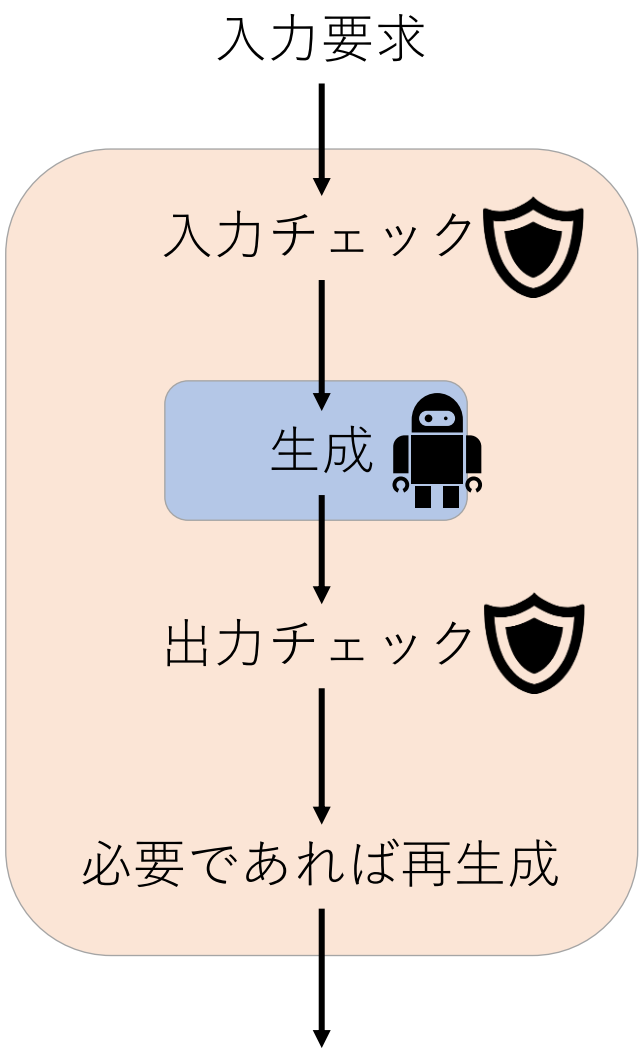
人・AI・システムのフィードバックループ設計による継続的改善

- 計画・実行・評価・改善ループ
- 自己プロンプト
- 介入ポイント設計
- ガバナンスと説明責任

[Osmani 26]ほか

OpenAI, ”ハーネスエンジニアリング：エージェントファーストの世界における Codex の活用”, 2026, https://openai.com/ja-JP/index/harness-engineering/
X. Qin, X., et al., “Harnessing Embodied Agents: Runtime Governance for Policy-Constrained Execution,” arXiv:2604.07833, 2026
Addy Osmani, "Loop Engineering,“ 2026, https://addyosmani.com/blog/loop-engineering/

ハーネスエンジニアリングの例: ガードレールを例に

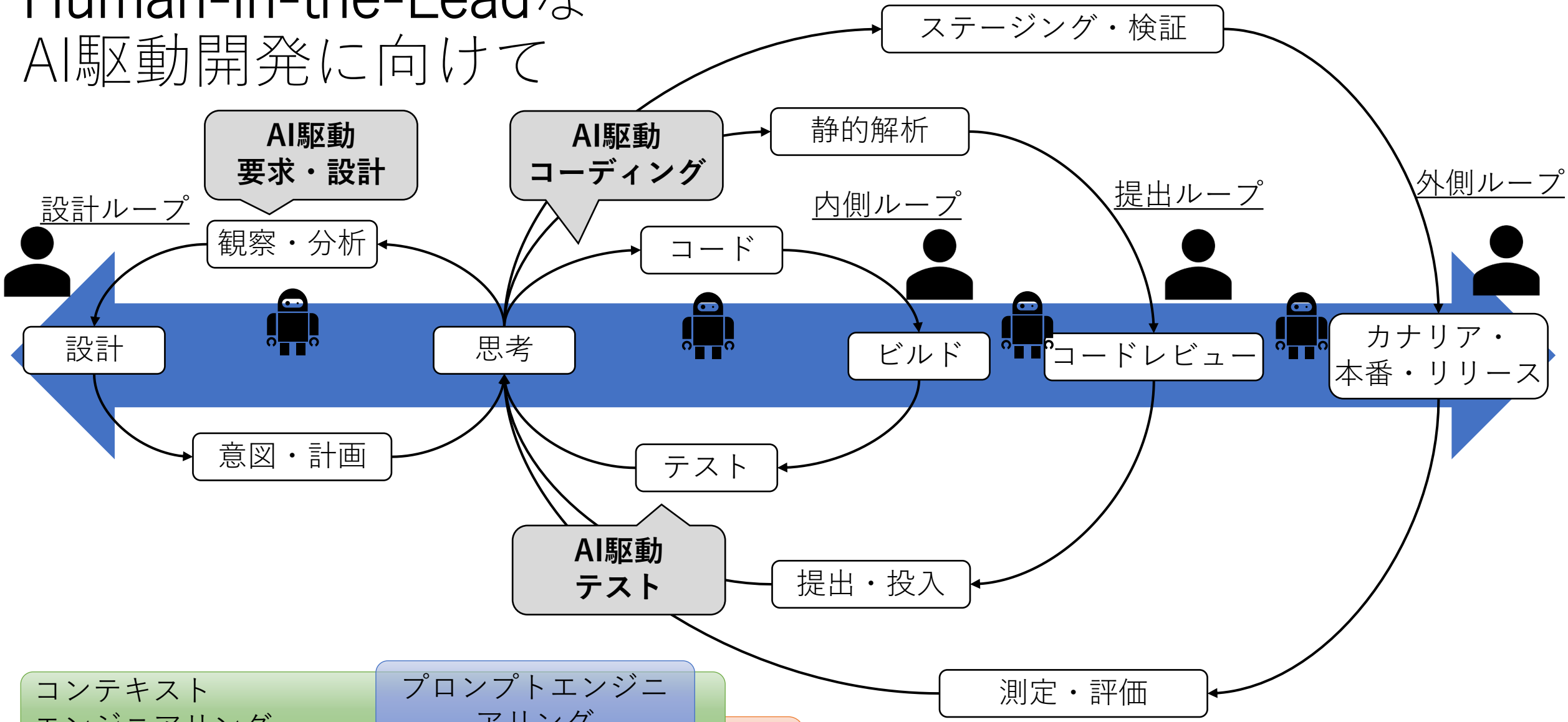



```
sql = claude.generate(user_request)
safe, reason = sql_injection_safeguard(sql)

if safe:
    database.execute(sql)
else:
    sql = claude.generate(
        ...
```

```
def sql_injection_safeguard(sql: str) -> tuple[bool, str]:
    dangerous_patterns = [
        " OR 1=1",
        " OR '1'='1”,
        ...
    ]
    normalized = sql.upper()
    for pattern in dangerous_patterns:
        if pattern.upper() in normalized:
            return False, f"Detected: {pattern}“
    ...
    return True, "OK"
```

Human-in-the-Leadな AI駆動開発に向けて

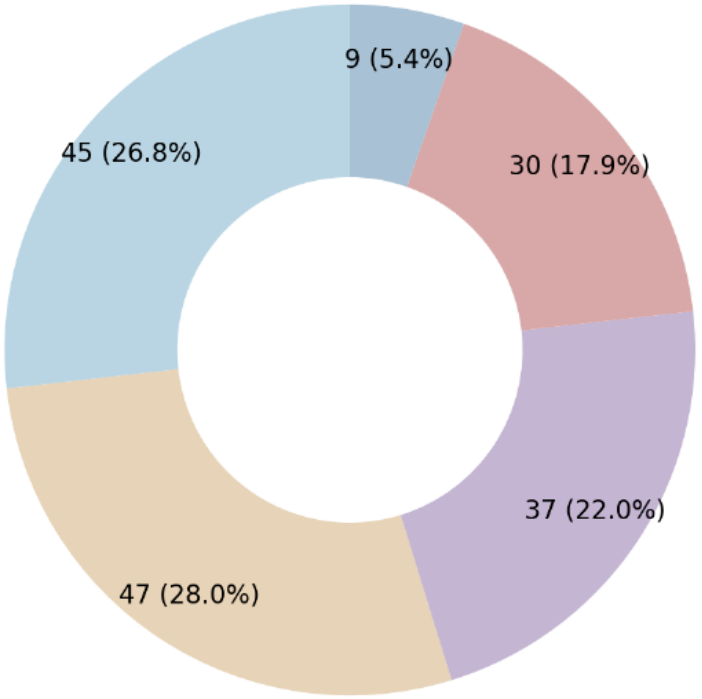


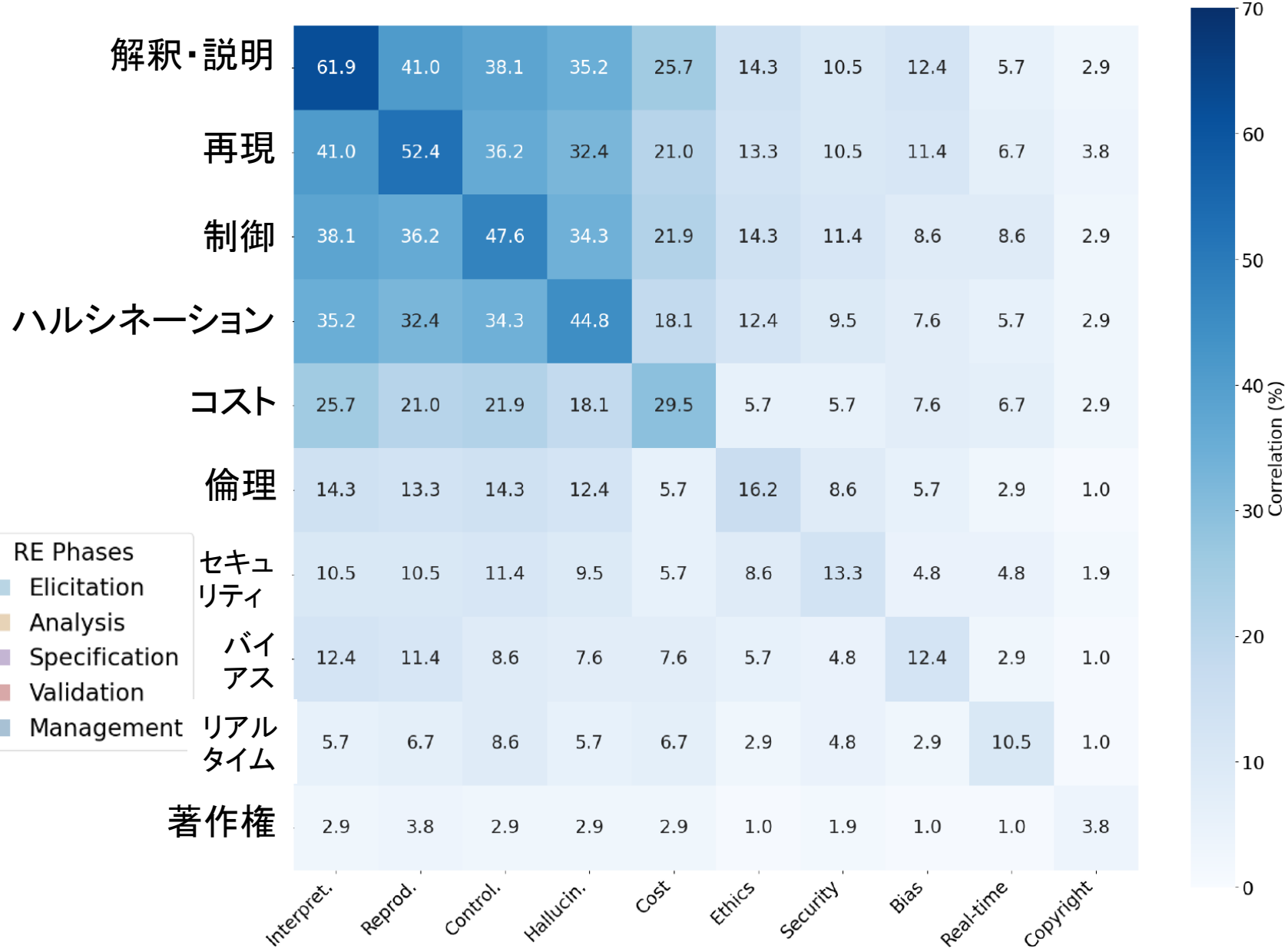
Addy Osmani, “Beyond Vibe Coding: From Coder to AI-Era Developer,” O’Reilly, 2025, https://beyond.addy.ie/

生成AIによる品質考慮の要求・設計

要求工学におけるLLM活用と課題 [Cheng+26]

- 要求抽出から分析、仕様化、検証まで幅広く。マネジメントは少数。
- 解釈・説明、再現性、制御性、ハルシネーションの扱いが最重要課題



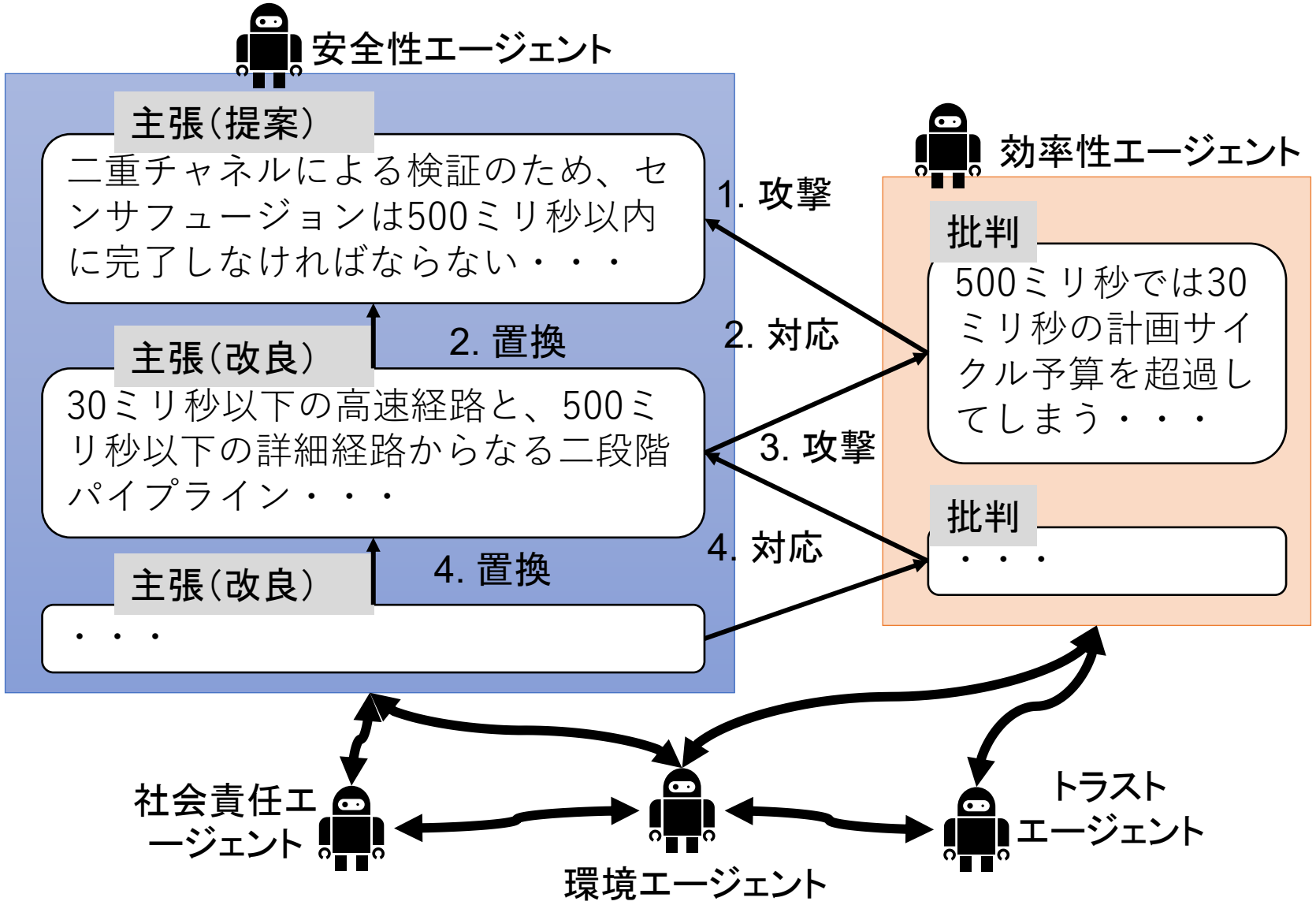


Haowei Cheng, Jati H. Husen, Yijun Lu, Teeradaj Racharak, Nobukazu Yoshioka, Naoyasu Ubayashi, Hironori Washizaki, “Generative AI for Requirements Engineering: A Systematic Literature Review,” Software: Practice and Experience, 56(2), 2026

62

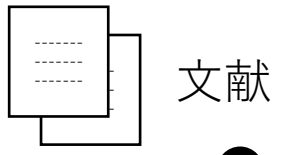
要求分析の例: マルチエージェントによる交渉 [RE'+26]

- 品質要求間の対立の扱い
- 提案、批判、統合を通じた弁証法的交渉



H. Cheng, M. Kim, F. Khomh, T. Racharak, N. Yoshioka, N. Ubayashi, H. Washizaki, "QUARE: Quality-Aware Requirements Engineering through Multi-Agent Dialectic Negotiation," 34th IEEE International Requirements Engineering Conference (RE 2026) **(Open Science Award)**

生成AIによるアーキテクチャ設計支援 [Mori+26]



文献

満たす

ECU集約

重量削減

満たす

ECU数削減

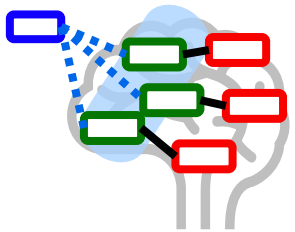
満たさない

単一障害点回避

複数文献知識を統合

比較観点	ECU集約	…
重量	＋	
ECU数	＋	
単一障害点	－	

トレードオフ表



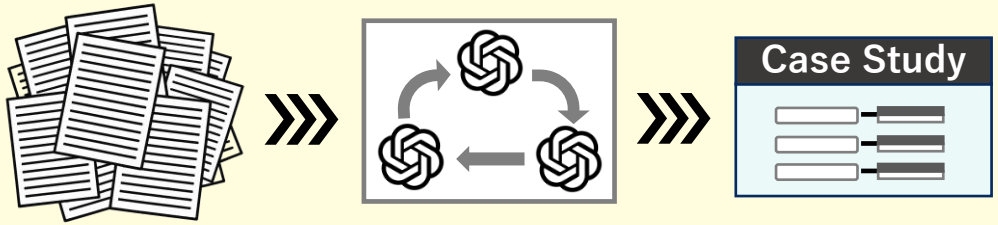
HippoRAG参考: 人間の海馬の記憶想起を模して、知識をグラフ構造として保持し関連概念を連想的に探索 [Gutinerrez+24]

Shunsuke Mori, Kohei Nishio, Taku Inagawa, Hironori Washizaki, Naoyasu Ubayashi, “Toward Evidence-Based Architectural Decision Making over Fragmented Knowledge,” 23rd IEEE International Conference on Software Architecture (ICSA), 2026.

B. Gutierrez, et al. “Hipporag: Neurobiologically inspired long-term memory for large language models.” *Advances in Neural Information Processing Systems*, 2024

LLMエージェントによるセキュリティ知識整理と論証生成
[CAI'25][SQAA4AI'26]

脅威知識の構造化・体系化



脅威レポート
（非構造化テキスト）

LLMベース
パイプライン

攻撃シナリオ
（Case Study）

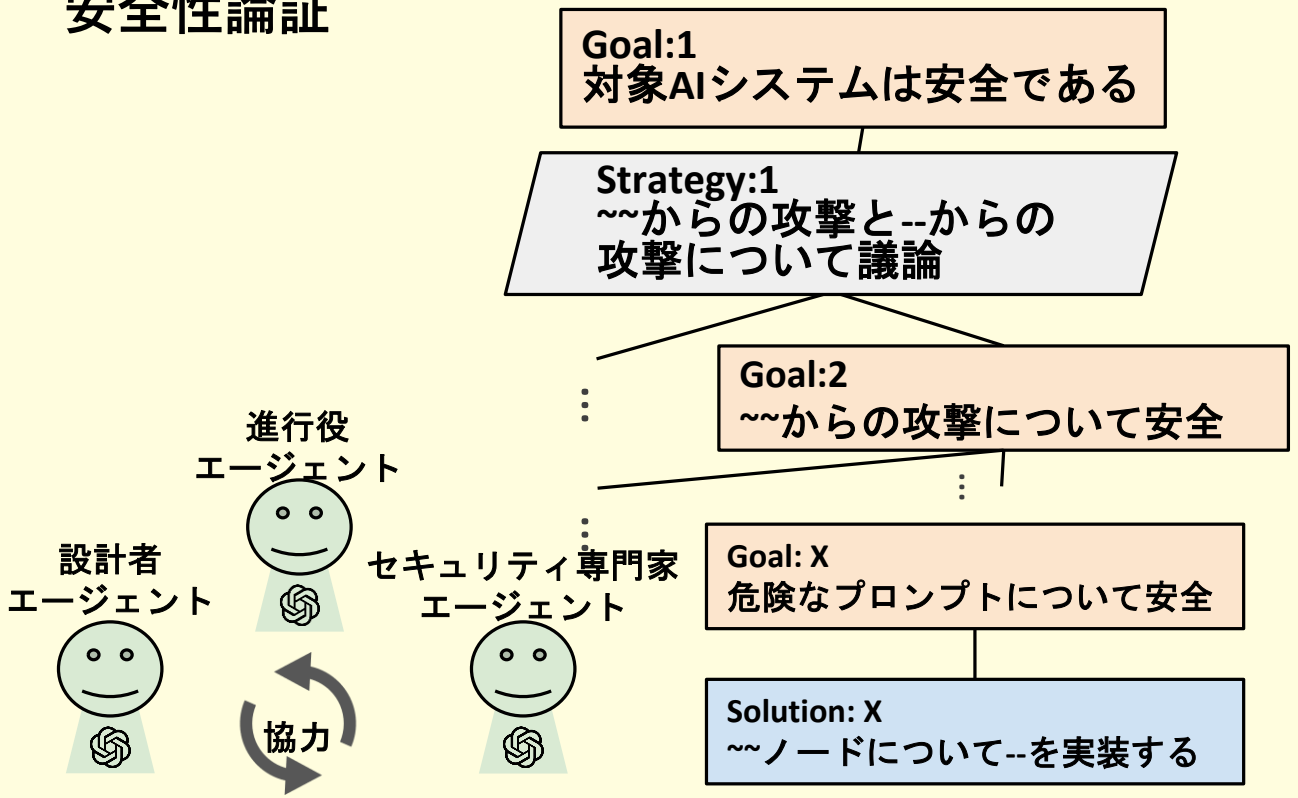
Case Study

Incident Date: ~~ | Reporter: ~~
Actor: ~~ | Target: ~~
Summary: ～～
Procedure:

01 攻撃者は～を埋め込んだ。	Tactic AA / Technique XX
02 攻撃者は～を生成した。	Tactic BB / Technique YY
03 攻撃者は～の検出を回避した。	Tactic CC / Technique ZZ

⋮

安全性論証



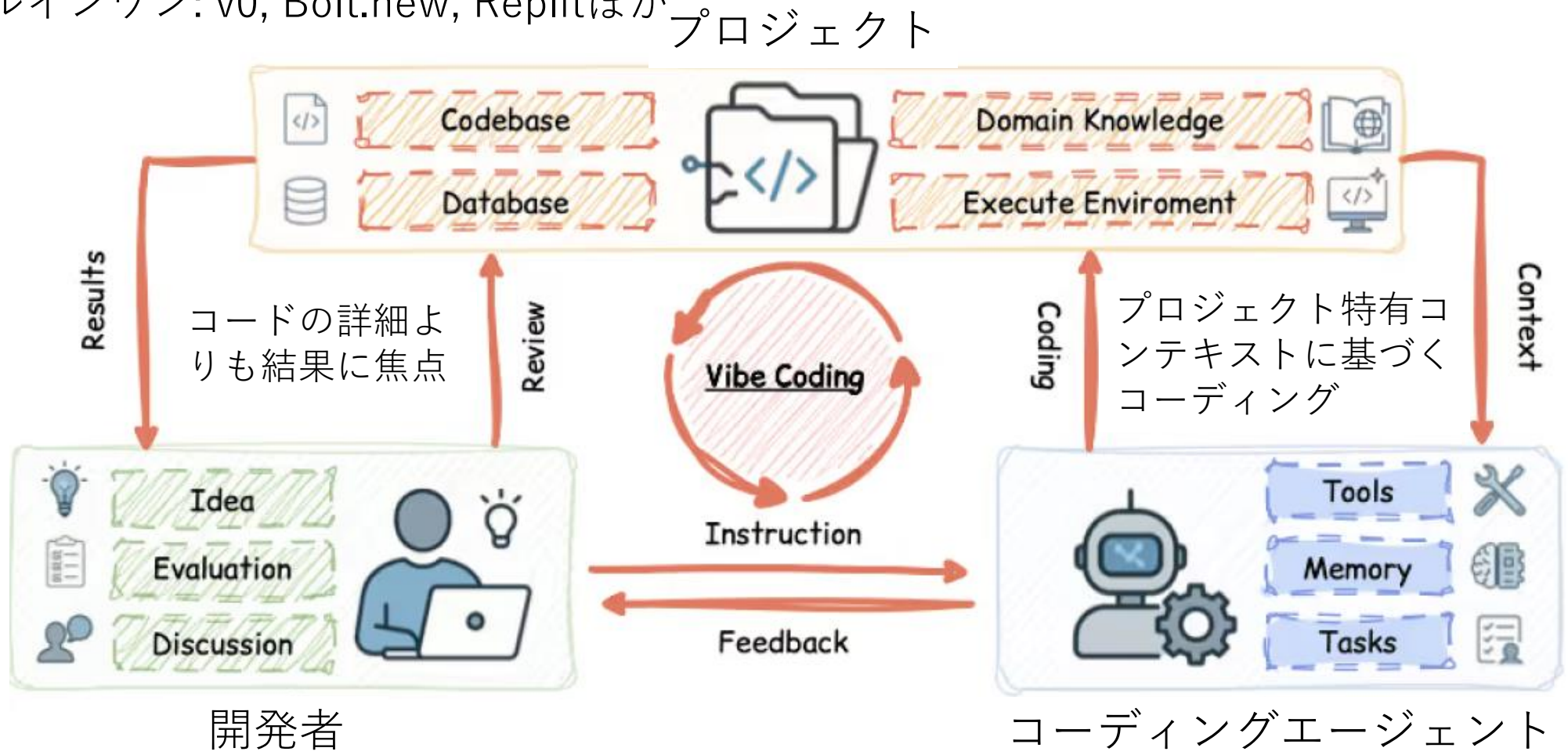
Yuya Fujiwara, Hironori Washizaki, Naoyasu Ubayashi, Rikuho Miyata and Takuma Tsuchida, "LLM-based Automated Mitigation and Assurance Case Generation Against Threats to AI Systems," The IEEE Conference on Artificial Intelligence (IEEE CAI 2025)

Takuma Tsuchida, Yuya Fujiwara, Hironori Washizaki and Naoyasu Ubayashi, "From Threat Reports to Security Knowledge: Building an LLM-based Pipeline for AI Systems," 3rd Workshop on Software Quality Assurance for AI (SQA4AI 2026), co-located with SANER 2026

Vibe Codingと品質

Vibe Codingとツール

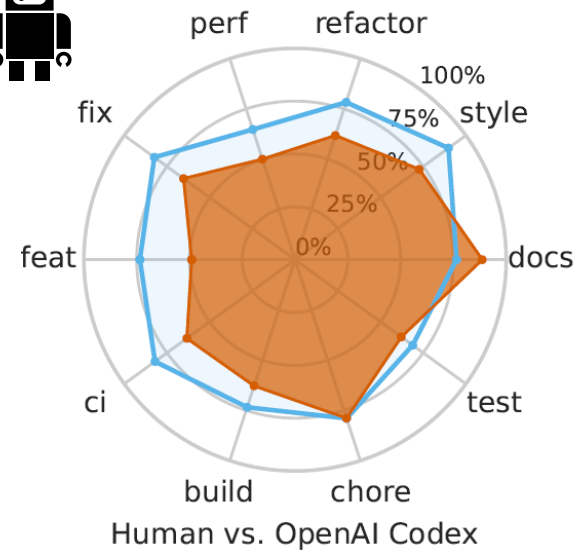
- LLMを用いて自然言語による指示・対話・協調によりプログラム作成
- 統合開発環境・プラグイン: GitHub Copilot, Cursor, Windsurf, Tabnine, CodeWhisperererほか
- コマンドラインツール: OpenAI Codex, Anthropic Claude Code, Gemini CLI, Qwen Code, Aider, Clineほか
- オールインワン: v0, Bolt.new, Replitほか

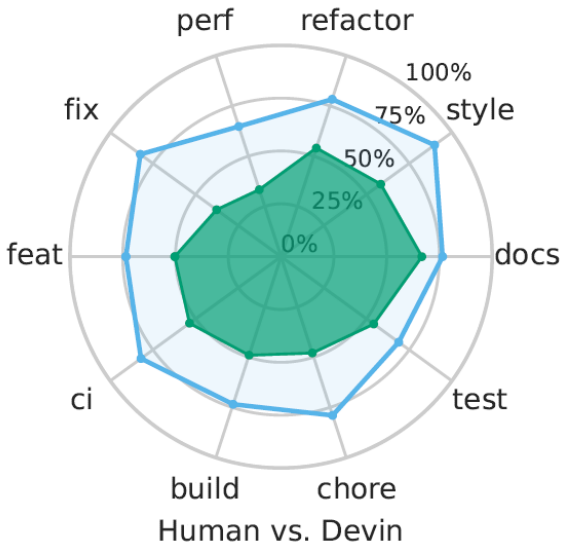


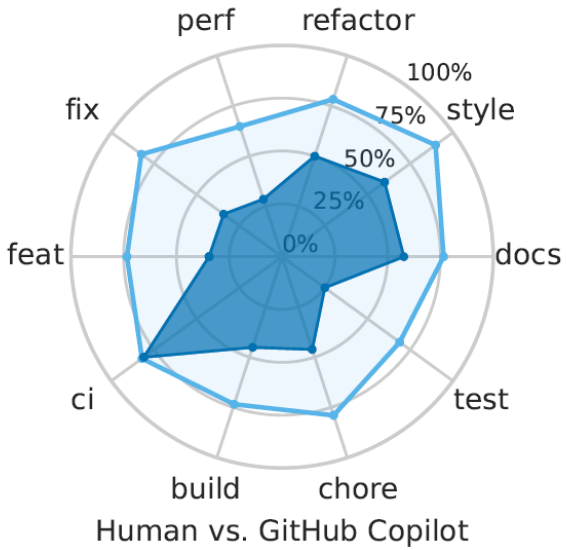
Yuyao Ge, et al., A Survey of Vibe Coding with Large Language Models, arXiv:2510.12399, 2025

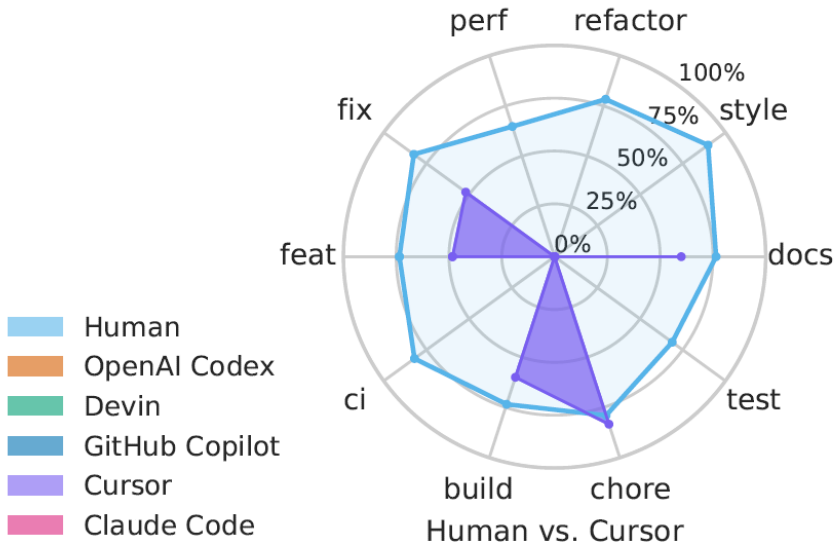
コーディングエージェントと人の協調に向けて[Li+25][Peralta+26]

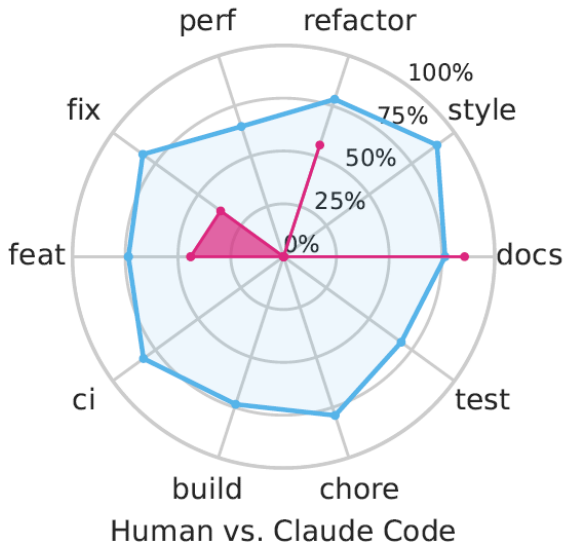
- エージェントは速度面で優れるものの、プルリクエスト（変更提案）の承認率は低い
- 不承認の主な原因はエージェントの技術的失敗（例: テストの誤り）ではない
- 人の介入やフィードバックループにより承認率を高められる可能性











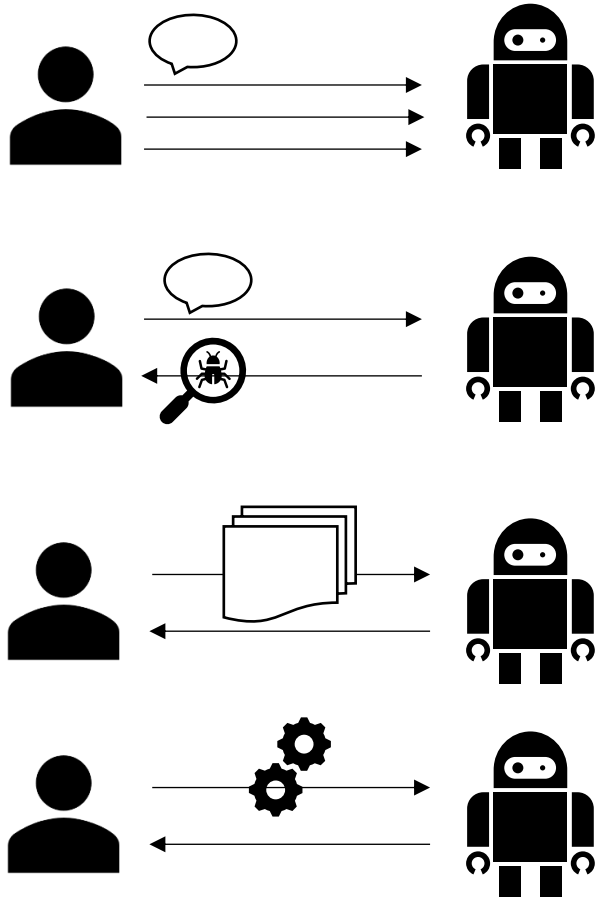
build	ビルド
ci	CI
chore	雑事
docs	ドキュメント
feat	新機能
fix	バグフィックス
perf	パフォーマンス
refactor	リファクタリング
style	コードスタイル修正
test	テスト

https://gist.github.com/minop1205/5fc4f6ef0ec89fb1738833ba25ae00a0

Hao Li, et al., “The Rise of AI Teammates in Software Engineering (SE) 3.0: How Autonomous Coding Agents Are Reshaping Software Engineering,” arXiv:2507.15003, 2025.
Sien Reeve O. Peralta, Fumika Hoshi, Hironori Washizaki, Naoyasu Ubayashi, Inase Kondo, Yoshiki Higo, Hiroki Mukai, Norihiro Yoshida, Kazuki Kusama, Hidetake Tanaka, Youmei Fan, “Why Are Agentic Pull Requests Merged or Rejected? An Empirical Study,” MSR 2026 **(Distinguished Mining Challenge Paper Award)**

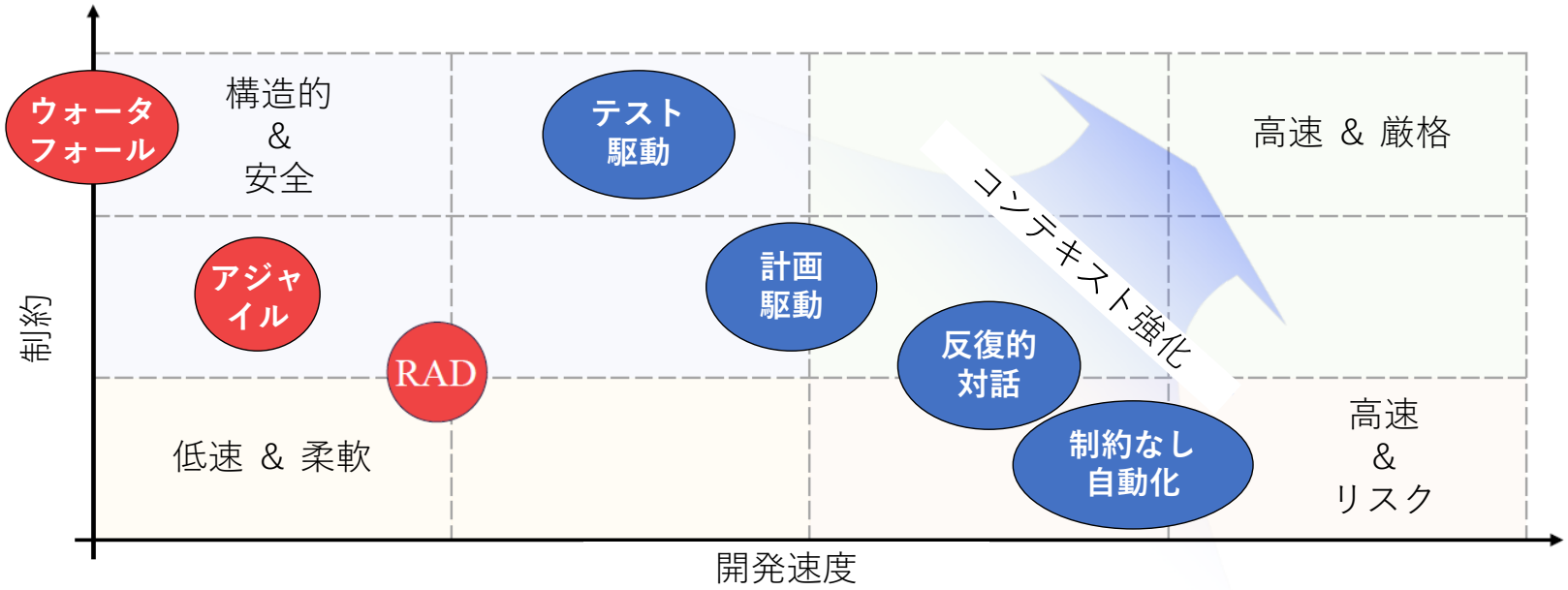
AIとの協調プロセスモデル

- 制約なし自動化（Unconstrained Automation Model: UAM）
  - AIを信頼、いわゆる「Vibe Coding」
  - スピード重視のプロトタイピング向き、リスク大
- 反復的対話協調（Iterative Conversational Collaboration Model: ICCM）
  - AI出力を都度レビュー、AIとのペアプログラミング
  - スピードと品質のバランス
- 計画駆動（Planning-Driven Model: PDM）
  - 人による計画と設計仕様に基づき、AIによる段階的な実装
  - いわゆる「仕様駆動開発」と概ね共通（ただし関心が異なる）
- テスト駆動（Test-Driven Model: TDM）
  - 正確な仕様としてのテストを通るようにAIが実装
  - テスト駆動開発のVibe Codingへの適用、品質重視
- コンテキスト強化（Context-Enhanced Model: CEM）
  - RAG、コードベースベクトルインデックス作成、ドキュメント読込み、ルール制約など
  - 他のモデルとの組み合わせ



Yuyao Ge, et al., “A Survey of Vibe Coding with Large Language Models,” arXiv:2510.12399, 2025.

協調プロセスモデルの比較



重視するエンジニアリング

プロセスモデル	事前投資	制御性	構造的制約	開発速度	コード品質	保守性	セキュリティ	技術的負債	対応概念	プロンプト	コンテキスト	ハーネス	ループ
制約なし自動化	無し	無し	無し	**最速**	低	低	低	**高**	RAD	**大**	中	小	小
反復的対話	小	**高**	中	速	高	高	中	低	ペアプロ	**大**	**大**	中	**大**
計画駆動	**大**	**高**	**厳格**	中	高	**厳格**	高	低	ウォータフォール	中	**大**	**大**	中
テスト駆動	**大**	中	**厳格**	中	**厳格**	高	**厳格**	無し	TDD	中	中	**大**	中
コンテキスト強化	中			+1	+1	+1	+1	-1		中	厳格	**大**	**大**

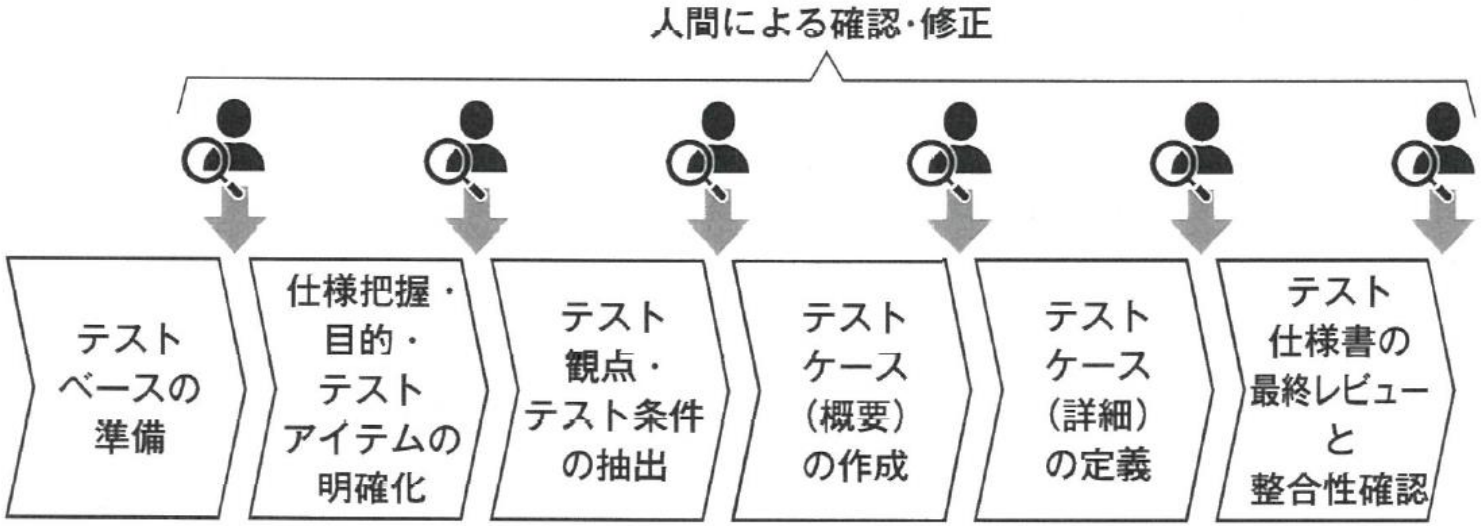
Yuyao Ge, et al., “A Survey of Vibe Coding with Large Language Models,” arXiv:2510.12399, 2025（「重視」は講演者追加）

生成AIによるテスト

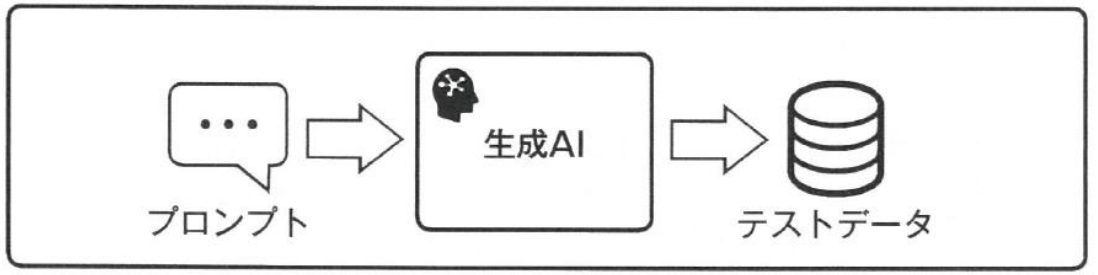
テストにおける活用

- テストケースの生成
  - テスト仕様書
  - テストコード
- テストデータの生成

テスト仕様書生成の流れ



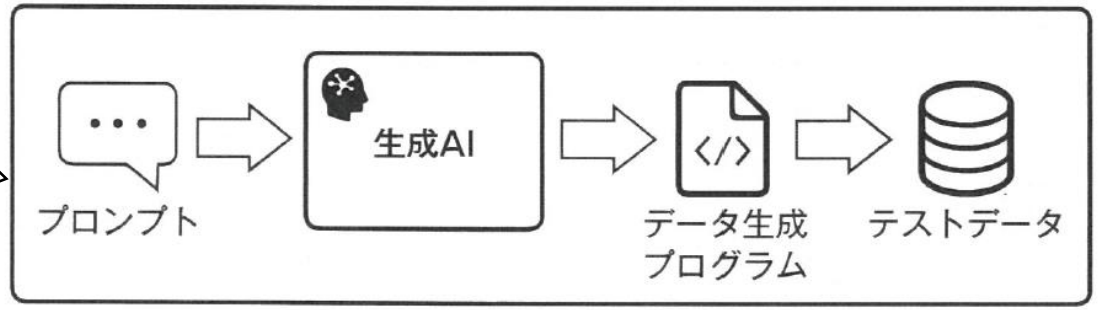
多様性に優れる



(a)パターン1：テストデータを直接生成

テストデータの生成

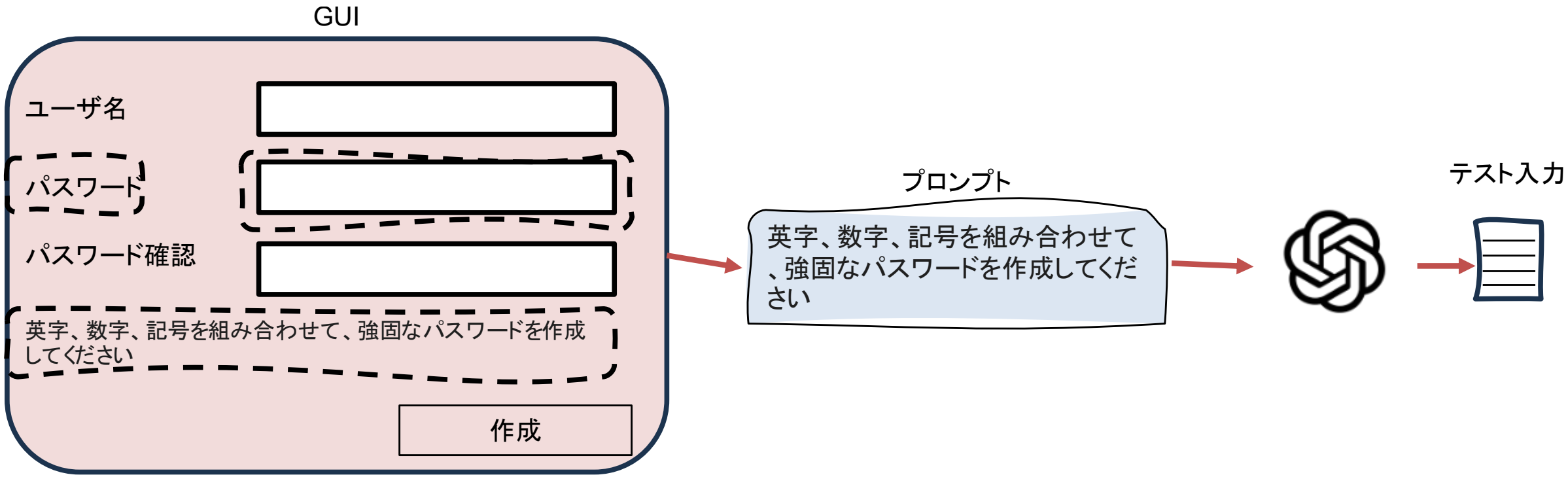
コスト、実行時間に優れる



(b)パターン2：データ生成プログラムを生成

情報処理学会 監修, 鷲崎弘宜 編著, 家村康佑, 鵜林尚靖, 蝦名拓也, 近藤将成, 徳本晋, 中川尊雄, 増田航太, 石川冬樹, 竹之内啓太, 小川秀人, 川上真澄, “生成AIによるソフトウェア開発: 設計からテスト、マネジメントまでを全て変革するLLM活用の実践体系”, オーム社, 2025

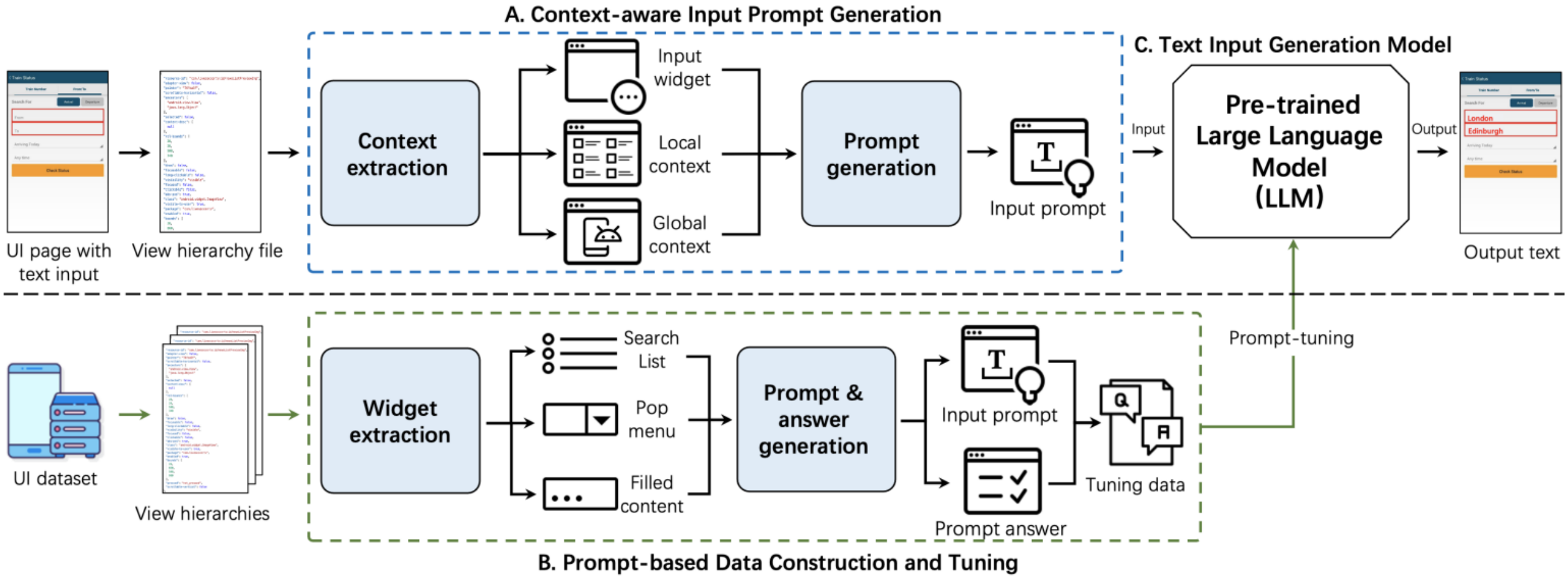
テスト入力生成におけるコンテキストエンジニアリング



Hussein Al Osman, Leveraging LLMs in Software Testing: Current State-of-the-Art and Future Directions, FATTW 2025

Z. Liu et al., "Fill in the blank: Context-aware automated text input generation for mobile GUI testing," in Proceedings of the 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE), IEEE, 2023.

テスト入力生成におけるコンテキストエンジニアリング（つづき）



Z. Liu et al., "Fill in the blank: Context-aware automated text input generation for mobile GUI testing," in Proceedings of the 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE), IEEE, 2023.

テストへの活用におけるプロンプトエンジニアリング

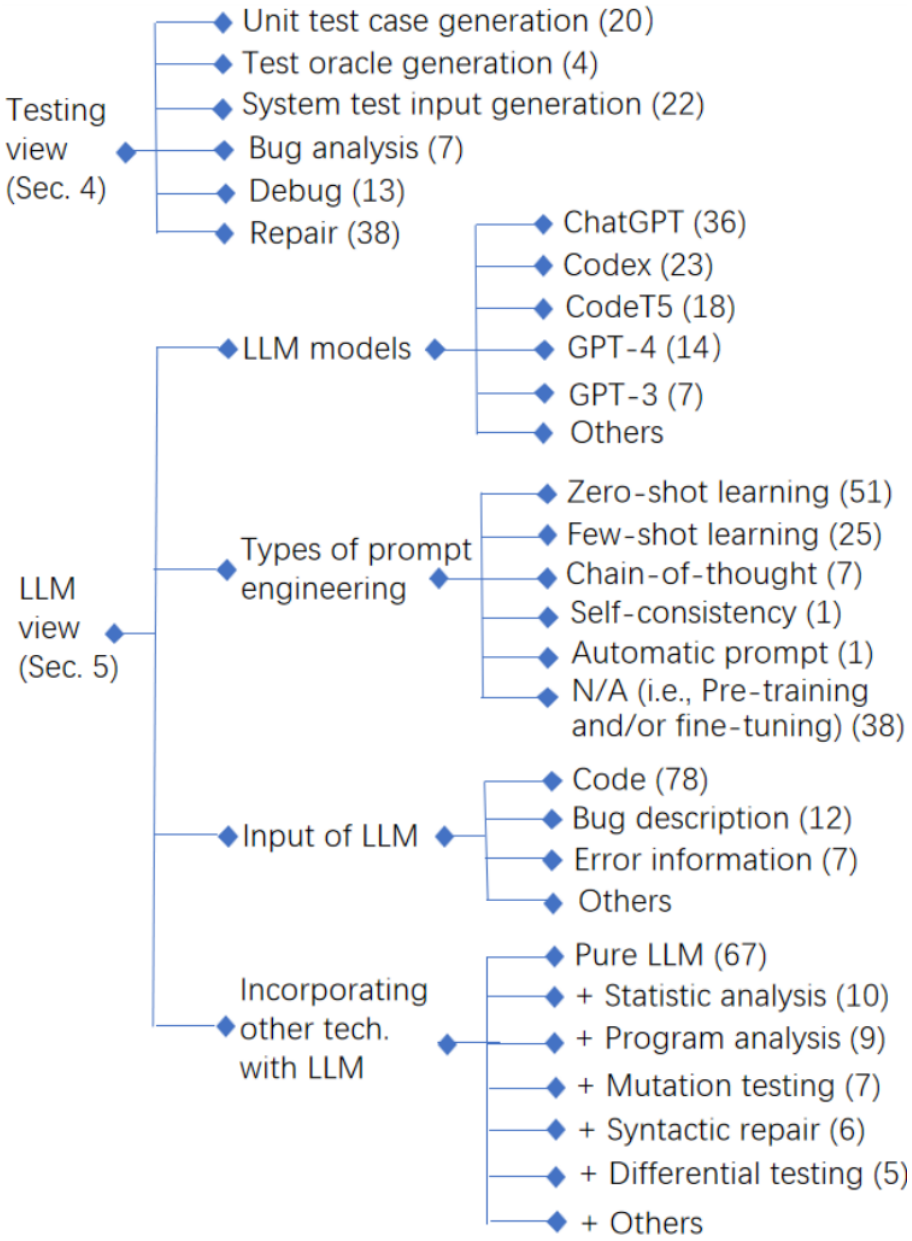
- 追跡性: 扱う要求などにID付与し追跡性確保
- 出力形式: 人とAIの両者が確認しやすいフォーマット指定
- コンテキスト: 背景知識や意図などを含める
- セルフリフレクション: 要求の網羅性や整合性のAI自己チェック

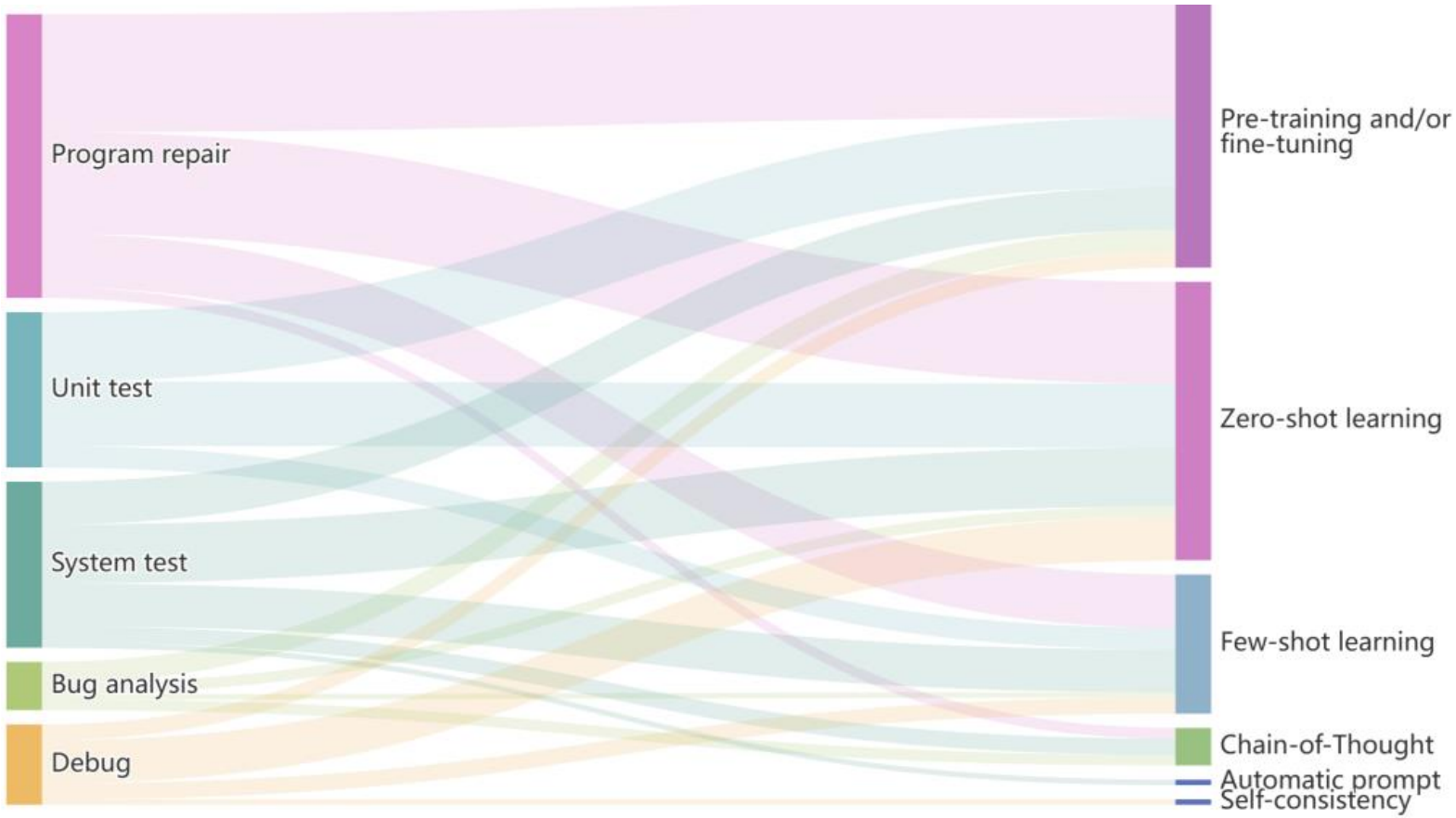
```
- 仕様を要約しテスト範囲・目的を明確化
- テストアイテム（モジュール / 画面 / 機能）を抽出し，ITEM-XXX 形式で ID 付与
- 要求・非機能要求・リスクを抽出し，REQ/NF/RISK-XXX 形式で連番 ID 付与
- 不明点は「不明」と記録，後日確認
- 出力形式：
 ・テストアイテム一覧（ITEM-XXX, 名称 / 範囲）
 ・要求・リスク一覧（REQ/NF/RISK-XXX, 名称 , 概要）
 ・テスト目的・完了基準（箇条書き可）
```

```
- 全 REQ/RISK が最終的に詳細ケース (D-XXX) まで到達しているか，AI で差分チェック
- 指摘があれば Step2 ～ 4 を再生成し，最終レビュー後にテスト実行へ移行
- 差分チェック例：「要求 / リスク一覧 vs 詳細ケース一覧で未カバー ID は？」
- 入力：Step1 ～ 4 の成果物（ID 付き一覧），レビュー指摘事項・追加要求
```

情報処理学会 監修, 鷲崎弘宜 編著, 家村康佑, 鵜林尚靖, 蝦名拓也, 近藤将成, 徳本晋, 中川尊雄, 増田航太, 石川冬樹, 竹之内啓太, 小川秀人, 川上真澄, “生成AIによるソフトウェア開発: 設計からテスト、マネジメントまでを全て変革するLLM活用の実践体系”, オーム社, 2025

LLMのテストや周辺への活用動向 [Wang+24]

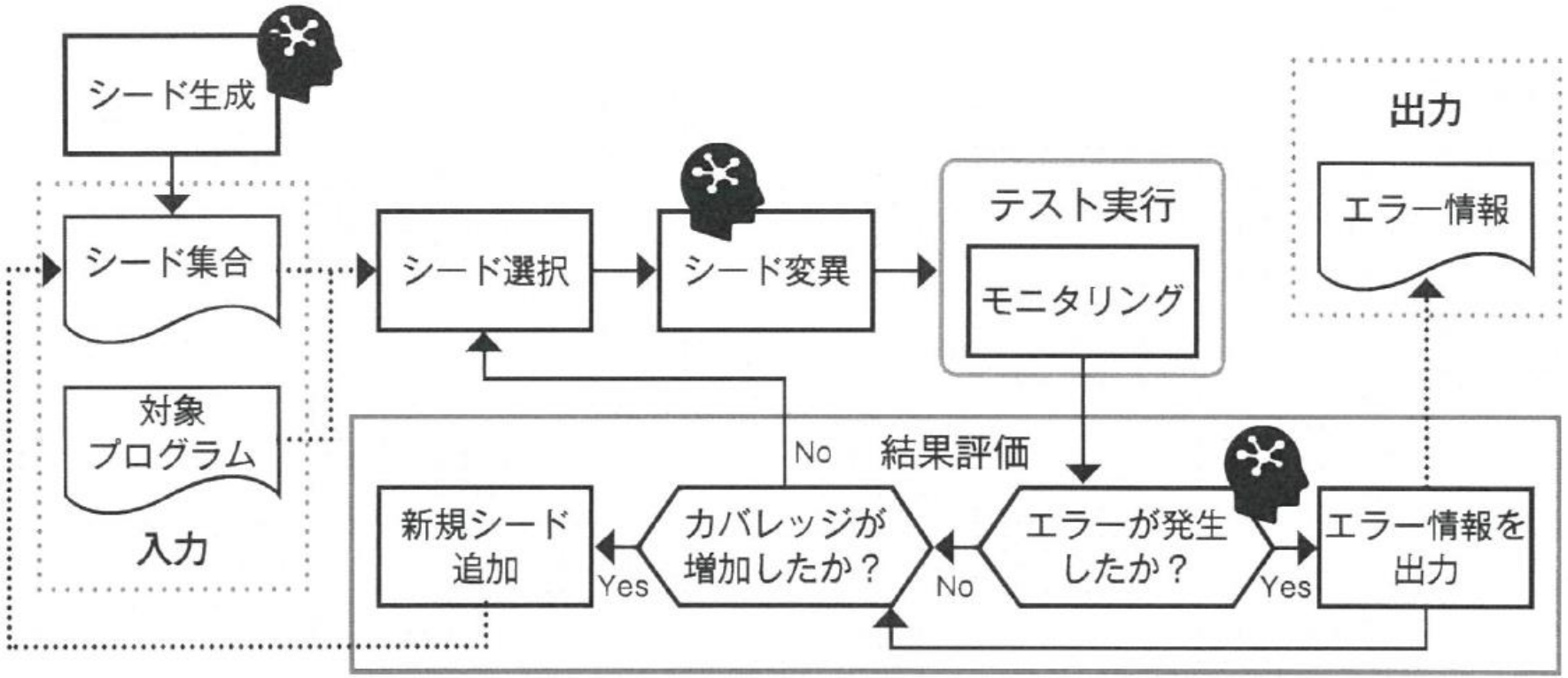




J. Wang, et al., Software Testing with Large Language Models: Survey, Landscape, and Vision, TSE, 2024

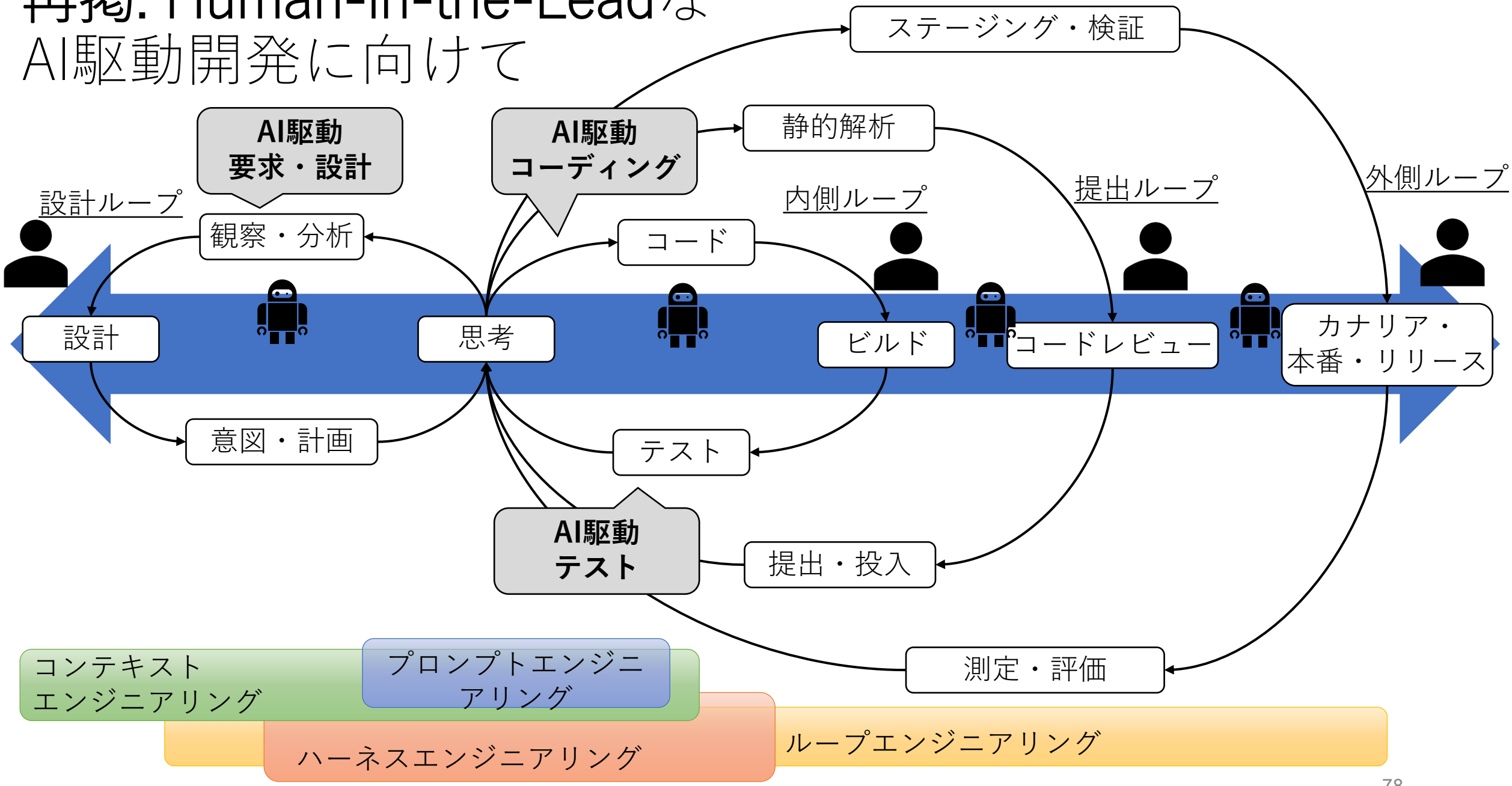
テスト周辺動向

- ファジング
  - ChatAFL [Meng+24]
  - Fuzz4All [Xia+24]
- GUIテスト
- APIテスト
- 静的解析
- デバッグ支援
  - 欠陥位置特定
  - 自動修正



R. Meng. “Large language model guided protocol fuzzing,” NDSS, 2024.
C.S. Xia, et al., “Fuzz4All: Universal Fuzzing with Large Language Models,” ICSE 2024.
情報処理学会 監修, 鷲崎弘宜 編著, 家村康佑, 鵜林尚靖, 蝦名拓也, 近藤将成, 徳本晋, 中川尊雄, 増田航太, 石川冬樹, 竹之内啓太, 小川秀人, 川上真澄, “生成AIによるソフトウェア開発: 設計からテスト、マネジメントまでを全て変革するLLM活用の実践体系”, オーム社, 2025

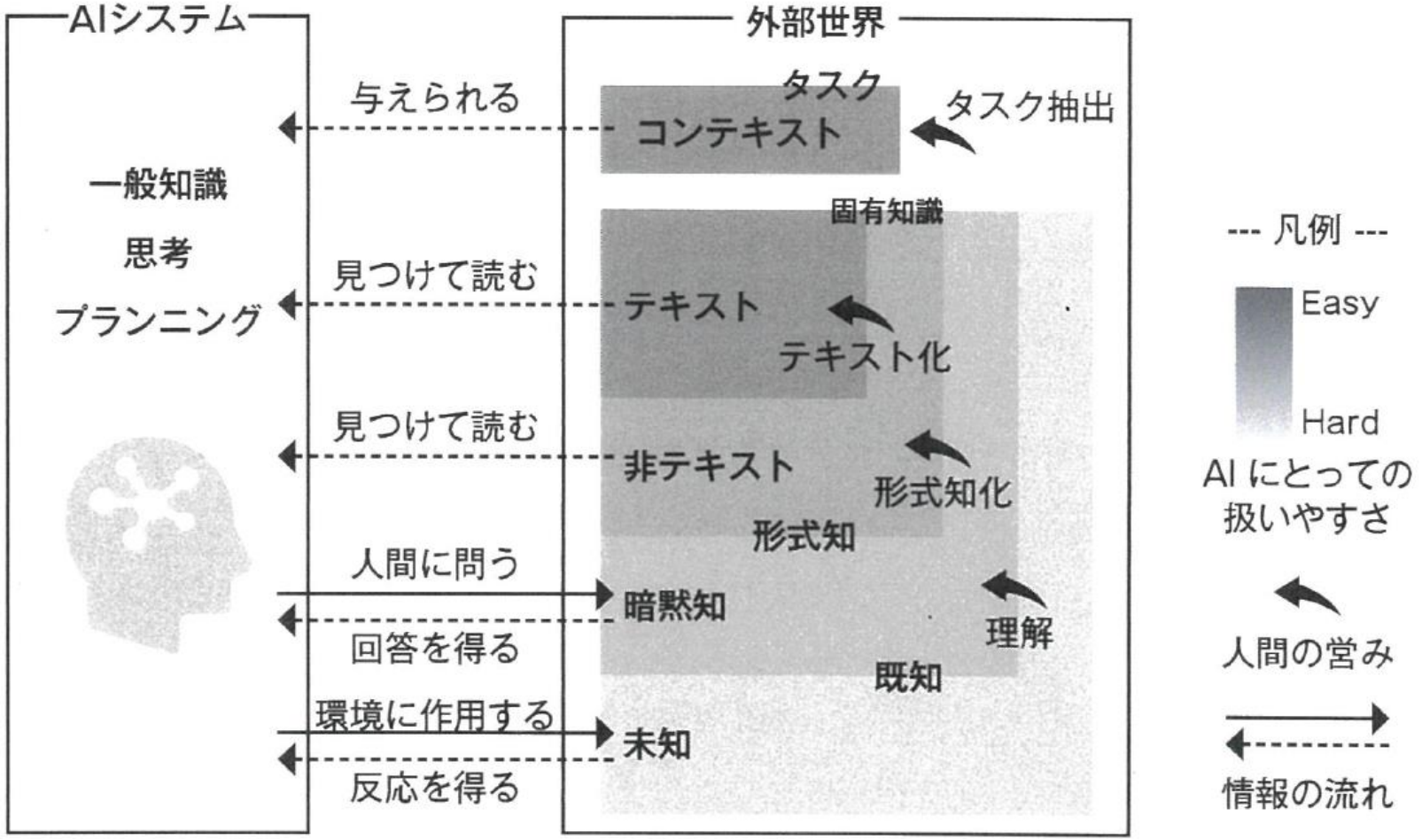
再掲: Human-in-the-Leadな AI駆動開発に向けて



Addy Osmani, “Beyond Vibe Coding: From Coder to AI-Era Developer,” O’Reilly, 2025, https://beyond.addy.ie/

導入・マネジメントに向けて

- ユースケースに適した生成 AI 技術の導入により開発作業の効率を向上
- 開発者の作業を AI で代替するためには開発者の知識を再現することが重要



出典：Responsible Software Development in the Era of Generative AI, NTT DATA Group, 2025
出典: 竹之内 啓太,生成AIに基づくソフトウェア開発のプロセスとマネジメント, スマートエスイーセミナー 2025年12月
情報処理学会 監修, 鷲崎弘宜 編著, 家村康佑, 鵜林尚靖, 蝦名拓也, 近藤将成, 徳本晋, 中川尊雄, 増田航太, 石川冬樹, 竹之内啓太, 小川秀人, 川上真澄, “生成AIによるソフトウェア開発: 設計からテスト、マネジメントまでを全て変革するLLM活用の実践体系”, オーム社, 2025

AIエージェント駆動開発時代のチャレンジ

- 高信頼・レスポンシブルな「AIエージェント」に向けて
  - 制御性、説明可能性、オープン・透明性、一貫性を通じた信頼
  - バイアスとデータ品質、計算コスト・消費電力
  - 汎用の大モデルとドメイン・タスク固有の小モデルの役割分担と連携
  - 法的および倫理的なコンプライアンス
  - エンジニアリング: プロンプト→コンテキスト→ハーネス→ループ
- 開発者の役割シフト
  - コード中心から意図中心へ
  - 全体の設計と検証・評価
  - コンテキスト管理、AIエージェント統制
- ライフサイクルに沿った活用と組織
  - 要求、設計、実装（Vibe Coding）、テストにおける活用
  - 協調プロセスモデル: 自動化、反復対話、計画駆動、テスト駆動、コンテキスト強化
  - 組織的な知識創造・循環の仕組み、 基礎的組織能力＆レディネス

まとめ

- AIと品質の関係
  - 規格やガイドライン、特に品質モデル
- AIシステムの品質保証
  - 設計による品質
  - 論証とリスク解析
  - テスト&性能調整
  - 生成AIの評価
- 生成AI＆エージェント駆動開発における品質
  - エンジニアリングの変遷: プロンプト→コンテキスト→ハーネス→ループ
  - 品質考慮の要求・設計
  - Vibe Codingと品質
  - テスト

参加のお誘い: ソフトウェア品質管理研究会（SQiP研究会）

- https://www.juse.or.jp/sqip/workshop/
- 生成AI&エージェント時代のソフトウェア品質の学習、研究、実践
- 業界第一人者による指導、ネットワーキング＆仲間づくり
- 11月に公開説明会 OpenDayを開催予定