

AIプロダクト品質保証ガイドライン

2024.01版



AIプロダクト品質保証コンソーシアム
(QA4AIコンソーシアム) 編

本ガイドラインについて

本ガイドラインは、AI プロダクトの品質保証に対する共通の指針を与えるものです。

機械学習に代表される AI 技術はいまだ発展途上のため、本ガイドラインは網羅性や完全性を企図したものではありません。したがって各組織では、本ガイドラインを指針として自ドメインや自社、自組織の状況などを熟慮し反映する必要があります。

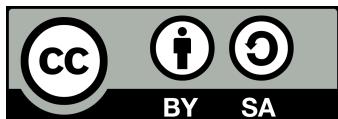
本ガイドラインに沿って開発または提供された AI プロダクトの品質について、本ガイドラインの作成元である AI プロダクト品質保証コンソーシアムおよび本コンソーシアムに属する個人または団体が責任を持つものではありません。

本ガイドラインは、本コンソーシアムに属する個人およびその所属する団体もしくは本コンソーシアムに属する団体およびその上位団体等の公式な見解を示すものではありません。

著作権表示

本ガイドラインの著作権は AI プロダクト品質保証コンソーシアムが保持します。

本ガイドラインは、クリエイティブ・コモンズ（表示 - 繙承 4.0 国際）ライセンスの下で提供されます。



来歴

2019 年 5 月 17 日	2019.05 版公開	初版
2020 年 2 月 1 日	2020.02 版公開	AI-OCR の章を追加
2020 年 8 月 1 日	2020.08 版公開	改訂版
2021 年 9 月 15 日	2021.09 版公開	改訂版
2022 年 7 月 15 日	2022.07 版公開	改訂版
2023 年 6 月 2 日	2023.06 版公開	改訂版
2024 年 1 月 17 日	2024.01 版公開	大規模言語モデル・対話型生成 AI の章を追加

2024.01 版の主な改訂点は以下である：

- 5 章 生成系システムの更新
- 10 章 生成系 AI(LLM) の追加

目次

1 目的とスコープ	1-1
1.1 背景と目的	1-1
1.2 AI プロダクトの品質保証上の課題と本ガイドラインのスコープ	1-2
2 AI プロダクトの品質保証の枠組み	2-1
2.1 AI プロダクトの品質保証の基本的考え方	2-1
2.1.1 AI プロダクトの品質保証において考慮すべき軸	2-1
2.1.2 Data Integrity	2-2
2.1.3 Model Robustness	2-3
2.1.4 System Quality	2-4
2.1.5 Process Agility	2-6
2.1.6 Customer Expectation	2-7
2.2 AI プロダクトの品質保証の分類軸ごとのチェックリスト	2-9
2.2.1 Data Integrity	2-9
2.2.2 Model Robustness	2-11
2.2.3 System Quality	2-12
2.2.4 Process Agility	2-13
2.2.5 Customer Expectation	2-14
2.3 AI プロダクトの品質保証の構築・評価	2-14
2.3.1 バランスに着目した構築・評価	2-14
2.3.2 開発段階に着目した構築・評価	2-16
2.3.3 余力と過剰品質	2-17
3 技術カタログ	3-1
3.1 AI プロダクト固有の品質特性	3-2
3.1.1 教師あり学習のモデルに対する性能指標	3-2
3.1.2 データに対する評価	3-4
3.1.3 頑健性	3-4
3.1.4 公平性	3-4
3.1.5 説明可能性	3-5
3.1.6 機械学習を用いたシステム全体における品質	3-5

3.2 AI プロダクトにおける品質管理	3-6
3.3 AI プロダクトの品質保証技術	3-7
3.3.1 疑似オラクル	3-7
3.3.2 メタモルフィックテスティング	3-7
3.3.3 頑健性検査	3-8
3.3.4 ニューラルネットワークにおけるカバレッジ	3-8
3.3.5 説明可能性・解釈性のための技術	3-8
3.4 参考文献	3-9
4 機械学習における説明可能性・解釈性	4-1
4.1 はじめに	4-1
4.2 説明可能性・解釈性を付与する手法の分類	4-1
4.3 説明可能性・解釈性を付与する代表的手法	4-7
4.3.1 GLM/GAM	4-7
4.3.2 DT	4-8
4.3.3 Surrogate	4-9
4.3.4 TCAV	4-10
4.3.5 Attention	4-12
4.3.6 Sensitivity Analysis	4-14
4.3.7 CAM	4-17
4.3.8 LIME	4-19
5 コンテンツ生成系システム	5-1
5.1 想定するシステム	5-1
5.1.1 本章で扱う応用領域	5-1
5.1.2 背景 - 近年の AI による生成モデルの進展 -	5-5
5.1.3 議論に用いるユースケース	5-7
5.2 特有の課題	5-9
5.3 期待される品質特性	5-9
5.3.1 全ユースケースに共通する品質特性	5-9
5.3.2 コンテンツの指定に関する品質特性	5-10
5.3.3 動画に関する品質特性	5-10
5.4 品質評価・保証のための技術アプローチ	5-10
5.4.1 指標による評価	5-11
5.4.2 機械学習による品質評価 AI の構築	5-12

5.4.3 ルールベースの AI など他実装との比較	5-14
5.5 品質保証レベル	5-14
5.6 テスト設計事例	5-16
5.6.1 対象システムの概要	5-16
5.6.2 対象となる品質特性	5-17
5.6.3 テスト設計	5-18
5.6.4 実験	5-19
5.6.5 テスト設計事例のまとめ	5-20
5.7 Text-to-Image に対する評価事例	5-21
5.8 参考文献	5-22
6 Voice User Interface (VUI)	6-1
6.1 想定するシステム	6-1
6.2 VUI システムの特徴	6-2
6.2.1 音声認識	6-2
6.2.2 自然言語理解	6-3
6.2.3 音声合成	6-4
6.2.4 その他 - インフォテインメント	6-4
6.3 特有の課題	6-4
6.3.1 システムの課題	6-4
6.3.2 Process Agility の課題	6-5
6.4 VUI における機能安全	6-7
6.4.1 セーフティ	6-7
6.4.2 外部機器の操作	6-8
6.5 期待される品質	6-8
6.5.1 5 つの軸から見た期待される品質	6-8
6.5.2 音声出力の UX	6-10
6.6 テストアーキテクチャ	6-12
6.7 有効な手法	6-19
6.7.1 n 段階評価法	6-19
6.7.2 スモークテスト	6-20
6.7.3 音声認識の認識精度の評価方法	6-21
6.7.4 自然言語理解のテストケース	6-21
6.7.5 社内ユーザーテスト	6-24
6.7.6 データ変更時の精度評価、モデル変更時の精度評価	6-24

6.8	品質保証レベル	6-24
6.9	VUI における個人情報およびプライバシーの保護	6-26
6.9.1	個人情報保護法	6-27
6.9.2	プライバシー保護	6-28
6.9.3	その他音声データの取り扱いに関する考慮できる内容	6-30
7	産業用プロセス	7-1
7.1	検討の前提と対象	7-1
7.1.1	本章で用いる用語定義	7-4
7.2	産業用システムへの AI 技術適用にあたっての重点課題	7-5
7.3	参照システムアーキテクチャ	7-6
7.4	想定ステークホルダー	7-8
7.5	品質保証活動	7-9
7.6	産業用システムにおける 5 つの指標の具体化	7-12
7.6.1	QA4AI 品質保証観点の解釈	7-12
7.6.2	Customer Expectation	7-13
7.6.3	Data Integrity	7-29
7.6.4	Model Robustness	7-37
7.6.5	System Quality	7-44
7.6.6	Process Agility	7-59
7.7	AI プロダクト開発プロセスでの品質保証観点	7-69
7.7.1	PoC	7-69
7.7.2	開発工程	7-70
7.7.3	運用工程	7-75
7.8	モチーフへの品質保証の適用	7-77
7.8.1	AI を活用する背景	7-77
7.8.2	モチーフにおけるステークホルダー	7-77
7.8.3	製品改良プロジェクトの全体の流れ	7-78
7.8.4	PoC 工程における開発の流れと品質保証	7-79
7.8.5	開発工程における開発の流れと品質保証	7-87
7.8.6	運用工程における開発の流れと品質保証	7-93
8	自動運転	8-1
8.1	はじめに	8-1
8.1.1	全体構成	8-1

8.1.2	背景	8-1
8.1.3	目的	8-1
8.1.4	想定読者	8-2
8.1.5	前提知識とスキル	8-2
8.1.6	想定対象製品	8-2
8.2	用語集	8-2
8.3	自動運転の前提知識	8-2
8.3.1	自動運転レベル	8-2
8.3.2	一般的な自動運転システム構成	8-4
8.3.3	AI が使われている自動運転の機能	8-5
8.4	特有の課題と対策	8-6
8.4.1	自動運転システムの特徴	8-6
8.4.2	AI プロダクト品質保証上の課題	8-7
8.4.3	AI 開発プロセスにかかる課題	8-8
8.4.4	課題に対する考え方とアプローチ	8-9
8.5	自動運転における AI 品質保証の考え方	8-11
8.5.1	自動運転における AI 品質保証の考え方とアプローチ	8-11
8.5.2	ML モデルの品質保証	8-12
8.5.3	ML モデル開発と既存開発プロセスの I/F	8-15
8.5.4	ML モデルの既存開発プロセス上での扱い	8-17
8.5.5	ML モデルを統合したシステムの V&V	8-18
8.6	ML 品質要求事項	8-19
8.6.1	ML 品質要求の考え方とアプローチ	8-19
8.6.2	ML モデル要求事項の構造と分類	8-20
8.6.3	ML 品質要求事項	8-22
8.7	ML モデル開発プロセス	8-29
8.7.1	ML モデル開発に必要な開発プロセス	8-29
8.7.2	ML モデル開発プロセス群（例）	8-30
8.7.3	ML モデルの Process Flow Diagram	8-33
8.7.4	ML モデル開発のトレーサビリティと一貫性	8-34
8.7.5	関連プロセスへの考慮事項	8-34
8.8	ML 関連開発と品質保証	8-35
8.8.1	自動運転開発における ML 関連開発活動と品質確認観点	8-35
8.8.2	ML コンポーネントの品質保証	8-36
8.8.3	安全性要求への対応アプローチ	8-39

8.9 付録	8-40
8.9.1 自動運転関連標準	8-40
8.9.2 画像認識 AI プロダクトのテスト技術	8-42
9 AI-OCR	9-1
9.1 本章の背景と目的	9-1
9.2 前提となるシステム構成	9-2
9.2.1 前処理モジュール	9-3
9.2.2 文字位置検出モジュール	9-3
9.2.3 OCR モジュール	9-4
9.2.4 項目抽出モジュール	9-4
9.3 AI-OCR 特有の課題と考慮すべき点	9-4
9.3.1 AI-OCR における Data Integrity	9-5
9.3.2 AI-OCR における Model Robustness	9-7
9.3.3 AI-OCR における Process Agility	9-9
9.3.4 AI-OCR における System Quality	9-10
9.3.5 AI-OCR における Customer Expectation	9-11
9.4 品質保証技術の AI-OCR 適用例	9-12
9.4.1 メタモルフィックテスティングを適用した品質評価例	9-12
9.4.2 帳票項目分析を利用した品質評価事例	9-13
9.5 推奨する品質評価レベル	9-22
10 大規模言語モデル・対話型生成 AI	10-1
10.1 LLM・対話型生成 AI の概要	10-1
10.1.1 構成と動作	10-1
10.1.2 活用のユースケースと留意点	10-4
10.1.3 典型的な懸念	10-5
10.2 LLM における品質特性	10-7
10.2.1 QC01：回答性能	10-9
10.2.2 QC02：事実性・誠実性	10-10
10.2.3 QC03：倫理性・アライメント	10-11
10.2.4 QC04：頑健性	10-13
10.2.5 QC05:AI セキュリティ	10-13
10.2.6 その他の品質観点	10-14
10.3 一般的な LLM に対する品質評価手法	10-15

10.3.1 QC01：回答性能の評価	10-16
10.3.2 QC02：事実性・誠実性の評価	10-18
10.3.3 QC03：倫理性の評価	10-19
10.3.4 QC04：頑健性の評価	10-20
10.3.5 QC05：AI セキュリティの評価	10-20
10.4 LLM を用いた個別システムに対するカスタムの品質と評価	10-20
10.4.1 対象システムが扱うタスク固有の回答性能評価	10-21
10.4.2 対象システム固有の知識に関する事実性・誠実性の評価	10-21
10.4.3 対象システム固有のリスクに関する AI セキュリティの評価	10-21
10.4.4 自動評価の実現手段と評価内容	10-21
10.4.5 利用する自然言語	10-22
10.5 QA4AI ガイドラインの 5 軸の品質特性に対する生成 AI の特徴	10-22

11 AI プロダクト品質保証コンソーシアムについて

11-1

付録A エックリストの新旧対照表

A-1

1. 目的とスコープ

1.1 背景と目的

機械学習をはじめとする AI 技術は進化しながら普及の一途を辿り、様々な産業の競争力の源泉となるだけでなく、既存の産業構造を破壊・変革し、新たな産業を創出するようになってきている。それに伴い、AI 技術を用いた製品やサービス（AI プロダクト）が生活や社会、経済に及ぼす影響も大きくなっている。

その一方で、AI 技術は進化途上であるとともに、ハードウェアや従来型のソフトウェア、サービスなどに比べ、その技術的特質により、品質の把握、評価、説明、管理など品質保証が非常に難しい。特に機械学習ではデータの学習によりふるまいが帰納的に決定されるため、従来型のソフトウェアに対する品質保証手段が利用できない。開発プロセスの管理による品質保証が寄与する割合も小さい。したがって AI プロダクトの品質保証技術が確立されたとは到底言いがたい状況にある。すなわち我々の生活や社会、経済には、AI プロダクトの品質事故という甚大なリスクが内在されているのである。

同時に注意すべきなのは、AI 技術が持つ技術的特質を無視し AI プロダクトの品質に社会が過度の期待を持つことが、品質保証のための過度な活動を要請し、AI 技術の適切な活用や適時のリリース、さらなる進化を妨げる圧力を与えてしまう点である。我々は AI プロダクトの品質リスクを軽減するとともに、過度の品質圧力を予防し、AI 技術が安心して活用され進化できるようにする必要がある。

したがって、我々が安心・安全な生活や社会、経済を営んでいくためには、AI プロダクトに対する品質保証技術の調査・体系化、適用支援・応用、研究開発が急務である。同時に、AI プロダクトの品質に対して、技術的特質を踏まえた適切な理解を社会が持ちうるような啓発活動も進めいかねばならない。

そこで我々は、AI プロダクトの品質保証のためのガイドラインを発行する。このガイドラインは、各組織において AI 技術への過度の期待を予防し、適切な活用や適時のリリースを行うための、AI プロダクトの品質保証に対する共通的な指針を与えるものである。

機械学習に代表される AI 技術は著しく速く進化しているため、本ガイドラインは年次程度に定期的に更新される。各組織や産業において標準的な文書を作成したり適合性の程度を評価したりする場合には、その点について十分に考慮する必要がある。

1.2 AI プロダクトの品質保証上の課題と本ガイドラインのスコープ

AI プロダクトの基盤となる AI 技術には大きく分けて、演繹的に開発できるルールベース技術と帰納的に開発する機械学習技術がある。前者の品質保証は伝統的な品質保証技術によって可能になるが、後者の品質保証は伝統的な品質保証技術では困難である。

演繹的開発とは、定義された仕様に対して内部設計や実装を明示的に関連付けることができ、それらに基づいて様々な検証を行うという進め方の開発である。またその関連に従って開発プロセスとプロダクト品質との関係が比較的明確になっている。したがって品質を向上するための経験が知識として蓄積されており、内部設計や実装のレビューやメトリクス、プロセスに着目した品質保証という手段が有効となる。

一方、帰納的開発では、定義された仕様に対して内部設計や実装を明示的に関連付けることができない。すなわち内部設計や実装の良し悪しのレビューが極めて難しく、統計的なメトリクスによって品質を評価することもできない。同様に、開発プロセスとプロダクト品質との関係が不明になりやすいため、プロセス品質保証という手段は極めて限定的となる。例えば残存バグ数を推定することは難しいし、プロセス監査は期待通りの効果を及ぼさない。

そこで帰納的開発では、探索的開発と呼ぶべきスタイルで開発を進めることが多い。少しづつ学習を進めたりプロダクトを構築したりするなど、小規模かつ反復的に開発を進め、テストや試験稼働、実運用によって品質が向上・確保できていることを実証するスタイルである。

機械学習技術には、線形性や分布を仮定できる技術とできない技術があり、現在 AI プロダクトに用いられている技術は後者の非線形的技術が非常に多い。またニューラルネットワークなどでは多くのニューロン同士が非常に複雑な構造を持つ。

品質保証とはあらゆる条件における動作の正当性を将来に渡って予測することによって達成するものであるが、コストや納期の面から網羅的な条件を考慮することは現実的に不可能であるし、将来的予測は原理的に未知となる。そこで従来は、何らかの線形性や分布を仮定し分割統治を行うことによって、網羅的でなく典型的な条件のみを考慮し、小規模なコンポーネントの品質保証成果を積み上げながら、現在達成された品質に基づいて将来の品質を保証することにより、現実的なコストや納期で品質を保証してきた。

一方、非線形的技術や分布が仮定できず、非常に複雑な構造を持つ技術を用いる場合は CACE 性と呼ばれる性質（Changing Anything Changes Everything: 少しでも変更すると全体に影響が及ぶ性質）を呈し、線形性や分布、分割統治に基づかず、学習や変更が行われる度に品質を保証する必要がある。すなわち、コンポーネント全体に対して全ての条件で高頻度にテストを行う全体全数高頻度検証（FEET: Frequent, Entire and Exhaustive Testing）が必要となる。FEET にはテストや構成管理といった品質保証技術の自動化が欠かせない。

同時に、非線形で分布が仮定できず非常に複雑な構造を持つため、ある一群の学習によって誤判

別が修正された場合であっても、その因果関係の説明や理解は極めて困難である。こうした説明可能性などに関する技術は eXplainable AI (XAI) として現在盛んに研究されている。

AI プロダクトの品質保証においては、まず AI コンポーネントの核となるモデルの質と、モデルを決定づけるデータの質がまず重要となる。モデルの質やデータの質についての議論は、統計学や機械学習の分野で固有技術として盛んに議論されている。品質保証についてもそれらを踏襲すればよいが、次元の大きなデータや取得可能な実データ、オンライン学習といった実務的な側面も考慮する必要がある。ミッションクリティカルなドメインにおいては、基本的にモデルの精度は 100% にならないという点も理解しておく必要がある。

AI プロダクトの開発組織は、データサイエンスとソフトウェア開発という 2 つの側面を持つ。データサイエンス系の位置付けの強い開発組織では、モデルの精度こそが品質だと矮小化している場合がある。ソフトウェア開発系の位置付けの強い開発組織では、プロセスやメトリクスこそが品質保証だと盲信している場合がある。前者の組織が AI プロダクトの品質保証について論ずる場合には、システム全体としての AI プロダクトの品質をどのように保証するかの視座で捉える必要がある。例えばシステムとしての価値の把握、想定される品質事故の致命度の評価、品質事故を引き起こしうる事象の発生頻度の見積もりなど、演繹的開発の品質保証の考え方方が役立つことを認識する必要がある。

後者の組織が AI プロダクトの品質保証について論ずる場合には、そもそも高い品質を達成できている組織とできていない組織との比較という視座で捉える必要がある。演繹的開発において、プロセスやメトリクスを遵守していても品質が低い組織は多々見られる。一方、技術力の高い開発者が強く納得し、チームや組織において、また顧客とともにその納得感を強く共感している組織は総じて品質が高い。プロセスやメトリクスを遵守するから納得感の共感が強いのではない。開発チームが強く納得感を共感できているから、結果として技術的に必要十分なプロセスで開発を行いメトリクスを達成しているのである。こうした組織は探索的に開発を行うことで、納得感の共感を強めやすくしている傾向が高い。

AI プロダクトの品質保証における大きな懸念点がもう一つある。AI プロダクトの特性について理解が乏しい顧客である。そもそも演繹型開発か帰納型開発かに関わらず、期待が大きい場合には、品質保証をよりしっかりと行う必要がある。さらに AI プロダクトの特性についての理解が乏しい顧客の場合には、品質保証が困難になるリスクを抱える可能性がある。

AI プロダクトの特性について理解が乏しい顧客は、「自分が特に何もしなくても AI プロダクトや開発組織が常によろしく完璧にやってくれる」と考えている場合がある。もちろん、これは誤った考え方である。こうした顧客は、データの質や量に関しての認識が甘かったり、探索的開発を許容せず必要な権限を与えなかったり、顧客側組織の変化の必要性を拒絶したりする。また 100% の精度を求めたり、非線形なふるまいを拒絶したり、モデルのふるまいに関する合理的で詳細な説明を求めたりする。こうした顧客の理解を適切に制御することが、AI プロダクトの品質保証において重要なとなる。

本ガイドラインでは、以上のような AI プロダクトの品質保証上の課題を考慮していく。まず基本的な考え方や考慮事項といった枠組みを 2 章で述べる。AI プロダクトの品質保証の構築や評価において考慮すべき 5 つの軸である Data Integrity、Model Robustness、System Quality、Process Agility、Customer Expectation を提示し、それらのバランスや余力について論ずる。次に、AI プロダクトの品質保証を推進していくための技術をカタログとして整理する。また、関連して近年特に注目されている機械学習における説明可能性・解釈性に関する技術動向を取り上げる。そして、それらを基盤としてコンテンツ生成系システム、スマートスピーカー、産業用プロセス、自動運転の 4 つのドメインに対する個別のガイドラインを例示していく。

機械学習に代表される AI 技術は著しく速く進化しているため、本ガイドラインは年次程度に定期的に更新される。各組織や産業において標準的な文書を作成したり適合性の程度を評価したりする場合には、その点について十分に考慮する必要がある。

同様に AI の技術はいまだ発展途上そのため、本ガイドラインは網羅性や完全性を企図したものではない。したがって各組織では、本ガイドラインを指針として自ドメインや自社、自組織の状況などを熟慮、反映し、自組織の責任の下において活用する必要がある。

なお、AI プロダクトの品質保証を議論するとき、契約において品質保証をどのように規定すべきかという問題も避けて通れない。本ガイドラインは、契約上の問題については、現時点では基本的にスコープに含めないが、以下、簡単に検討の視点を示す。

品質「保証」という表現は、ベンダーが、開発する AI プロダクトの品質に責任を負うという文脈でとらえられるがちであるが、本ガイドラインの品質保証の内容は、性質上、ベンダーのみがその責任を負うものではなく、また、責任の内容も様々である。例えば、AI プロダクトが第三者の著作権を侵害していた場合にはベンダーが責任を負うべき（ベンダーが著作権の非侵害を保証すべき）ことも多いと思われる。他方で、学習に用いるデータを提供することや、当該データを適法に利用できる状態で提供することはユーザーが責任を負うべきことが多い。また、AI プロダクトが確率的に動作することなどの不可避的なリスクについては、どちらが義務を負うかという観点ではなく、リスクについて適切な説明がなされたか、適切なリスクの分担が定められているか、といった観点から検討されるべきであると思われる。

一般的に、システム開発では、判例等において、ベンダーにプロジェクトマネジメント義務、ユーザーには協力義務がそれぞれ課せられうることが指摘されてきた。AI プロダクトの開発においても、ユーザー・ベンダーの双方が義務を負うこと、その義務の性質は様々であることを前提に検討がなされるべきであろう。この点については、今後、本ガイドラインのチェックリスト等を手掛かりに、それぞれの項目について誰がどのような責任を負うべきか、契約にはどのように反映しうるかといった観点から議論の深化が期待されるところである。

また、請負契約において AI プロダクトの品質が契約内容に適合しないと考えられる場合や、準委任契約においてベンダーに開発過程に善管注意義務違反があると考えられる場合には、ユーザーに対するベンダーの責任が問題になりうる。しかし、契約において品質について定めがない場合には、

どの水準を下回ればベンダーに責任が生じるのかが明らかでない。その結果、不明確さゆえにトラブルが生じることも考えられるなど、ユーザー・ベンダーの双方にとって好ましくない状況に陥りかねない。そのようなリスクを排除するためにも、契約において AI プロダクトの品質をどのように定めるべきかについて、踏み込んだ議論が必要になる。

2. AI プロダクトの品質保証の枠組み

2.1 AI プロダクトの品質保証の基本的考え方

2.1.1 AI プロダクトの品質保証において考慮すべき軸

まず品質の高い AI プロダクトを開発するためには、データがきちんととしていなくてはならない。すなわち、質においても量においても適切かつ充分なデータの確保が重要であり、学習用データと検証用データが独立している必要がある。この視点から考慮すべき軸を本ガイドラインでは Data Integrity と呼ぶこととする。

同様に品質の高い AI プロダクトを開発するためには、モデルがきちんととしていなくてはならない。すなわち、精度が高く頑健性が確保されたモデルが重要となる。また学習などにおいてデグレードに適切に対処する必要がある。この視点から考慮すべき軸を本ガイドラインでは Model Robustness と呼ぶこととする。ここで、本ガイドラインにおいて「モデル」はいわゆる学習済みモデル、いわばモデルのインスタンスを指すこととし、モデルの種類を「アルゴリズム」と呼ぶこととする。本ガイドラインでは単一のアルゴリズムによるモデルだけでなく、複合したアルゴリズムによるモデルも対象とする。機械学習のコンポーネントを一切使わない AI プロダクトの場合、品質保証は演繹的開発の品質保証でも有効となるため、本ガイドラインでは明示的に取り扱わない。とはいえ非機械学習の AI プロダクトに対しても品質保証はもちろん必要であるし、本ガイドラインのうち機械学習のコンポーネントに固有でない記述は同様に有用だろう。

また品質の高い AI プロダクトを開発するためには、システム全体として価値が高く、何かが起きても何とかならないといけない。すなわち、AI プロダクト全体の品質が確保できていることを保証することが重要となる。この視点から考慮すべき軸を本ガイドラインでは System Quality と呼ぶこととする。ただし本ガイドラインでは、狭義および広義の Quality、Reliability、Dependability、Safety、Security などの類似した概念の上位下位関係を定義することは意図せず、こうした概念の集合として品質や Quality という概念を用いることとする。したがってドメインによっては Safety や Security、Dependability などと読み替えると有用だろう。

そして品質の高い AI プロダクトを開発するためには、納得感を共感した開発者や開発チームが自動化された開発環境を駆使して臨機応変に探索的開発を進めていく必要がある。すなわち、プロセスが機動的であることが重要となる。この視点から考慮すべき軸を本ガイドラインでは Process Agility と呼ぶこととする。

品質の高い AI プロダクトを開発するために忘れてはならないのは、よい顧客との関係性である。すなわち、良くも悪くも顧客の期待が高いかどうかが重要となる。良い意味で顧客の期待が高いと

品質保証をしっかりやる必要があるし、悪い意味で顧客の期待が高いと AI プロダクトの特性についての理解が乏しい顧客によって品質保証が困難になるリスクに対処することになる。この視点から考慮すべき軸を本ガイドラインでは Customer Expectation と呼ぶこととする。

次節ではこれらの 5 つの軸にしたがって、AI プロダクトの品質保証を評価・構築のための考慮事項を提示する。ただし機械学習分野の技術はいまだ発展段階にあるため、提示する考慮事項が網羅的であるとは限らない。本ガイドラインを利用する各組織では、これらの 5 つの軸を活かしながら考慮事項を適宜追加する必要がある。

2.1.2 Data Integrity

Data Integrity では、質においても量においても適切かつ充分なデータの確保と、学習用データと検証用データが独立しているかどうかなどについて考慮する。Data Integrity については統計学や機械学習の分野で固有技術として盛んに議論されているので、品質保証についてもそれらを踏襲すればよいだろう。

まずデータの量が充分であり、コストが適正である必要がある。ただし意味のある量でなくてはならない。例えば画像データの明度や色彩を変えるなど、あるデータに演算や変換を行って別のデータとする「かさ増し」を行うことがあるが、それによって汎化性能が落ちる場合は意味のある量の増加にならない。

データの質という点では、サンプルに対する統計的な性質を満たしているかを考慮する必要がある。求める母集団に属するサンプルなのかどうか、実際のデータなのか人為的に作成されたデータなのか、不要なデータやノイズ、異なる母集団のデータが含まれていないか、などを考慮する必要がある。また偏りやバイアス、汚染は無いか、自分たちが考えている偏りの源だけでよいのか、などを考慮する必要もある。

AI プロダクトの扱うデータは画像など高次元かつ複合的であることが多いので、必要な要素を適切に含んだサンプルなのか、複雑すぎないか単純すぎないか、なども考慮する必要がある。多重共線性といったデータ内の性質についても検討する必要がある。教師あり学習の場合は、ラベルの妥当性についても考慮する必要がある。

また外れ値や欠損について考慮する必要がある。それぞれのデータは常識的な値なのか、外れ値は本当に外れているデータであって意味を持たないのか、欠損そのものや欠損の理由や背景に意味は無いのか、外れ値や欠損値の扱いは適切なのかどうか、などを考慮する必要がある。

AI プロダクトは学習と検証を繰り返すため、学習用データと検証用データを独立させるべきであることが多い。その際には、独立性を担保する仕組みや機密性などについても考慮する必要がある。人為的に作成したデータの場合は、データ生成プログラムの品質も保証する必要がある。同様に、学習用プログラムの品質など学習の過程も保証する必要がある。さもないとデータの意味が毀損されてしまう。

またオンライン学習を行う際には、どのようなデータが与えられどのような学習を行う可能性があり、それによってどのような影響が発生するのか、についても検討しておかなくてはならない。

以上の点に加え、実際のデータを用いる場合には、法的・倫理的観点からも検討が必要である。具体的には、契約による制限、第三者の権利による制限、法令による制限、倫理やプライバシー等の問題による制限といった観点から、データの利用に制限がないかを検討すべきである。

すなわち、まず、契約において秘密保持条項やデータの利用条件が定められているなど、契約上の義務が課せられることからデータの利用が制限されることがある。

また、データに対し著作権などの第三者の権利が及ぶことによりデータの利用が制限されることもありうる（なお、著作権については、第三者の著作物であっても学習等の情報解析の用に供する目的で利用する場合には利用しうること（著作権法 30 条の 4）などにも留意して検討する必要がある。）。

加えて、データに個人情報が含まれる場合には個人情報保護法に従った取り扱いが求められる。営業秘密や限定提供データの保護を図った不正競争防止法などの法令の遵守も求められる。

さらに、プライバシーへの配慮、倫理的な問題への対応等によりデータの利用が制限を受けることもあります。

データの利用にあたっては、これらの点に適切に対応する必要がある。

2.1.3 Model Robustness

Model Robustness では、モデルの精度と頑健性、デグレードなどについて考慮する。Model Robustness については、Data Integrity と同様に統計学や機械学習の分野で固有技術として盛んに議論されているので、品質保証についてもそれらを踏襲すればよいだろう。

まず、正答率や適合率、再現率、F 値といった精度と、汎化性能を充分に考慮する必要がある。また AUROC (Area Under Receiver Operating Characteristic : ROC 曲線における AUC) といったモデルのよさを示す指標を考慮する必要がある。

精度や汎化性能は、学習の度に適切な頻度で検討する必要がある。その際には、学習が適切に進行したか、局所最適に陥っていないかなども考慮する必要がある。

学習の進行に応じて、適切なアルゴリズムかどうか、適切なハイパーパラメータであるかどうかなどの考慮を行う必要がある。

モデルの検証という視点では、交差検証を充分に行ったか、ノイズに対して頑健なモデルであるか、充分に多様なデータでモデルの検証を行ったか、などを考慮する必要がある。数理的な多様性だけでなく、意味的な多様性や社会的・文化的な多様性についても考慮する必要がある。

また学習の進行に応じて、デグレードへの対処について検討する必要がある。ここでデグレードとは、ある学習を行う前には正しく判別できていたデータが学習後に誤判別を起こすといった現象を指す。デグレードが許容可能な範囲なのか、デグレードの影響範囲をきちんと把握できているか、

などを考慮する必要がある。許容できないデグレードに対して学習を行うことで対処する場合には、汎化性能の低下などに充分注意する必要がある。デグレードをきちんと考慮するために、学習が再現可能である必要がある。また学習時のふるまいに対して運用時のふるまいが齟齬を起こしていないかを検討する必要もある。

学習の過程において、また中長期間にわたる運用の過程において、モデルが陳腐化しないかどうか、既に陳腐化していないかどうかかも考慮する必要がある。また実際のデータに対する予測品質が劣化していくことがないかどうかかも考慮する必要がある。陳腐化や劣化について、データが原因なのか、ハイパーパラメータやアルゴリズムが原因なのか、システムの設計やサービスの考え方方が原因なのか、なども合わせて検討し対処する必要がある。

またこうした検討を行う場合に、目標となるメトリクスが顧客満足度など実際のサービスにおけるメトリクスのように、実際には直接計測できないこともある。その場合には計測できるメトリクスを代用特性として目標値化することになるが、目標メトリクスと計測メトリクスの関連が妥当かどうかなどを考慮する必要がある。

2.1.4 System Quality

System Quality では、AI プロダクト全体の品質の確保について考慮する。System Quality には、AI コンポーネントを一つの特殊なコンポーネントと捉えることによって、従来の演繹的開発の品質保証のノウハウが活用できる可能性がある。

まず、システム全体として価値を適切に提供できているかどうかを検討する必要がある。システムの価値が何を意味するのかはシステムやドメイン、ビジネスモデルに依存するが、AI プロダクトの場合にはまだ「とりあえずやってみる」という事例も散見されるため、自分たちが AI プロダクトの価値をどのように捉えているかも含めて反復的に検討する必要がある。

そしてその価値が適切に提供されていることを、システム全体として、および意味のある単位で評価したかを考慮する必要がある。AI プロダクトは全体として演繹的開発と帰納的開発の混合となるので、前者について分割統治が可能となるが後者については困難となり、その見極めが重要となる。

品質保証では一般に、致命的な品質低下やその影響、両者を含めた事象全体を品質事故と呼ぶことがある。AI プロダクトにおいても、発生しうる品質事故の致命度が許容できる程度に抑えられるかどうかを検討する必要がある。「事故」という言葉の響きから身体や生命への危害のみを企図するようにも思えるが、ドメインによって品質事故は経済的ダメージや社会・環境への影響、不快感、魅力の低さ、意味の無さ、反倫理性など様々な企図が考えられる。

AI プロダクトの品質事故については、誤判別といった機能的な品質事故だけでなく、性能低下やユーザビリティの悪化といったシステム全体のふるまいが劣化していかないか、についても考慮する必要がある。

品質事故を考慮する際には、トリガーとなる事象についても充分検討する必要がある。トリガー

となる事象を網羅的に検討する必要があるし、トリガーとなる事象の発生頻度はそれぞれどの程度かを検討する必要もある。トリガーとなる事象は歩行者の急な飛び出しのように AI プロダクトの外部で発生することもあるし、バグのように AI プロダクトの内部で発生することもある。利用環境を統制することによって、発生頻度を抑えられることもある。その場合、システムや環境、ユーザー、使い方、ドメイン、ビジネスモデルなどによって環境統制性が種別できる。例えばある機能が品質事故を起こすトリガー事象であるとする。警告文や免責条項などによって開発側が意図的にトリガー事象を引き起こすユーザーの利用を制限できる場合（意図）と、トリガー事象が偶発的に発生するため開発側が意図的に制限できない場合（偶発）、セキュリティ攻撃が予想されるためどんなに発生確率の小さいトリガー事象でも無視出来ない場合（攻撃）の 3 種は、品質保証の程度や方法が異なるため識別しておかなくてはならない。

トリガー事象から品質事故に到達する程度や被害を低減させる程度を考慮する必要もある。例えば、防護機構や安全機能、耐攻撃性の有無や数、それらの良し悪しなどについて検討する必要がある。ただし防護機構などが複雑になりすぎるとシステム全体の品質が低下し、トリガー事象が増加する、品質事故への到達度が上がる、品質事故の被害が大きくなる、といった弊害が発生する所以があるので充分に注意が必要である。またシステム全体としては、回避性や制御性を高めることで品質事故への到達度や被害を低めたり、自己修復性を持たせたりする場合などもあるので、そうした点についても考慮が必要となる。

AI プロダクトの場合には、AI コンポーネント同士、もしくは非 AI コンポーネントとの構造も考慮する必要がある。AI コンポーネントのふるまいが局所化される設計であったり非 AI コンポーネントや人間の判断でオーバーライドされる設計であったりといったように、AI の寄与度が適切に抑えられている設計かどうかも考慮すべきである。AI の寄与度はまた、AI コンポーネントと非 AI コンポーネントの両方において変更があった場合に迅速かつ適切に反映できるか、不具合があった場合に影響を充分低く抑えられるか、といった考慮事項も含む。

また品質保証を行う際には、その品質保証活動で本当に品質が保証できるという確信をステークホルダーが持ちうるかどうかも重要になる。こうした保証性や説明可能性、納得性の確保は、AI プロダクトの場合はそれほど容易ではない。演繹型開発のようにプロセスやメトリクスによる確保は意味を持たない。探索的開発や FEET のため、人手でドキュメントを作成し整備することは現実的でない。とはいえ学習の過程や FEET の結果の出力を自動化しても、膨大な出力を頻繁に検討することも現実的ではない。eXplainableAI の技術はいまだ端緒である。したがって保証性や説明可能性、納得性の確保は、開発者やチームの納得感の共感のように、品質保証の技術者や部門における納得感の共感の確保が極めて重要となる。自分たちが行っている品質保証活動はどのような意味があり、どのように技術的に品質向上に寄与し、どのように開発者やチームに納得感の共感を増やしてもらい、彼らがムダな作業や意味の無い作業だと思わないよう尽力しているかを、品質保証の担当者や組織自身が常に確信を持ち続けなくてはならない。品質保証の担当者や組織は、管理屋ではなく技術者であるという意識を持たなくてはならない。

なお、AI プロダクトが第三者の著作権および特許権等の知的財産権を侵害していないか、オープンソースソフトウェアを利用する場合に当該ソフトウェアに課された利用条件に違反していないか、AI プロダクトの利用が法令に違反していないか、といった観点からの検討も必要である。

また、AI が何らかの機器に組み込まれて用いられるような場合には、機器の品質に問題が生じた場合における、機器の製造者と AI の開発者との間での責任の分担という観点も重要になる。

すなわち、AI が何らかの機器に組み込まれた場合には、AI プロダクトの品質は、当該機器の製造物責任として現れることがある（本ガイドラインでも扱う自動運転車などがその典型例である。）。AI のモデルそれ自体は有体物ではないため、製造物責任の対象となる「動産」（製造物責任法 2 条 1 項）に該当しないものの、機器に組み込まれた場合には、当該機器に製造物責任法の「欠陥」（同条 2 項）の問題として、AI のモデルの品質も問題になりうるからである。もっとも、基本的には、製造物責任を負う「製造業者」（同条 3 項 1 号）とは、機器自体の製造者であり、AI のモデルの開発者ではないと考えられている。したがって、機器に欠陥があった場合に被害者から直接的に損害賠償を請求されるのは AI のモデルの開発者ではなく機器の製造者であることが多いと思われる。ただし、被害者から責任を追及された機器の製造者は、欠陥の原因が AI のモデルにあると考えた場合には、AI の開発者に対し、開発契約に基づく責任を追及するであろう。このように、機器の欠陥により生じた損害は、最終的には、機器の製造者と AI の開発者のどちらが損害を負担するか、という問題になりうるため、責任の分担を契約で定めておくことが望ましい場合もあることに留意すべきである。

2.1.5 Process Agility

Process Agility では、プロセスの機動性について考慮する。AI プロダクトの品質を保証するためには、納得感を共感した開発者や開発チームが自動化された開発環境を駆使して臨機応変に探索的開発を進めていく必要がある。したがって、開発が臨機応変か、充分自動化されているか、開発者や開発チームは納得感を共感しているか、について考慮する必要がある。

AI プロダクトの開発を臨機応変に進めるためには、データ収集の速度とスケーラビリティが充分である必要がある。同時に、充分短い反復単位での反復型開発や、モデルやシステムの品質向上の周期が充分短い必要がある。また運用状況の継続的なフィードバックも頻繁に行われる必要がある。

それによって、モデルやシステムがよりよくなっていく見込みがあるかについて検討する必要がある。学習の進行だけでなく、新しい特徴量を迅速に追加したりモデルを迅速に改善したりすることができるかについて考慮する必要がある。そのためには、学習や推論のデバッグを行う迅速に手段や環境、仕組みを保有しておく必要がある。

AI プロダクトは非線形性によって思わぬ品質事故が発生する可能性があるが、そのような場合にリリースロールバック（切り戻し：リリースをなかったことにして直前やそれ以前のバージョンに戻して再リリースすること）を簡便で迅速に行える必要がある。同様に、段階的リリースやカナリアリリースの頻度や程度は適切であるか、各リリース時にモデルやシステム全体の評価やテストを

行っているか、についても考慮する必要がある。

探索的開発や学習時・リリース時の FEET を行うためには、開発・探索・検証・リリースなどを自動化しておくことが必要となる。その際には、データ、モデル、環境、ソースコード、出力などについて適切に構成管理が行われている必要がある。

AI プロダクトは開発技術も品質保証技術もまだ発展途上であるため、開発者や開発チームが品質に寄与する割合が大きい。開発者や開発チームがプロダクトのふるまいや開発の進め方などに技術的に充分納得し共感しているか、顧客も含めて充分納得し共感しているか、は極めて重要になる。自分たちがいま何をどう作っているか、筋のよい開発や探索をしているか、何かが起こったら何をどこまで戻せばよいか、どのようなリスクがあるか、そうしたリスクにどう対処すればよいかが概ね分かっているか、自分たちが分からぬことをどこまで分かっているか、などを納得して共感する必要がある。とはいっても AI プロダクトはその特色のため、あらゆるリスクとそれらに対する対処を全て完璧に想定することは極めて困難であるため、分かることと分からぬことの経験的なバランス感覚も必要となる。

また開発チームは適切な能力を持った人材を備えているかどうかは、納得感の信頼度を高めるために必要である。データサイエンスや機械学習の専門技術と、ソフトウェア開発の専門技術、ドメイン技術のそれぞれについて、きちんと技術力の高い人材が必要となる。その際には、座学で取得できる資格や認定制度のみに頼らず、実務による経験も踏まえて評価する必要がある。

AI プロダクトの開発においては、探索的開発を進めていきながら開発者や開発チーム自身がさまざまな「学習」、すなわち経験や試行錯誤を洞察の獲得や技術の向上に反映させることが必要となる。技術的な反映策だけでなく、振り返りといった人間的な反映策も重要となる。

2.1.6 Customer Expectation

Customer Expectation では、よい顧客との関係性について考慮する。AI プロダクトの品質を保証するためには、良くも悪くも顧客の期待が高いかどうかが重要となる。良い意味で顧客の期待が高いと品質保証をしっかりやる必要があるし、悪い意味で顧客の期待が高いと AI プロダクトの特性についての理解が乏しい顧客によって品質保証が困難になるリスクに対処することになる。

まず、良い意味で顧客の期待が高い場合には、品質保証をしっかり行う必要がある。無償のホビー型プロダクトなど品質事故が起きても顧客に迷惑をかけないような AI プロダクトよりも、自動運転や金融取引など品質事故が起きると顧客の身体や財産に損害を与える AI プロダクトの方が、顧客の期待が高く、品質保証もそれだけしっかり行わなくてはならない。これは帰納型開発であっても演绎型開発であっても変わらない。

しかし悪い意味で顧客の期待が高い場合、すなわち AI プロダクトの特性についての理解が乏しい顧客の場合には、品質保証が困難になるリスクを抱える可能性がある。こうした顧客は「自分が特に何もしなくとも AI プロダクトや開発組織が常にようしく完璧にやってくれる」と考えている場合

がある。もちろん、これは誤った考え方である。品質保証の技術者やチーム、組織は、開発や営業と共に、AI プロダクトに関する顧客の理解を深めるような活動を行うことで、顧客の期待を適切に制御しなくてはならない。

AI プロダクトは確率的に動作する、と言われている。これは少し曖昧な表現である。AI プロダクトはコンピュータプログラムである以上、全く同じ状態の AI プロダクトに全く同じ入力を与えると全く同じ出力を返す。この意味では AI プロダクトは確定的に動作する。しかし想定される入力群全体に対しては、精度が 100% にならないため確率的に動作するようと思える。そして非線形にふるまうため、人間にとて同じと思える入力なのに異なる出力をすることがあるし、学習をすると思いもよらないところまでふるまいが変化することがあり、確率的に動作するようと思える。また同じデータで同じアルゴリズム、(初期値を除く) 同じハイパーパラメータであっても異なるモデルができるアルゴリズムがあるため、確率的に動作するようと思える。顧客によっては演繹的開発において「不具合が無いこと」を要求する習慣があることがあり、こうした AI プロダクトの確率的動作を許容しない場合がある。これは過学習につながり品質を低下させたり、ムダな作業を発生させ開発を停滞させたりする可能性がある。顧客がリスクや副作用を理解していなかったり、許容しなかったりする場合も同様である。

AI プロダクトは探索を伴うため、いきなり実運用レベルの開発を行うことは稀である。そのため PoC (Proof of Concept : 概念検証) やβリリースを行うことが多く、それに合わせて品質保証も行う必要がある。しかし PoC やβリリースといった開発段階を理解していない顧客の場合、PoC 段階にも関わらず実運用段階で問題になるようなリスクの指摘をして解決を求めることが発生することがある。これもまた品質の低下や開発の停滞を招く可能性がある。

顧客によってはデータの質や量に関する認識が甘いことがある。自社がどのようなデータをどれくらい保有しており、社外からどのようにデータを獲得できるのかを顧客が必ず理解しているとは限らない。開発前や契約前に、充分な認識をしてもらう必要がある。

AI プロダクトであっても演繹的開発であっても、何を目的として、どのようなデータを用いて、どのような精度で、どのような出力を行いたいのかを明確にする必要がある。時に顧客は「人間並み」という要求を行うことがある。その場合に、精度の目標として明示され合意されたものが、顧客が暗黙的に意図しているものと同じかどうかはきちんと検討し確認する必要がある。例えばある障害物を避ける精度が明示的な目標であったはずなのに、障害物の鼻先をかすめるような避け方を人間は行わない、というような全体的なふるまいを顧客が意図していることがある。またトロッコ問題のような倫理的問題など、人間の経験やセンス、数式やルールでも判定の難しいものは AI プロダクトでも基本的には判定できない。「人間並み」という要求が顧客から発せられた場合には、その意味について充分議論して合意するか、PoC で探索しながら明確にしていくという過程を理解してもらう必要がある。

ドメインによっては、AI プロダクトの利用が第三者のプライバシー等を侵害していないか、倫理的に問題がないか、といった観点からの検討が必要とされる。同種のシステムや仕組みがまだ存在

しない場合、社会的な受容が必要な場合もある。AI プロダクトは先進的な役割を果たすことがあるため、顧客側がこうした点を軽んじている可能性がある。

また顧客によっては、確率的動作や機械学習モデルの構造の複雑さによる事実上のブラックボックス性を理解せず、開発中に発生した誤判別や不具合、リスクについて合理的で詳細な説明、外挿や予測を求める場合がある。これは水掛け論となり開発が停滞する可能性がある。

こうした AI プロダクトの特性についての理解が乏しい顧客には、顧客と開発者・チームの間で納得感を共感する風土や雰囲気、仕事の進め方が不足していることが多い。同様に、顧客の担当者やチームで意志決定できる権限が少なく範囲が狭いことが多い。したがって品質保証としても、納得感を共感する風土や雰囲気、仕事の進め方になるよう、また適切な意志決定の権限や範囲にできるよう尽力する必要がある。これは演繹的開発に慣れた品質保証では業務範囲外だと考えるかもしれない。しかし AI プロダクトの品質保証の技術者やチームは、演繹的開発で矮小化された範囲を飛び越えて、品質を保証するために必要なあらゆることを、組織の壁を乗り越えてあらゆるステークホルダーと、納得感の共感を行いながら進めていかなくてはならないのである。

2.2 AI プロダクトの品質保証の分類軸ごとのチェックリスト

2.2.1 Data Integrity

- (a) 学習データの量の十分性
 - (a.i) 想定する学習手法の適用前提や統計的観点から十分な量のデータがあるか.
 - (a.ii) 想定する要求・適用環境において、希少な状況や分類クラスの偏りがある場合であっても、それらに対して十分な量のデータがあるか.
 - (a.iii) データ量が少ない場合、「かさ増し」(人工的なデータ生成など) で補完が可能か.
- (b) 学習データの妥当性
 - (b.i) 想定する要求・適用環境に意味の観点から対応した適切なデータとなっているか.
 - (b.ii) 要求・適用環境の想定にそぐわないデータが入っていないか.
 - (b.iii) 人工的に作成・加工したデータについても、要求・適用環境を適切に表現しているといえるか.
 - (b.iv) データの収集等の費用対効果の観点からも適切であるか.
- (c) 学習データの要件適合性
 - (c.i) データに関するステークホルダーの要求事項を満たしているか.
 - (c.ii) データが満たすべき不变条件や整合性条件、学習対象となる判断の公平性、個人情報の有無など、データに対する制約を満たしているか.
- (d) 学習データの適正性

- (d.i) 潜在的なバイアスや汚染の可能性について、多様なステークホルダーや社会への影響の観点から検討し、データが適切であることを確認したか。
- (e) 学習データの複雑性
 - (e.i) 学習させたい推論機能に対して、必要以上の情報量や傾向を含む複雑なデータとなっていないか。
 - (e.ii) データを単純化しすぎて、必要な情報が入っていないことはないか。
- (f) 学習データの性質の考慮
 - (f.i) 想定する学習手法の適用前提となるようなデータの性質（多重共線性など）は適切に考慮されているか。
- (g) 学習データの値域の妥当性
 - (g.i) データに含まれている値は、対象ドメインの知識などと照らし合わせて現実的に発生する妥当な値となっているか。
 - (g.ii) 外れ値と欠損値と判断した値は、真に現実的な値ではなく取り除くべきであることを確認したか。データを取り除くための前処理は適切であったか。
- (h) 学習データの法的適合性
 - (h.i) データの利用が契約や第三者の知的財産権により制限されないか、データの利用に法令上、倫理上の問題はないか、プライバシー等への配慮が必要ないか。
- (i) 検証用データの妥当性
 - (i.i) 学習用データと検証用データは独立しているか。
- (j) オンライン学習の影響の考慮
 - (j.i) インクリメンタルに追加や置き換え、削除されるデータについて、適切な運用機構・体制を設け、監視、制御や制限、検証を行っているか。
- (k) データ処理プログラムの妥当性
 - (k.i) データに対する前処理、作成・加工などの処理を行うアルゴリズムの特性や、そのライブラリやそれを呼び出すプログラムの不具合、誤った利用により、データの適切さが失われていないか。

2.2.2 Model Robustness

- (a) モデルの精度の充分性
 - (a.i) 正答率, 適合率, 再現率, F 値といった推論性能に関する評価指標の値は, 要求に對して十分か.
- (b) モデルの汎化性能の充分性
 - (b.i) 汎化性能は確保されているか.
- (c) モデルの評価の充分性
 - (c.i) (AUROC といった) 精度以外のモデルのよさを表す指標についても適切な指標を選定し充分に評価したか.
- (d) 学習過程の妥当性
 - (d.i) 学習は適切に進行したか.
 - (d.ii) 学習結果が局所最適に陥っていないか.
- (e) モデル構造の妥当性
 - (e.i) 適切なアルゴリズムやハイパーパラメータかどうかの検討は行ったか.
- (f) モデルの検証の妥当性
 - (f.i) 十分に交差検証などを行ったか.
- (g) モデルの頑健性
 - (g.i) ノイズに対して頑健か.
- (h) 検証用データの多様性
 - (h.i) 数理的多様性, 意味的多様性, 社会的文化的多様性などを考慮し, 十分に多様なデータで検証を行ったか.
- (i) モデル更新に対する検証の充分性
 - (i.i) モデルを更新する場合, 以前の振る舞いとの変化について把握しているか, それが許容可能であることを確認しているか.
 - (i.ii) 特に自動でのモデル更新・配備を行う場合, 自動化された検査内容は十分であるか.
- (j) モデルの陳腐化への考慮
 - (j.i) 運用時における傾向の変化により, モデルの性能, 妥当性, 有用性が低下する可能性を検討し, それに対するモデルの頑健性確保, 運用における監視などの対策をとっているか.
- (k) プログラムとしてのモデルの適切性
 - (k.i) 学習アルゴリズムの特性や, そのライブラリやそれを呼び出すプログラムの不具合や誤った利用により, 不適切なモデルとなっていないか.

2.2.3 System Quality

- (a) システムによる提供価値の適切性
 - (a.i) システム全体により価値は適切に提供されているか、提供価値を計測できているか。
 - (a.ii) 価値の計測が難しい場合、計測できる代替メトリクスとの関連は妥当か。
- (b) AI のシステムへの影響
 - (b.i) AI の導入や変更がシステム全体のふるまいや性能などの品質に悪影響を与えていないか。
- (c) システム評価単位の妥当性
 - (c.i) システムを全体として、および意味のあるサブシステム単位で評価を行ったか。
- (d) 事故による影響の抑制
 - (d.i) 発生しうる品質事故の致命度は、許容できる程度に低く抑えられているか。
 - (d.ii) 品質事故を引き起こしうる事象の発生頻度は低いと見積もることができるか。
 - (d.iii) 事象の発生頻度、事象の網羅性、事象に影響を与える環境の制御可能性に関する検討は十分か。
- (e) 事故発生の回避性
 - (e.i) システムの事故到達度は十分に抑制しているか。
 - (e.ii) 十分な安全機能や耐攻撃性を提供しているか。
- (f) AI の影響度の抑制
 - (f.i) システムの様々な要素の AI への設計上の依存度や結合度を抑えられているか。
 - (f.ii) システムが依存する他の（AI の、もしくは非 AI の）システムの変更の影響は、迅速かつ適切に反映できるか。
 - (f.iii) AI の不具合の影響を十分に低く抑えられる設計や設計変更が可能か。
- (g) ステークホルダーの納得性
 - (g.i) ステークホルダーに対する保証性、説明可能性、納得性は十分か。
- (h) AI システムの法的適合性
 - (h.i) AI プロダクトが第三者の知的財産権を侵害しないか、AI プロダクトの利用が法令に違反しないか。
- (i) システムの品質低下への考慮
 - (i.i) 繙続的な運用に伴うシステムの性能などの品質が低下する可能性を検討したか。
 - (i.ii) 運用中のシステム品質低下を検知する仕組みを検討したか。

2.2.4 Process Agility

- (a) データ収集の迅速性
 - (a.i) データ収集の速度とスケーラビリティは十分か.
- (b) 開発の迅速性
 - (b.i) 十分に短い反復単位で反復型開発を行っているか.
 - (b.ii) モデル・システムの品質向上の周期は十分に短いか.
 - (b.iii) 運用状況の継続的なフィードバックは頻繁に行っているか.
- (c) 問題解析の迅速性
 - (c.i) 問題が発生した時にその原因を解析するために、問題発生時の状況を記録し取得できる仕組みを備えているか.
 - (c.ii) 取得した問題発生時の状況をもとに事象を再現できるか.
- (d) 回復の迅速性
 - (d.i) リリースロールバックは簡便で迅速に行えるか.
- (e) 改善の迅速性
 - (e.i) 新しい特徴量を迅速に追加したりモデルを迅速に改善したりできるなど、よりよくなっていく見込みはあるか.
- (f) リリースの迅速性
 - (f.i) 段階的リリースやカナリアリリースの度合は適切か.
 - (f.ii) リリース直前にシステム全体やモデルの評価を行っているか.
- (g) 自動化の十分性
 - (g.i) 開発・探索・検証・リリースなどの自動化は十分か.
- (h) 構成管理の適切性
 - (h.i) データ、モデル、環境、コード、出力などの構成管理が適切に行われているか.
- (i) 開発チームの適性
 - (i.i) 開発者やチームは技術的に十分に納得し共感しているか.
 - (i.ii) 開発チームは適切な能力を持った人財を備えているか.
- (j) 技術進化の迅速性
 - (j.i) 経験を技術に反映させられているか.
- (k) ステークホルダーの納得性
 - (k.i) 開発チーム外のステークホルダーは十分納得しているか.

2.2.5 Customer Expectation

- (a) ステークホルダーの期待度
 - (a.i) 顧客の期待は高いか.
 - (a.ii) 狙っているのが「人間並み」か.
- (b) ステークホルダーの技術理解度
 - (b.i) 顧客は確率的動作という考え方を受容していないか.
 - (b.ii) リスク・副作用を理解していないか, もしくは安易に受容して必要な対策を怠っていないか.
 - (b.iii) データの量や質に対する認識は甘いか.
 - (b.iv) “合理的”説明を求める傾向や, “外挿”や“予測”をしたがる傾向, “原因”や“責任(者)”を求めたがる傾向はあるか.
- (c) 運用に対する期待度
 - (c.i) 繙続的実運用にどのくらい近いか.
- (d) 標準適合性の必要度
 - (d.i) AI プロダクトの利用に法令上, 倫理上の問題があるか, 第三者のプライバシー等への配慮を必要とするか, AI プロダクトの利用について社会が否定的か.
- (e) ステークホルダーとの関係性
 - (e.i) 納得感を共感する風土や雰囲気, 仕事の進め方は少ないか.
 - (e.ii) 顧客担当者・チームで意思決定できる権限や範囲は少ない・狭いか.

2.3 AI プロダクトの品質保証の構築・評価

2.3.1 バランスに着目した構築・評価

AI プロダクトの品質保証の構築・評価を行う際には、2.1 節や 2.2 節で挙げた 5 つの軸のバランスが取れている必要がある。バランスがよいとは、顧客の期待に沿っていると保証できることが期待できる状況を意味する。それは二つの条件から構成される。

一つは、Data Integrity/Model Robustness/System Quality/Process Agility に不足がないという条件である。それらのどれか一つが欠けていても品質が確保できたとは言えない。もう一つは、Data Integrity/Model Robustness/System Quality/Process Agility に対して Customer Expectation が適切であるという条件である。顧客の期待を適切に把握し、可能ならば期待を適切な程度に收め、Data Integrity/Model Robustness/System Quality/Process Agility が総合的に期待を満たすようにできている場合を意味する。

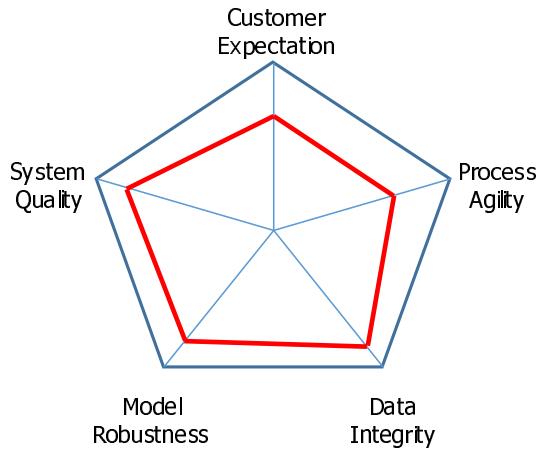


図 2.1 バランスのよい品質保証のイメージ

5つの軸のバランスが悪いとは、顧客の期待に対して Data Integrity/Model Robustness/System Quality/Process Agility が不足している状況を意味する。すなわち、Data Integrity/Model Robustness/System Quality/Process Agility のいずれかが不足している状態か、顧客の期待が高すぎる状況である。

顧客の期待を適切に収める行動を期待値コントロールと呼ぶことが多いが、短期的で個別的な期待値コントロールと中長期的で社会的な期待値コントロールの両方が必要なことがある。産業向けなど特定顧客向けの AI プロダクトの場合には、前者の施策が重要となる。すなわち、顧客のキーパーソンを特定し、確率的動作やデータの価値といった AI プロダクトの特質を理解させ、顧客組織の真の目的やニーズを明確化し具体化することで、顧客が過度の期待をしないことと顧客事業における戦略的重要性を低めないことの両方を実現する必要がある。同時に、顧客組織内においてきちんと納得感を共有しながら仕事を進める雰囲気や、顧客担当者やチームの意志決定の自由度を適切にすることも働きかける必要がある。また一般顧客向けの AI プロダクトの場合には、後者の施策として UI の改善や動画による理解の促進、SNS やメディア、イベントなどを通じた顧客との対話だけではなく、専門家や自治体・政府機関などの連携による情報発信を通じた世論形成などが考えられる。

図 2.1 に、バランスがよい品質保証の例を示す。5つの軸に対しそれぞれの状態をプロットができる五角形が正五角形に近いほど、直感的にもバランスがよいと認識できる。図 2.2 のように正五角形から崩れた品質保証の場合には、直感的にもバランスが悪いと認識できる。またバランスは本質的に Customer Expectation に対する各軸の関係であるので、図 2.3 のように 4 つの軸で四角形を構成し、Customer Expectation を表す円との相対関係を示す記法も考えられる。

本ガイドラインでは、各軸の値や線形性、軸間の相対的関係については提示しない。すなわち 2.2 節でのチェックの数がそのままバランスを表すわけではない。したがってこのバランスについては、各軸について開発、品質保証、顧客の 3 者を中心として充分に議論し納得感を共感しておく必要が

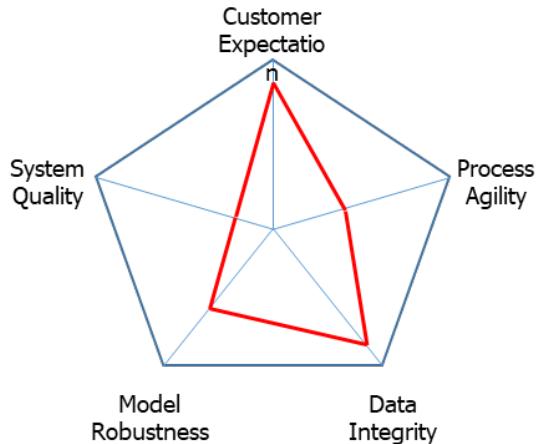


図 2.2 バランスの悪い品質保証のイメージ

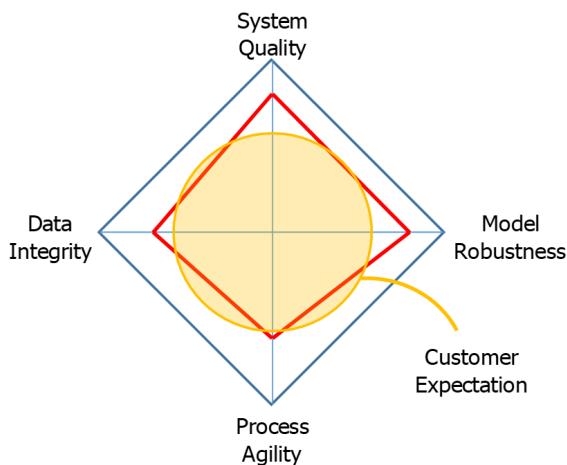


図 2.3 四角形と円による品質保証のバランスのイメージ

ある。

2.3.2 開発段階に着目した構築・評価

AI プロダクトの品質保証の構築・評価を行う際には、開発の段階に合わせて変えていく必要がある。PoC 段階ではそれほどしっかりした品質保証は必要ないかもしれないが、βリリース段階や継続的実運用段階になるにつれてしっかりした品質保証が必要になる。ただし開発段階に関わらず、バランスは取れている必要はある。図 2.4 に開発段階に合わせた品質保証の大きさの変化のイメージを示す。

本ガイドラインでは、開発段階ごとに推奨される各軸の値については提示しない。したがって、ど

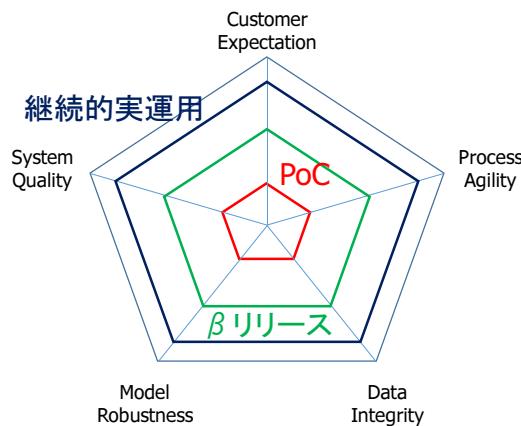


図 2.4 開発段階に合わせた品質保証の大きさの変化のイメージ

の開発段階でどこまで品質保証を行うかについては、各軸について開発、品質保証、顧客の 3 者を中心として充分に議論し納得感を共感しておく必要がある。

2.3.3 余力と過剰品質

AI プロダクトの開発や品質保証の経験が豊富になると、Data Integrity/Model Robustness/System Quality/Process Agility が顧客の期待を上回る状況が発生する。例えば図 2.5 の赤線で囲まれたオレンジ色の領域の示すような状況である。

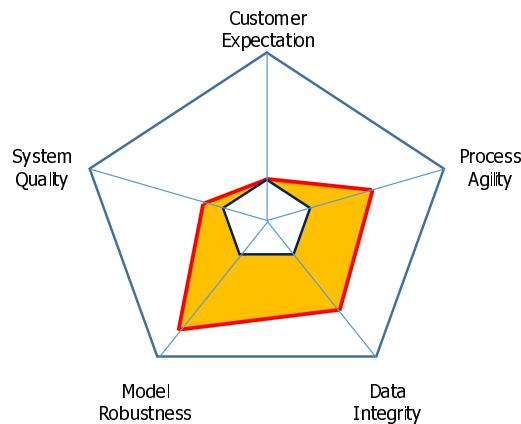


図 2.5 品質保証の「余力」のイメージ

この状況は誤解を生みやすいが、「余力」と捉えるのが適切である。すなわち、将来的に開発段階が進んだり運用範囲が広がったりすると顧客の期待が高めるため、その時に備えた活動を事前に行っていると解釈すべきである。また、一つのプロジェクトに余力があると、他のプロジェクトの

品質向上の基盤にもなりうる。したがって、むしろ積極的に余力を増やす方向に戦略的な技術投資を行う方が持続的かつ全体的に最適となる可能性が高い。

この状況を「過剰品質」と呼んではならない。過剰品質とはむしろ、プロジェクトや QA が顧客の期待やビジネスの方向性、自分たちの開発の状況や技術力を見失い、自分たちが何をやっているのかを分からずにただムダな手数だけを増やしている状況を指す。もし高い品質を達成するためにムダな手数が増えてしまうのであれば、Process Agility が低いために手数を減らせなかったり、形式的な会議や文書が低かったりする場合が考えられる。また Data Integrity や Model Robustness が低いために手数をかけても精度や汎化性能が頭打ちになる場合が考えられる。もしくは System Quality を達成する際に技術的本質を軽視するため、総花的で包括的だが効果の薄いプロセス的施策を進めている場合が考えられる。こうした場合は過剰品質などと誤ったレッテルを貼らず、適切に投資を行って改善を進めていくべきである。さもないと、誤ったレッテルのせいで余力を高める活動への反発が大きくなり投資が抑制され、品質の低下や開発の停滞が発生してしまうだろう。

3. 技術力タログ

機械学習は、教師あり学習、教師なし学習、強化学習に分類される。教師あり学習においては、訓練データとして、入力値とそれに対し正解となる出力値の対を多数与えることで、入力に対して出力を求めるようなモデルを得る、出力がクラス群への振分けとなるタスクを分類、出力が数値となるタスクを回帰という。これに対して教師なし学習においては、正解を定義することなく、データから規則性や判断基準などを抽出する。データをグループ分けするクラスタリングや、データから相関ルールを抽出するアソシエーション分析が代表的である。強化学習においては、ゲームプレイヤや探索ロボットの構築などにおいて、状況に応じてどのような行動をとるべきかを訓練を通して学習する。これらのうち、教師あり学習は特にその性能向上やタスクの明確さから実応用が特に進んでおり、本章では教師あり学習の品質を中心に扱う。

機械学習を用いたシステム開発では、訓練データ（学習データや教師データともいう）を入力として、学習アルゴリズムを実装したプログラム（以後学習プログラム）を実行することにより、識別や予測、制御などを行うソフトウェア部品を得る。この部品は、入出力の関係性を表したものであり、モデルと呼ぶ（本章における「モデル」という語はこの用法に従う）。機械学習を用いたシステム開発では、訓練データ、学習アルゴリズムやそのパラメータ・オプション（訓練で値を決めるパラメータと区別するためハイパー・パラメータという）が設計の対象となり、プロダクトに埋め込まれ活用されるモデルはそれらから間接的に得られることになる。

機械学習、特に深層学習で得た複雑なモデルの振る舞いは、プログラマが演繹的に（一般的規則に基づき）書き下したものではなく、訓練データから帰納的に（個々の事例に基づき）生成したために、ブラックボックスであり以下のような特徴をもつ。

- 様々な入力の可能性すべてに対し、正解や期待値（テストオラクル）を明確に定義できない、あるいは定義することが人手の作業を要するなど高コストであることが多い。
- 機能は原則として不完全であり、正解を定義できたとしても、その正解を常に求めることはできず、性能に限界がある。その性能を事前に見積もることは困難であり、またできることとできないことの境界を明確に把握することはできない。
- 個々の入力に対し、どうしてその出力が得られたかを演繹的には説明できない。
- 訓練データを変えてモデルを再構築すると、その挙動が大きく変わることがあり、その変化の予測をすることは難しい。
- 入力の微量な変化（例えば人にはわからない程度の画像の変化）により、出力が大きく変わることがある。これは敵対的サンプル（Adversarial Example）として知られる〔Goodfellow 2015〕。

機械学習を用いた人工知能システムの品質においては、これらの特徴を考慮すべきである〔大場 2018〕〔石川 2018〕〔Breck 2016〕。

3.1 AI プロダクト固有の品質特性

機械学習を用いて得たモデルは、出力に対する正解が定義される場合でも 100% 正解を求めるとはできない。このため、既知のデータを用いて、それらに対しどれだけ正解や正解に近い値を求めることができたかを評価することが最も基本的な品質評価となる、そのための品質特性として正解率や精度等の指標で表される性能を考えることが多い。機械学習分野では性能という語が広く使われるため本章でも性能という用語を使うが、例えば ISO25010 品質モデル (SQuaRE) では正確性に該当する。また、機械学習モデルや、機械学習を用いたシステムに固有の品質特性としては、**頑健性 (Robustness)** や**説明可能性・解釈性 (Explainability/Interpretability)** がある。そのほか、対象アプリケーションによっては、判断の公平性 (fairness) など、文化的・社会的な要求の反映という観点からの品質を考慮する必要がある。

以下では、AI プロダクト固有の側面について論じる。AI プロダクト固有の品質特性全体像については、欧州の倫理ガイドライン〔EC 2019〕や、ISO25010 品質モデルの拡張を論じた〔Kuwajima 2019〕においても示されている。

3.1.1 教師あり学習のモデルに対する性能指標

(Model Robustness に関する)

教師あり学習のモデルに対する基本的な評価としては、既存のデータに対してどれだけ正解を求めることができるかという性能の評価を行う。最も単純な指標である**正解率 (Accuracy)** は「正解数」と「問題総数」の比である。正解率等の性能指標は、テスト対象となるデータに対して相対的に決まるため、テストデータの選定が非常に重要となる。想定する要求および運用環境を表現するようなテストデータが必要となる。

機械学習で得たモデルのうち、入力を特定のクラス群に振り分ける分類タスクを扱うものについては、情報検索分野にて用いられてきた品質指標が用いられる。正解率は、分類先のクラスに偏りがある場合には、分類性能を適切に表さない場合がある。例えば正常ケースが 99.9%、異常ケースが 0.1% 存在する場合、常に「正常」と答える（実質分類していない）モデルの正解率は 99.9%となってしまう。

よりモデルの特徴を把握できるように、誤検出と検出漏れを区別する。まず分類先のクラスが**陽性・陰性**の 2 種類である検出タスクを考える。検出されたもの（陽性）のうち、本来検出されるべきもの・検出されるべきでないもの（**真陽性・偽陽性**）、検出されなかったもの（陰性）のうち本来検出されるべきでないもの・検出されるべきもの（**真陰性・偽陰性**）それぞれの数を考える。これら 4

つの数を行列として書き出したものを混同行列（Confusion Matrix）という。また、品質指標として以下を用いる。

- **適合率・精度 (Precision)**。「真陽性の数」と「真陽性と偽陽性の総数」の比。検出したものすべてのうち、どれだけ正しいものが入っているかを表す（誤検出の少なさ）。
- **再現率 (Recall)**。「真陽性の数」と「真陽性と偽陰性の総数」の比。検出すべきものを、どれだけ検出できているかを表す（検出漏れの少なさ）。
- **F 値 (F-Measure)**。 $\frac{2}{\frac{1}{\text{適合率}} + \frac{1}{\text{再現率}}}$ で表され、適合率と再現率の双方がバランスよく高いかどうかを表す。

適合率と再現率はトレードオフの関係にある。検出基準が厳しいと、適合率は高く、再現率は低くなる。検出基準がゆるいと、適合率は低く、再現率は高くなる。多くの場合双方を高くすることは難しく、要求やユースケースを踏まえてどちらを優先するか判断することが多い。分類に用いるしきい値の設定などにより適合率と再現率のトレードオフがどう変わっていくかを見る ROC 曲線（Receiver Operating Characteristic Curve）やその際の総合性能を見る AUC（Area Under Curve）も用いられる。

分類先のクラスが 3 つ以上の場合は、同様に適合率等の指標を考える。例えば、クラスごとに適合率を求めたり、その平均をとったりする（マクロ平均と呼ばれる）。

機械学習で得たモデルのうち、数値を予測する回帰のタスクを扱うものについては、予測値と実測値の誤差をとらえる指標が用いられる。代表的なものとして、平均二乗誤差（RMSE : Root Mean Squared Error）や決定係数（Coefficient of Determination、R²）がある。

モデルは、与えた訓練データから抽出した規則性や判断基準を表したものであるが、運用時には、訓練データに含まれない値が入力として与えられることになる。このため、訓練データに限らない一般的なデータに対する性能（汎化性能）に留意する必要がある。基本的には訓練データ外のデータに対しては性能が落ちるため、一般化エラーを評価するともいう。

汎化性能が重要であるため、最終的な評価に用いる評価データは予め分離しておく。その傾向の分析や、特に訓練への利用は決して行わないようとする。また訓練時においても、訓練データを訓練用と評価用に分割し、その分割の仕方も変えながら、訓練に用いたものとは異なるデータで評価を行うことを繰り返すことにより、訓練の手続き自体の妥当性を汎化性能の観点から評価する（交差検証）。

モデルが訓練データ固有の規則性・判断基準、特に局所的なノイズやばらつきに特化しすぎており、他の入力データに対する性能が悪い場合、そのモデルは過学習（オーバーフィッティング、高バリアンス（High Variance）ともいう）の状態にあるという。逆に、モデルが（訓練データ内ですら）必要な規則性・判断基準を表現できていない場合、そのモデルは未学習（アンダーフィッティング、高バイアス（High Bias）ともいう）の状態にあるという。正解値と予測との偏差であるバイア

スと、予測の散らばり度合いであるバリアンスとの間のトレードオフを考えるということだとともうらえられる。これらについては、出力の関係性を表すモデルの形や学習アルゴリズム、ハイパーパラメータなどに起因して生じる。訓練データの増加に伴い訓練データに対する性能と評価データに対する性能とがどう変化するかなどを学習曲線を分析する。

3.1.2 データに対する評価

(Data Integrity に関する)

訓練データの品質は、それから導出するモデルの品質に影響を及ぼすため、内部品質ではあるものの非常に重要である。訓練データにおいて正解と分類づけされている出力が、実際には不正確である場合、あるいは本来の要求を踏まえると不適切である場合、その訓練データを用いて導出したモデルは、同様な不適切な出力を出してしまう可能性がある。また、訓練データ内の規則性（傾向、分布）と運用時のデータ規則性（傾向、分布）が異なる場合や、訓練データ内には現れない範囲の入力が運用時に与えられる場合なども、運用時の性能が低くなることが多い。このため、訓練データの品質として、正確さや、システム要求や運用環境の想定に対する妥当性および十分性が重要となる。

また、3.1.1 で述べた性能指標は、評価対象として用いられるデータに対して相対的となる。評価に用いるデータ（訓練データおよびテストデータ）についても、システム要求や運用環境の想定に対する妥当性および十分性が重要となる。

このほか一般的なデータの品質、例えば ISO/IEC 25012 および JIS X 25012 データ品質モデルにて定義される追跡可能性や移植性などのデータ品質特性についても考慮する必要がある。

3.1.3 頑健性

(Model Robustness に関する)

本章のはじまりにて述べた敵対的サンプルのように、入力における微少なノイズ等によりモデルの出力が変化することがある。これはノイズに対して頑健（ロバスト）ではないといえる。訓練データと異なる入力値に対して十分な性能がでない場合、入力領域の変化に対して頑健（ロバスト）ではないといえる。これらのように、何かの変化に対してもモデルが安定して性能を達成するという品質特性を、**頑健性（Robustness）** と呼ぶ。

3.1.4 公平性

(Data Integrity および Model Robustness に関する)

公平性（Fairness） は、システムの出力や振る舞いが、人種や民族、性別などの特性による差別、偏見、偏愛に相当するような不公平な偏りを示すことがない程度を表す [Mehrabi 2019]。公平性はその定義から社会的・文化的な観点を含んでおり、様々なステークホルダー、組織および社会の要

請に応じて定まるものであり、個々のシステムにおいて何が該当するのか注意深く検討する必要がある。

不公平性な出力や振る舞いは、機械学習技術の特性上、意識せずにシステムに埋め込まれてしまうことがある点に留意すべきである。単純には、モデルの学習を駆動する訓練データにおいてそもそも不公平な出力や振る舞いを示唆するような場合がある（性別差別を含む過去のデータを使ってしまう場合など）。より暗黙的な場合として、3.1.1 節で挙げたような性能指標を用いた場合、与えられたデータセット全体に対して性能がよいことをもって満足しがちである。しかし、データセット内に少数しか含まれない特定部分（特定の人種に属するデータなど）を抜き出すと、それらに対応する入力に対しては非常に性能が悪い、つまり人種等に応じて性能が変わる不公平さが生じていることがある。不公平な判断などを学習させたわけではなくとも、結果として不公平な出力や振る舞いが生じることになる。該当システムにおいて重要となる公平性について定義し、評価を行う必要がある。

公平性の具体的な定義としては、Demographic parity や Equalized odds がある。Demographic parityにおいては、性別や人種などのセンシティブな属性について、予測ラベルの分布や予測性能指標値が一致していることを目指す。例えば採用判断を行う AI であれば、男性の採用率と女性の採用率の等しさを確認する。後者においては、センシティブな属性以外について同様な属性値を持つような二つの入力に対して、同等な予測を行うことを目指す。例えば、性別以外が同一である採用者候補二名について、同様な採用率が期待されることを確認する。それぞれの評価基準について、データ内に偏りがあるかないかといった適用仮定や、手続きの公平性と結果の公平性の重視度合いといった差がある。どのようなときにどのような評価観点を用いるべきかについては、社会的な合意はとれておらず、応用事例とそのステークホルダーに応じ十分な議論が必要である。

3.1.5 説明可能性

(System Quality に関連)

説明可能性・解釈性 (Explainability/Interpretability) は、システムからの出力を用いる人間が、出力を得る際に用いられた判断基準（モデルが学習した規則性）について把握することができる程度を表す [Gunning 2016]。要求や振る舞いを明確に形式知として書き出せない場合でも機能を実現できることが機械学習の強みである。しかし、人間が出力を参考にして意思決定を行う場合など、アプリケーションによっては説明可能性・解釈性が必要となる。

3.1.6 機械学習を用いたシステム全体における品質

(System Quality に関連)

どれだけ工数をかけてもモデルの精度等は 100% にはなり得ず、またそれらを高くすることはシ

システム開発の根本的な目標ではない。また教師なし学習の場合は、正解を定めてそれと比較するような相対的な性能指標を用いることができない。このため、ビジネスやシステムの全体観点からの目標指標である **KPI (Key Performance Indicator)** を定義し、その観点から品質を論じる必要がある。KPI となるシステムの指標に関する評価を行ったり、精度等のモデルの性能指標と KPI となるシステムの指標との関連性を測定したりする。例えば、Web アプリケーションなどユーザーの満足度や行動促進等が KPI となる場合、A/B テスティングなどによってシステムの投入有無の比較を行い、投入によって KPI が向上することの仮説検定 (Hypothesis Testing) を行うなどしてシステム全体の品質を評価する。

3.2 AI プロダクトにおける品質管理

(Process Agility に関連)

本章はじめ述べた機械学習の特徴から、モデルの品質、およびモデルを含むシステム全体のマネジメントにおいては以下を留意する必要がある。

- 前もって担保できる品質（特に精度等の性能指標）を予測し合意することは難しく、実験を繰り返しながらどこまで品質を実現できるのかを探る試験的・探索的なプロセスを採ることになる。場合によっては、例えば要求やユースケースを調節して、「適合率が低くても再現率が高ければ受け入れる」といった判断が必要となる。試験的・探索的なプロセスについて、顧客をはじめとしたステークホルダーの理解・協力が必要となる。
- 訓練に用いたデータと運用時に入力されるデータの分布が異なる場合に、品質（特に精度等の性能指標）が劣化する場合がある。これは開発時の想定が不十分であった場合に生じるが、自動運転等の実世界を扱う場合に想定を尽くすことは困難である。さらに、顧客の行動や嗜好、実世界に存在する物体やそれをとらえるカメラ特性等は変化するため、もし開発時の想定が十分なされていたとしても、実行時の性能劣化があり得る。このため、実行時の監視機構を設け、性能劣化を検出するなどの対応が必要である。問題追及をより容易とするためには、結果としての性能だけでなく、入力値の範囲や分布など個々の要素やそれらの関係についての監視を行うようにする [Breck 2016]。
- オンライン学習を行う、つまり実行時のデータを訓練データとして継続的にモデルを更新し続ける場合、最新の分布に適応することができる。しかしこの場合、不適切な訓練データにより不適切なモデルが得られる可能性がある。継続的な学習（モデルの更新）を自動化する際に、訓練データの選別、性能評価やテスト等も併せて自動化する必要がある。
- 機械学習を用いたシステムにおいてプロダクトの一部となるのは生成されたモデルだが、得られたモデルは、訓練データ、学習アルゴリズムを実装したプログラムとそのハイパーパラメータに依存する。同じモデルを再構築したり、そのモデルを構築した状況を理解したりす

るために、再現性を実現するための記録を行うべきである。例えば、コマンドラインで学習アルゴリズムに関するオプションを設定した場合に、その情報が失われないように記録をする必要がある。

3.3 AI プロダクトの品質保証技術

3.1.1 で述べた性能観点以外の品質保証技術については、特にテスト技術については盛んな研究が行われている [Zhang 2020] もの、現在発展途上段階にあり、またアプリケーションへの依存性が高い。個々のツールについては新しいものが発表されていることもある点に留意いただきたい。

3.3.1 疑似オラクル

(Model Robustness および System Quality に関する)

機械学習で得たモデルやそれを含むシステムに対しては、様々な入力の可能性すべてに対し、正解や期待値（テストオラクル）を明確に定義できない、あるいは定義することが人手の作業を要するなど高コストであることが多い。このため、テストケースの数が限られがちである。多数・多様な入力を試すテストケースを設けるために、比較対象となる疑似オラクルを用意する。例えば、別の実装（Nバージョンプログラミング）や、古いバージョン、ルールベースの実装などとの比較を行う。必ずしも出力が完全に合致するわけではない場合でも、誤差・距離を定義し、それが大きいケースを調べることで誤りに気づいたり知見が得られたりすることがある。誤差・距離が大きくなるようなテストケースを、進化計算等により探索するサーチベースドテスティングを用いることもできる [Pei 2017]。

3.3.2 メタモルフィックテスティング

(Model Robustness および System Quality に関する)

モデルの精度等は 100% ではないため、出力に誤りがあるとしてもそれはコーディングの誤りなどの実装不具合を意味するわけではない。このため従来のように、出力が期待値とずれていたら不具合の存在を確信する、という形でのテスティングはできない。メタモルフィックテスティングにおいては、「入力に対してある一定の変化を与えると、出力の変化が理論上予想できる」という関係（メタモルフィック関係）を用いることで、正否判断が可能なテストを得る [Chen 2018]。そのメタモルフィック関係が本来成り立つと確信できるものであれば、その関係が成り立たないケースは実装不具合を示唆すると考えることができる。あるいは、期待していた関係が成り立たないことは、実装されたモデルに対する理解が誤っていたことを示すこともある。

メタモルフィックテスティングにおいては、「データ点の属性に一定値を加えるあるいは乗算する」「データ点を追加する・削除する」「データ点を入れ替える」といったパターンから、成立するで

あろうメタモルフィック関係を定め、それを用いたテスティングを行う。

メタモルフィックテスティングは、入力摂動に対するモデルの頑健性検査において用いられる [Tian 2018] ほか、学習アルゴリズムなど訓練パイプラインの検査にも用いることができる [Dwarakanath 2018]。

3.3.3 頑健性検査

(Model Robustness および System Quality に関する)

3.1.3 で述べた頑健性を評価するために、入力となるテストデータに変化を加えて出力が変化しないかを評価することがある。例えば入力が画像である場合、照度の変化や、雨や霧の画像合成による追加、一部欠損やゆがみなどを考える。サーチベースドテスティングにより、入力の変化により出力がより大きく変わるようなケースを探索するようなツールは多数研究されている [Tian 2018]。

3.3.4 ニューラルネットワークにおけるカバレッジ

(Model Robustness に関する)

従来のソフトウェアのホワイトボックステスティングにおいては、分岐カバレッジなどのカバレッジ指標を行い、実装されたプログラムに含まれる多様な状況がどれだけテストされたかを評価してきた。一方、学習アルゴリズムとしてニューラルネットワークに基づくもの（主に深層学習）を用いた場合、実装されたモデルの振る舞いは、論理的な条件分岐ではなく数値の大小によって変わる。このため、モデルを表すプログラムにおいて、分岐カバレッジ等を用いても少しのテストでカバレッジ値が 100% に到達してしまう。これに対し、ニューラルネットワーク内における様々な計算部品（ニューロン）の数値大小を用いて、テストの多様性に関する評価や多様なテストの生成を行うことが考えられている [Pei 2017] [Ma 2018]。ただし、必ずしも求められる品質と強い相関があるわけではないことも報告、議論されており [Harel-Canada 2020]、少なくとも盲目的に従来のプログラムに対するカバレッジ指標と同等に安易に解釈、運用すべきではない。

3.3.5 説明可能性・解釈性のための技術

(System Quality に関する)

3.1.4 で述べた説明可能性・解釈性に関する技術としては、例えば個々の出力に対する説明を行うものや、モデル全体に関する説明を行うものがある [Gunning 2016]。前者の場合は、入力データのうちどの部分がある出力の決定に影響を大きく与えたのかといった情報を提示する。後者の場合は、説明可能性が高い IF-THEN ルール形式や木構造の形式（決定木）などを用いて、モデル全体がどのような規則性・判断基準を身につけているかを、人が解釈できるようにする。本技術については次章にてより詳細に動向を示す。

3.4 参考文献

- 〔Goodfellow 2015〕 Ian J. Goodfellow, Jonathon Shlens, Christian Szegedy, Explaining and Harnessing Adversarial Examples, <https://arxiv.org/abs/1412.6572>, v3, 2015 年
- 〔大場 2018〕 AI システムの品質保証の動向, SQuBOK Review 2018 Vol.3, pp. 1-12, 2018 年
- 〔石川 2019〕 石川 冬樹, 徳本 晋, 機械学習応用システムのテストと検証, 情報処理 Vol. 59 No.1, pp.25-33, 2019 年
- 〔Breck 2016〕 Eric Breck, Shanqing Cai, Eric Nielsen, Michael Salib, D. Sculley, What's your ML test score? A rubric for ML production systems, Reliable Machine Learning in the Wild - NIPS 2016 Workshop, 2016 年
- 〔EC 2019〕 European Commission, High-Level Expert Group on AI, Ethics Guidelines for Trustworthy Artificial Intelligence, <https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai>
- 〔Kuwajima 2019〕 Hiroshi Kuwajima, Fuyuki Ishikawa, Adapting SQuaRE for Quality Assessment of Artificial Intelligence System, The 30th International Symposium on Software Reliability Engineering, pp.13-18, 2019 年
- 〔Mehrabi, 2019〕 Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, Aram Galstyan, A Survey on Bias and Fairness in Machine Learning, <https://arxiv.org/abs/1908.09635>, 2019 年
- 〔Gunning 2016〕 David Gunning, Explainable Artificial Intelligence (XAI), <https://www.darpa.mil/program/explainable-artificial-intelligence>, 2016 年
- 〔Zhang 2020〕 Jie M. Zhang, Mark Harman, Lei Ma, Yang Liu, Machine Learning Testing: Survey, Landscapes and Horizons, IEEE Transactions on Software Engineering, 2020 年
- 〔Pei 2017〕 Kexin Pei, Yinzhi Cao, Junfeng Yang, Suman Jana, DeepXplore: Automated Whitebox Testing of Deep Learning Systems, The 26th Symposium on Operating Systems Principles, pp.1-18, 2017 年
- 〔Chen 2018〕 Tsong Yueh Chen, Fei-Ching Kuo, Huai Liu, Pak-Lok Poon, Dave Towey, T. H. Tse, Zhi Quan Zhou, Metamorphic Testing: A Review of Challenges and Opportunities, ACM Computing Surveys, Vol.51 No.1, 2018 年
- 〔Tian 2018〕 Yuchi Tian, Kexin Pei, Suman Jana, Baishakhi Ray, DeepTest: Automated Testing of Deep-Neural-Network-driven Autonomous Cars, The IEEE/ACM 40th International Conference on Software Engineering, pp.303-314, 2018 年
- 〔Dwarakanath 2018〕 Anurag Dwarakanath, Manish Ahuja, Samarth Sikand, Raghatham M. Rao, R. P. Jagadeesh Chandra Bose, Neville Dubash, Sanjay Podder, Identifying Implementation Bugs in

Machine Learning based Image Classifiers using Metamorphic Testing, The 27th ACM SIGSOFT International Symposium on Software Testing and Analysis, pp.118-128, 2018 年

〔Ma 2018〕 Lei Ma, Felix Juefei-Xu, Fuyuan Zhang, Jiyuan Sun, Minhui Xue, Bo Li, Chunyang Chen, Ting Su, Li Li, Yang Liu, Jianjun Zhao, Yadong Wang, DeepGauge: multi-granularity testing criteria for deep learning systems, The 33rd ACM/IEEE International Conference on Automated Software Engineering, pp.120-131, 2018 年

〔Harel-Canada 2020〕 Fabrice Harel-Canada, Lingxiao Wang, Muhammad Ali Gulzar, Quanquan Gu, Miryung Kim, Is neuron coverage a meaningful measure for testing deep neural networks?, The 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2020 年

4. 機械学習における説明可能性・解釈性

4.1 はじめに

本技術カタログは AI プロダクトの開発者、テスター、および品質保証に関わる人物、いわゆる説明責任を求められる人物が、これを参考にできるようにまとめたものである。

機械学習は高い精度での予測・認識を行うことができるが、その多くは判断根拠を説明していない。「中身が説明できないものは使えない」という懸念は AI プロダクトへの不信となり、導入を阻害する要因になりうる。そのため、AI プロダクトの判断根拠を説明することへの社会的ニーズが高まっている。

日本では総務省から「AI 利活用原則案」が策定されており、そこには「透明性の原則」として、AI プロダクトの利用者は AI システムの入出力の検証可能性、および診断結果の説明可能性に留意するよう記載されている。また「アカウンタビリティ（説明責任）の原則」として、AI プロダクトの利用者およびステークホルダーに対してアカウンタビリティを果たすよう求められている。海外では、EU の「一般データ保護規則（GDPR）」において、ユーザーに関する意思決定に説明責任が課されており、アメリカでは XAI（Explainable AI）の語源である DARPA での XAI プロジェクトが進められている。

機械学習モデルの説明可能性・解釈性に関する研究への注目は 2016 年以降に特に増してきている。近年の研究をまとめたサーベイ論文 [4][3] では、説明可能性・解釈性の研究は今後も増加する傾向となっている。こうした盛んに研究が行われる一方で、説明可能性・解釈性を求めるユーザーは、どの手法をどういった場合に使用すればいいかがわかり難い。本章では、機械学習モデルに対する説明可能性・解釈性を付与する手法を整理、分類し、以降に代表的な手法の詳細について紹介する。

なお、機械学習モデルが説明できたとしても、ユーザーがその説明内容を納得しなければならない。あくまで説明はユーザーが納得するための手段である。そのため、機械学習モデルがブラックボックスだと困る領域を対象とし、ブラックボックスモデルを全て否定する意図ではない。

4.2 説明可能性・解釈性を付与する手法の分類

機械学習モデルに対して、説明可能性・解釈性を付与する手法の分類方法は種々ある。例えば、文献 [2][1][6] の大阪大学産業科学技術研究所の原准教授によると、「どのような説明がしたいか」という出口ベースの分類として、大きく「大域的な説明」、「局所的な説明」、「説明可能なモデルの設計」の 3 つに分類される。

[原准教授による分類方法]

1. 大域的な説明

複雑なブラックボックスモデルを可読性の高い説明可能なモデルで表現することで説明とする方法。

2. 局所的な説明

特定の入力に対するブラックボックスモデルの予測の根拠を提示することで説明とする方法。
深層学習モデル、特に画像認識モデルの説明法も含む。

3. 説明可能なモデルの設計

そもそも最初から可読性の高い説明可能なモデルを作ってしまう方法。

また、三菱電機株式会社では、説明可能性・解釈性を 3 つの軸で分類している [5]。

[三菱電機株式会社による分類方法]

- (1). 説明可能性・解釈性を与える対象。
- (2). 説明可能性・解釈性を与えるタイミング。
- (3). 説明可能性・解釈性を与える方法。

(1) の分類は説明可能性・解釈性を与える対象で、モデル自体に付与する手法を Global、個々の推論結果に対して付与する手法を Local なアプローチと定義している。Global では予測傾向についての統計的な説明や、解釈できる簡易モデルに大域的に近似することができる。対して Local では、個々の予測結果についての説明となる。

(2) の分類は、説明性・解釈性を与えるタイミングで、Intrinsic と Post-hoc がある。Intrinsic はモデルの学習時に、モデルの内部的な特性や仕組みに対して説明可能性・解釈性を与える。あるいは最初から説明・解釈可能なモデルを学習するなどのアプローチを指す。Post-hoc はモデルの学習後に、学習済みのモデルを対象に説明可能性・解釈性を与えるアプローチを指す。説明可能性・解釈性の手法は Global/Local、Intrinsic/Post-hoc の組み合わせで大きく 4 つに分類される。さらに、(3) の説明可能性・解釈性を付与する方法 (Explanator) では、大きく 3 つのアプローチ (ルールベース、特徴ベース、インスタンスベース) に分類される。ルールベースは「～ならば～」といったルールを基に推論ロジックや判断根拠を説明する方法であり、特徴ベースは入力データのうち推論に大きく影響を与える重要な要素を重みづけする方法である。インスタンスベースはモデルの推論結果をデータセットに含まれるインスタンスや、それらを加工したデータにより説明する方法である。

これらの分類軸を用いて代表的な手法を整理し、原准教授の分類方法との対応を示したものを図 4.1 に示す。各々の代表的な手法については、後に詳細を述べる。XAI 技術は現在も研究が盛んな分野であり、AI 開発者はどのような XAI 技術があるのかその特性を広く把握し、用途に応じて使い分

ける必要がある。AI を利用する者や利用するシーンでどのような意思決定をするかを踏まえ、どのような情報を提示すれば懸念を払拭して AI の判断を信頼してもらえるのか等、人へのフィードバック方法の検討が非常に重要である。



図 4.1 説明性・解釈性の分類

Global の Intrinsic なアプローチは推論過程が明確な手法が用いられる。一般化線形モデル (GLM: General Linear Model) や一般化加法モデル (GAM: Generalized Additive Model) などのモデリング手法によってモデルの特徴からデータの透明性や解釈が得られる。決定木 (DT: Decision Tree) や決定則 (DR: Decision Rule) は、設定したルールに応じた解釈を得られる。また近傍ベースの手法 (Neighbor) は、ニューラルネットで学習させた上で個々のレイヤーの説明性を最近傍法 (nearest neighbor) でテストサンプルのクラス予測を行って説明する。

Global の Post-hoc なアプローチは、解釈可能な手法によりモデルの出力を近似する Surrogate の手法がある。例えば、DT Surrogate は、DT を用いて複雑なモデルを単純なサロゲートモデルに変換することで、複雑なモデルの内部を説明できる。

Local の Intrinsic な手法にはニューラルネットワークの性質を利用する方法が多く、特徴マップの自己回帰モデルにより注目点を推定する Attention などのアプローチがある。また影響関数 (Influence Function) により学習サンプルのひとつひとつが推論結果に与える影響を定量化したインスタンスベースの手法もある。

Post-hoc は、勾配ベースで顕著性マップ (Saliency Map) を推定する CAM(Class Activation Map)

やデータに対する摂動/変動を与え、出力の変換を解析するアプローチの Sensitivity Analysis がある。また入力データを摂動させて作った人工データを用いて解釈可能なモデルを学習する LIME などの Local Surrogate のアプローチがある。表 4.1 に各 Explanator について、代表的な手法を記載する。

表 4.1: Explanator

Explanator	Global /Local (G/L)	Intrinsic /Post-hoc (I/P)	代表的な手法の Reference
GLM/GAM	G	I	<p>GLM/GAM</p> <p>Hastie, T. J.; Tibshirani, R. J. (1990). Generalized Additive Models. Chapman & Hall/CRC. ISBN 978-0-412-34390-2.</p> <p>Madsen, Henrik; Thyregod, Poul (2011). Introduction to General and Generalized Linear Models. Chapman & Hall/CRC</p>
DT/DR	G	I	<p>DT/DR</p> <p>V. Schetinin et al., "Content interpretation of Bayesian decision tree ensembles for clinical applications," IEEE Trans. Inf. Technol. Biomed., vol. 11, no. 3, pp. 312319, May 2007.</p>
Neighbor	G	I	<p>KNN(k-nearest neighbor)</p> <p>N. Papernot, P. McDaniel, Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning (2018). arXiv:1803.04765.</p>
Surrogate	G	I	<p>Surrogate Model</p> <p>J. J. Thiagarajan, B. Kailkhura, P. Sattigeri, and K. N. Ramamurthy. (2016). "TreeView: Peeking into deep neural networks via feature-space partitioning." [Online]. Available: https://arxiv.org/abs/1611.07429</p>
Attention	L	I	<p>Attention</p> <p>Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.</p>

Influence Function	L	I	Influence Function Pang et al., Understanding Black-box Predictions via Influence Functions, arXiv:1703.04730, 2017.
Sensitivity Analysis	L	P	Gradient Boost Machine Friedman, Greedy Function Approximation: A Gradient Boosting Machine, Annals of statistics, vol. 29, no. 5, pp. 1189-1232, 2001.
Saliency Map	L	P	Saliency Map Karen et al., Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, arXiv:1312.6034v2, 2014.
CAM	L	P	Grad-CAM R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. arXiv preprint arXiv:1610.02391, 2016.
Local Surrogate	L	P	LIME M. T. Riphagen や R で XAI の代表的な手法が使えるようにツールの整理も進んでいる beiro, S. Singh, and C. Guestrin, “‘Why should i trust you?’: Explaining the predictions of any classifier,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2016, pp. 1135-1144.

前述した XAI 技術は、各論文の著者が個別に公開している実装以外にも、各種手法を集めて整理したライブラリが公開されている（表 4.2、表 4.3 参照）。

表 4.2: XAI 関連のライブラリ

名称	言語	説明	リンク

ELI5	Python	Python のライブラリ、各種の説明法・可視化法が実装されている。python で機械学習によく利用される scikit-learn にシームレスに繋がる様に設計されている。	https://eli5.readthedocs.io/en/latest/
iml	R	R のパッケージ、書籍 Interpretable Machine Learning: A Guide for Making Black Box Models Explainable の著者により種々の説明法がまとめられている。	https://github.com/christophM/iml
DALEX	R	R のパッケージ、パッケージ開発者の別リポジトリでは論文を初め、様々な情報がまとめられている。	https://github.com/pbiecek/DALEX

表 4.3: XAI 関連の Github

No.	説明	リンク
1	The Institute for Ethical AI & Machine Learning によってまとめられた AI 関連の技術リスト、Explaining Black Box Models and Datasets にライブラリや Git の情報が記載されている。	https://github.com/EthicalML/awesome-production-machine-learning#explaining-black-box-models-and-datasets
2	H2O.ai Machine Learning Interpretability team (https://github.com/h2oai/mli-resources) の機械学習ワークフローをベースとした AI 関連の技術リスト、Explainability- or Fairness-Enhancing Software Packages にライブラリや Git の情報が記載されている。	https://github.com/jphall663/awesome-machine-learning-interpretability/blob/master/README.md

4.3 説明可能性・解釈性を付与する代表的手法

4.3.1 GLM/GAM

表 4.4: GLM/GAM

概要	GLM (Generalized linear model, 一般化線形モデル) は機械学習モデルの一種であり、目的変数が説明変数の線形結合として表現される極めて単純なモデルである。 GAM (Generalized Additive Model, 一般化加法モデル) は GLM をより一般化したモデルである。両モデルは各説明変数が目的変数に与える影響を理解しやすいという特徴を持つ。
分類	Global, Intrinsic
対象ドメイン	理論上はどのようなデータに対しても適用可能。ただし画像に対しては可視化が難しく、かつ精度が出にくいためあまり用いられない。
対象モデル	GAM/GLM ともにそれ自体が解釈可能なモデルである。
実用例	データサイエンスプラットフォーム DataRobot にて GAM が実装されており、利用可能。
使用方法	GLM は、説明変数を係数パラメータで重みづけて線形に足し合わせた形で表現されるモデルである。訓練データに対し学習させたモデルのパラメータを分析することにより、目的変数に対する各説明変数の感度を知ることができる。 例えば、ある説明変数 x_1 の係数 w_1 が正の値であれば、 x_1 が増加すると目的変数 y の値も増加するという性質を持つことがわかる。あるいは w_1 が 0 であれば、 x_1 は y の値に影響を与えないといえる。 GAM は、説明変数を何らかの関数により変換したものを線形に足し合わせた形で表現されるモデルである。このため GLM より表現力が高い。GLM と同様に、モデルのパラメータを分析することにより、目的変数に対する各説明変数の感度を知ることができる。

効果：予測精度	GLM または GAM により、説明変数と目的変数の関係性を知ることができるが、人間が想定する関係性と異なる関係性が現れた場合、外れ値への過学習などを検知し再学習することができる可能性がある。
効果：信頼性	モデルのパラメータ分析の結果、人間が想定する説明変数と目的変数の関係性と同じ関係性が現れれば、高い信頼性が得られると考えられる。
懸念事項	GLM は極めて単純なモデルであり、解釈は容易だが精度が出ないケースもある。この場合はより複雑なモデルである GAM を用いるのが適切である。GAM は説明変数を何らかの関数により変換するが、この関数の複雑さによってモデルの説明性と精度のトレードオフが決定される。パラメータチューニングにより関数の複雑さを調整し、所望の説明性と精度を得る必要がある。
ライブラリ	pyGAM (Python 用 GAM ライブラリ) https://pygam.readthedocs.io/en/latest/ mrgcv (R 用 GAM ライブラリ) https://cran.r-project.org/web/packages/mrgcv/index.html
参考文献	[4.3.1-1] J Nelder, R Wedderburn, “Generalized Linear Models” . Journal of the Royal Statistical Society. Series A 135 (3) : 370 – 384. [4.3.1-2] T Hastie, “Generalized Additive Models” , Statistical models in S, 2017, Chapter 7.

4.3.2 DT

表 4.5: DT

概要	DT (Decision Tree:決定木) とは、データを段階的に分割していく、木構造の分析結果を出力する分析技法である。最初から解釈可能なモデルを出力する手法であり、機械学習モデルに対する解釈性の付与という意味では、利用者にとって利便性が高い。
分類	Global, Intrinsic
対象ドメイン	データの予測、判別、分類に用いられる。

対象モデル	DT は、それ自体が解釈可能なモデルである。
実用例	DataRobot ほか様々なツールに実装されており、利用可能。
使用方法	<p>1) データ分析の目的に沿って、決定木分析のアルゴリズムによりデータを分割していく。</p> <p>2) データ分割の場面では、不純度（データをきれいに分割できたかを表し、純粋な状態は 0（ゼロ）となる）と情報利得（分割の良さを表し、分割前の不純度 - 分割後の不純度で求める）により、データ分割の良さを判断する。</p> <p>3) 汎化性能を確保できるように、木構造の深さを制限する。</p>
効果：予測精度	木構造で表現されるため、分析結果の解釈が容易である。このため、得られた機械学習予測モデルの妥当性を判断しやすい。
効果：信頼性	木構造表現による可読性の高さから、信頼性は高い。
懸念事項	木構造が深くなると過学習（Over fitting）となる危険性がある。このため、木構造の深さを制限して、汎化性能を確保する必要がある。
ライブラリ	dtreeviz (可視化ツール) https://github.com/parrt/dtreeviz scikit-learn (Python 用機械学習ライブラリ) https://github.com/scikit-learn/scikit-learn rpart, partykit (R 用ライブラリ) https://cran.r-project.org/web/packages/rpart/index.html https://cran.r-project.org/web/packages/partykit/index.html
参考文献	[4.3.2-1] V. Schetinin et al., "Content interpretation of Bayesian decision tree ensembles for clinical applications," IEEE Trans. Inf. Technol. Biomed., vol. 11, no. 3, pp. 312319, May 2007.

4.3.3 Surrogate

表 4.6: Surrogate

概要	Surrogate は、Black Box のモデルを最初から解釈可能なモデルとして学習しなおすものである。初めに一般的な手法でモデルを生成し、後からそのモデルを近似する代理（Surrogate）モデルを解釈可能な手法で生成することで、そこから情報を読み取るものである。Global の Surrogate は、モデルが全体としてどの変数のどういった値に着目しているかを表すものである。
分類	Global, Post-hoc
対象ドメイン	画像、テキスト。
対象モデル	機械学習、深層学習。
実用例	なし。
使用方法	決定木における決定木代理モデルでは、元の複雑なモデルの元の入力と予測で決定木をトレーニングすることで作成される。変数同士の相互作用を見つけて確認するために、ICE (Individual Conditional Expectation) および部分従属 (Partial Dependence) を用いる。
効果：予測精度	決定木代理モデルによって変数の重要度、傾向や相互作用が読み取れる。決定木代理モデルでは、モデルそのものが等式や不等式で表される階層構造になっている。
効果：信頼性	期待値に対して整合している場合に、ユーザーへの信頼性を向上することにつながる。
懸念事項	単純なモデルが複雑なモデルの内部メカニズムを完全に表すわけではない。
ライブラリ	調査中
参考文献	[4.3.3-1] J. J. Thiagarajan, B. Kailkhura, P. Sattigeri, and K. N. Ramamurthy. (2016). “TreeView: Peeking into deep neural networks via feature-space partitioning.” [Online]. Available: https://arxiv.org/abs/1611.07429

4.3.4 TCAV

表 4.7: TCAV

概要	CAV (Concept Activation Vectors) は、定義したコンセプトが分類器の推論にどれくらい影響しているかを定量化する手法であり、これをデータセットに適用しテスト技術として適用したものが TCAV (Testing with Concept Activation Vectors) である。
分類	Global (TCAV:モデル全体), Local (CAV:一変数), Post-hoc
対象ドメイン	論文では画像処理が対象。入力となる画像データが、予め用意したコンセプトにどれくらい強く従うかを定量化することができる。
対象モデル	対象となる説明手法が適用可能な機械学習のモデル (DNN, SVM 等)。
実用例	なし。
使用方法	<p>通常の機械学習モデル開発は終了し、学習済みの分類器があるものとする。</p> <ol style="list-style-type: none"> 1) ユーザーが定義した「コンセプト」の例と、「コンセプト」と異なるランダムな例を用意 2) コンセプト例とランダム例を DNN に入力し、任意の中間層まで順伝搬させた圧縮特徴を、それぞれの特徴量とする 3) 特徴量空間をコンセプト、ランダムの 2 ラベルに分離する超平面を求め、これを CAV とする 4) 解析のターゲットとなるデータを入力し、2 と同一の中間層でデータを取得。これを感度とし、これと CAV との内積を Conceptual Sensitivity とする。 5) 評価対象のデータセットについて、データを CAV 方向に移動させたとき、対象のクラスに伝達される信号が強くなる度合いを算出する。これを TCAV 値と定義している。 6) TCAV 値が大きいほど、クラスが設定した概念に沿っていると判定できる。
効果：予測精度	TCAV の結果をもとに訓練データやモデルを修正し、予測精度の向上が可能。例えば TCAV の結果から、本来推論に影響すべきでないコンセプトが推論に影響していることがわかった場合、訓練データに新たなデータを追加することで、予測精度が向上する可能性があると考えられる。

効果：信頼性	開発したモデルが、期待した（人間の知覚と近しい）汎化された知識を獲得しているかどうかを定量化することができる。これ用いて、モデルの Validation が可能になると考えられる。 敢えてバイアスがかかるようなコンセプトを検証することで、そのバイアスによる悪影響がないことを示すことも可能。（論文中においても、画像内に文字を含めるという実験で検証している）
懸念事項	CAM などと共に懸念になるが、特徴抽出に選択する層を一意に決定する方法がなく、どの層で抽出した情報が本当に正しい解釈となるかが実験的である。 また、以下の 2 点について適用は限定される。 コンセプトデータの形成 取り扱うデータが、人間が一意的に解釈するコンセプトを形成できる必要がある。論文は画像を対象としており、例えば、シマウマが馬と、ストライプというコンセプトに分解できたことを紹介している。近年、急速に発達した自然言語処理（NLP）について考えると、コンセプトの形成は画像ほど単純ではない。 モデルの制約 「使用方法」に記載の通り、モデルは、入力を何らかの圧縮をしながら最終的分類信号を抽出するようなモデルである必要がある。決定木のようなシンプルなモデルに適用する場合は、その特徴量をどう設定するのが妥当かを検討する必要がある。
ライブラリ	https://github.com/tensorflow/tcav
参考文献	[4.3.4-1] Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV) , Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Jun, Cai James, Wexler Fernanda, Viegas Rory, Abbott Sayres, ICML 2018.

4.3.5 Attention

表 4.8: Attention

概要	Attention 機構（注意機構）とは、主に機械翻訳や画像処理等を目的とした深層学習モデルに導入される入出力要素ごとの関係性、注意箇所を学習する手法である。
分類	Local, Intrinsic
対象ドメイン	機械翻訳、画像処理。
対象モデル	DNN。
実用例	現段階 Attention 機構は研究レベルで、画像処理、自動運転 [4.3.5-4] と機械翻訳 [4.3.5-3] などの分野への応用が見込まれている。
使用方法	Attention 機構を用いた機械翻訳では、入力文字列を符号化するエンコーダ部の各単語の隠れ状態と、復号化するためのデコーダ部で目標単語翻訳時の隠れ状態を使って新たなコンテキストベクトルを計算する [4.3.5-3]。コンテキストベクトルはデコーダ部の単語推定時に使われる。Attention 機構を用いることで、長文での翻訳精度の向上と共に、alignment と呼ばれる機械翻訳分野で、翻訳前の文と翻訳後の文の対照関係を分析する処理でも有用な結果を生成できる。
効果：予測精度	機械翻訳では、単語と本文の関連性を示すことができ、言語の文法構造分析が可能である。 自動運転、画像処理の応用例としては、前方画像のピクセルが運転操作との関連性を可視化することができ、運転シーンへの重要度を直観的に評価できる。
効果：信頼性	Attention は入出力要素ごとの関係性を可視化することができ、ユーザーへの信頼性を向上することにつながる。
懸念事項	Attention 機構を用いることで、入出力データの対応関係を定量的に分析することができるようになった。機械翻訳で翻訳した単語と入力した単語の対応関係を分析できることが一つの例である。しかし、品質保証技術開発の時に、入出力データの対応関係は機械翻訳のように明らかになっていない場合がある。例えば、工場の機械に装着したセンサの故障検知タスクにおいて、Attention 機構を用いてセンサ値予測し、入出力センサデータの関連性を分析することはできるが、その関連性を人間に説明可能かという議論がある。Attention 機構から得られたデータ関連性を解釈可能にする技術がこれから注目されるだろう。

ライブラリ	https://www.tensorflow.org/tutorials/text/nmt_with_attention
参考文献	<p>[4.3.5-1] Mikolov, Tomáš, et al. "Recurrent neural network based language model." Eleventh annual conference of the international speech communication association. 2010.</p> <p>[4.3.5-2] Gehring, Jonas, et al. "Convolutional sequence to sequence learning." Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017.</p> <p>[4.3.5-3] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.</p> <p>[4.3.5-4] Liu, Nian, Junwei Han, and Ming-Hsuan Yang. "Picanet: Learning pixel-wise contextual attention for saliency detection."Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.</p> <p>[4.3.5-5] Wang, Dequan, et al. "Deep object-centric policies for autonomous driving." 2019 International Conference on Robotics and Automation (ICRA) . IEEE, 2019.</p> <p>[4.3.5-6] https://qiita.com/itok_msi/items/ad95425b6773985ef959</p> <p>[4.3.5-7] http://www.thothchildren.com/chapter/5c0b968d41f88f26724a70b8</p>

4.3.6 Sensitivity Analysis

表 4.9: Sensitivity Analysis

概要	Sensitivity Analysis（感度分析）は、重要な要因の中に不確実な要因がある場合に、これらの値を変動させることで、全体の結果にどの程度影響するかを把握する分析手法である。経済、数理、医療、システム分析など、長年さまざまな分野で活用されている。感度分析を AI に応用する場合、データを意図的に変動させてシミュレーションし、AI モデルの振る舞いと出力の予測について、精度、公平性、セキュリティ、および安定性が許容可能かどうかを評価することができる。また、評価結果を AI モデルの局所的な説明に使用することができる。
----	---

分類	Local, Post-hoc
対象ドメイン	AI モデルの実装が設計意図に合っているかを確認するための検証手法の 1 つとして活用できる。特に、入力データの小さな変化により、予測される出力が大きく変化する可能性の場合の検証に効果的である。また、AI モデルのデバッグ手法として活用することもできる。
対象モデル	本手法は、モデル非依存の手法であるため、広く活用することができる。
実用例	さまざまな分野で既に実用化されている手法である。
使用方法	<p>1) 説明または公平性に影響のある重要なデータを選定する。 例えば、LIME を活用することで、重要なデータの選定の参考にできる。</p> <p>2) 許容可能な説明または公平性の値の変化に対する許容可能なしきい値を決定する。 許容可能なしきい値が妥当であることを事前確認しておく必要がある。</p> <p>3) 手動または自動で入力データを変動させ、説明または公平性の値に対して、許容できない変化の監視を開始する。</p> <p>3a) これらの値の変化が許容可能なしきい値内であれば、説明と公平性の技術は安定しており問題ない。</p> <p>3b) これらの値の変化が許容可能なしきい値を超えた場合、デバッグして要因を究明した上で、AI モデルの改善を行う。</p>

効果：予測精度	<p>説明性の向上</p> <p>モデルの動作と出力が時間とともにどのように変化するかを示すことで、理解や説明性を高めることができる。また、特定のデータの変化に対するモデルの動作と出力を示すことで、局所的な説明に使用することもできる。</p> <p>備考</p> <p>従来の線形モデルと異なり機械学習アルゴリズムは、非常に複雑な非線形の単調でない応答関数を生成するため、入力変数とターゲット変数間の相関関係が複雑である。そのため、隠れた相関関係について静的なトレーニングデータを検索するよりも、感度分析手法を活用してモデル予測の潜在的な不安定性に焦点を当てた方が、短時間で適切な確認がしやすい。</p>
効果：信頼性	<p>感度分析によって十分に検証済みであれば、対象となる AI モデルが、人間が持つドメイン知識と期待を順守できる状態にあり、十分な信頼性が確保できていると言える。この場合、データが微妙に想定内の破損をしても、モデルの動作と出力は安定させることができる。</p>
懸念事項	<p>機械学習の場合、モデルパラメータの数値的不安定性にあまり焦点を当てない方がよい。機械学習アルゴリズムは非常に複雑な非線形の単調でない応答関数を生成するため、従来の線形モデル検証手法は適さない。</p> <p>どの入力データを、どのようにどの程度変動させればよいかを決定するのが難しい。ブラックボックスに包まれており、変動要因（パラメータ）と結果の関係性が不明確なためである。</p> <p>入力変数値の小さな変化により、予測される応答が大きく変化する可能性がある。例えば、機械的に上下 20 % の変動幅をとるのではなく、現実に起こりうる可能性を関係者間で議論した上で、個々の変数ごとに変動幅を決めることが望ましい。</p>

ライブラリ	Cleverhans https://oreil.ly/3038mur https://github.com/tensorflow/cleverhans Foolbox https://oreil.ly/31NsDoD What-If Tool https://oreil.ly/2KJGHZ7 https://pair-code.github.io/what-if-tool/ Interpretable Machine Learning with Python https://oreil.ly/33xjthx
参考文献	[4.3.6-1] An Introduction to Machine Learning Interpretability , 2nd Edition - An Applied Perspective on Fairness, Accountability, Transparency, and Explainable AI - https://www.oreilly.com/library/view/an-introduction-to/9781098115487/ [4.3.6-2] 観察研究における感度分析の勧め 入門編 . . . 観察研究に基づく意思決定に関わる全ての方へ . . . http://www.jpma.or.jp/medicine/shinyaku/tiken/allotment/sensitivity_analysis.html http://www.jpma.or.jp/medicine/shinyaku/tiken/allotment/pdf/sensitivity_analysis.pdf

4.3.7 CAM

表 4.10: CAM

概要	CAM (Class Activation Mapping) は、CNN (Convolution Neural Network) を用いた画像解析・分類において、その判断根拠を可視化する手法である。
分類	Local, Post-hoc
対象ドメイン	画像。
対象モデル	DNN。

実用例	日立製作所にて、技術者やコンサルタント向けツールとして、AI の判断根拠を顧客企業の現場担当者に説明し、必要に応じて AI モデルを改善するために、「Grad-CAM (Gradient-weighted Class Activation Mapping)」が利用されている。 https://xtech.nikkei.com/atcl/nxt/news/18/06939/
使用方法	CNN 層における特徴抽出計算後の各特徴マップに対して、GAP (Global Average Pooling)、つまり画素平均値を計算し、分類のクラスとマッピングを行う。マッピングの結果、各特徴マップに対する重みが出力されるので、各特徴マップに重み付けを行い、元の画像に重ねることで、マッピング結果の寄与度を示すヒートマップを作成する。[4.3.7-1]
効果：予測精度	システムが画像のどこに注目して分類を行ったかを示すことができるため、品質を検証することが可能になる。たとえ分類に成功していたとしても、注目領域が適切でなければ、学習方法を修正するなどの判断ができるようになり、予測精度向上につながる。
効果：信頼性	CAM は判断根拠を可視化することができ、ユーザーへの信頼性を向上することにつながる。
懸念事項	CAM は GAP や Softmax 関数を用いているため、ベースとなる画像分類モデルより精度が落ちてしまうことから、現在は各特徴マップの重みづけ部分と GAP の部分を逆伝搬時の勾配、つまりは変動が大きくなる部分で代用できる Grad-CAM (Gradient-weighted Class Activation Mapping) [4.3.7-2] という発展手法が主流となっており、さらにそこから特徴領域にキャプションを追加したり、画像についての質問と応答を行う VQA (Visual Question Answering) などの様々な可視化手法に応用されている。 また、このような手法の多くは敵対的攻撃に脆弱であり、画像中に目に見えないノイズを乗せることで、判断根拠としての注目箇所を変えることができることが知られている。[4.3.7-3]
ライブラリ	https://github.com/jazzsaxmafia/Weakly_detector

参考文献	<p>[4.3.7-1] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, Antonio Torralba, Learning Deep Features for Discriminative Localization, Computer Vision and Pattern Recognition, 2015.</p> <p>[4.3.7-2] Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization, Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra, Computer Vision and Pattern Recognition, 2016.</p> <p>[4.3.7-3] Ian J. Goodfellow, Jonathon Shlens, Christian Szegedy, Explaining and Harnessing Adversarial Examples, Computer Vision and Pattern Recognition, 2015.</p>
------	--

4.3.8 LIME

表 4.11: LIME

概要	LIME (local interpretable model-agnostic explanations) は、複雑なモデルを線形回帰で近似することで判断根拠を可視化する手法である。
分類	Local, Post-hoc
対象ドメイン	画像、テキスト。
対象モデル	線形回帰。
実用例	研究段階。
使用方法	対象とするサンプルの周囲のデータ空間から、サンプリングと予測を繰り返し行うことで得られるデータセットを教師データとして、線形回帰モデルを作成する。
効果：予測精度	システムが画像やテキストのどこに注目して分類を行ったかを示すことができるため、品質を検証することが可能になる。 たとえ分類に成功していたとしても、注目領域が適切でなければ、学習方法を修正するなどの判断ができるようになり、予測精度向上につながる。
効果：信頼性	LIME は判断根拠を可視化することができ、ユーザーへの信頼性を向上することにつながる。

懸念事項	LIME は局所的に近似するため、特徴空間において離れたデータを入力すると期待する結果と乖離する場合がある。
ライブラリ	https://github.com/marcotcr/lime
参考文献	[4.3.8-1] Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin, "Why Should I Trust You?": Explaining the Predictions of Any Classifier, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August, 2016.

参考文献

- [1] 【記事更新】私のブックマーク「機械学習における解釈性」. https://www.ai-gakkai.or.jp/my-bookmark_vol33-no3/.
- [2] 【記事更新】私のブックマーク「説明可能 AI」(Explainable AI). https://www.ai-gakkai.or.jp/my-bookmark_vol34-no4/.
- [3] A. ADADI et al: “Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI) .” In: VOLUME XX, 2018.
- [4] Guidotti et al. “A Survey of Methods For Explaining Black Box Models.” In: arxiv, 2018.
- [5] 濱光孝之 他. “機械学習モデルの解釈性に関する最新動向”. In: 電子情報通信学会. Vol. Vol.102, No.10. 2019/10. URL: https://app.journal.ieice.org/trial/102_10/k102_10_973/index.html.
- [6] 機械学習モデルの判断根拠の説明. https://www.slideshare.net/SatoshiHara3/ss-126157179?qid=91472032-d83b-4d83-9305-a60e80f3aed9&v=&b=&from_search=4.

5. コンテンツ生成系システム

5.1 想定するシステム

創造性・自然さ・面白さなど、人間が感じ取る何かの「良さ」を実現するような文章や会話、画像、動画などのコンテンツを生成するシステムを対象とする。こういったシステムにおいては、「どのようなものがどういう傾向（分布）で自然に存在するのか」を学習する、生成モデルと呼ばれる技術が用いられる（画像認識などで用いられる「ものの違い・境界」のみを学ぶ識別モデルと対比される）。2022 年の StableDiffusion をはじめとした高性能の Text-to-Image モデルの公開以降、より手軽に意図に応じた画像の生成が行いややすくなり、産業応用もより広く追及されるようになった。一方で、これらのシステムにおいては、人間の感覚的な満足を実現することが重要であり、品質評価を客観的に行ったり自動化したりすることが難しいという特徴がある。本章においては、生成モデルを用い、特に人の感覚により評価されるようなコンテンツ生成システムの品質保証アプローチについて論じる。

このようなシステムとしては、以下のようないわゆる商用化や研究開発されている。

- 画像生成・動画生成システム：自然な画像・イラストや狙ったアニメの生成や、大まかな姿勢の指定や属性を基にして、新たな画像や動画の生成。
- 文章生成・対話生成システム：川柳や小説など楽しむことを目的とする文章の生成や、特定のタスクの遂行ではなく面白さや意外性を目的とする対話の生成。
- 音声合成・声質変換システム：特定キャラクターの雰囲気や個性を出した、楽しませる音声の生成。

5.1.1 本章で扱う応用領域

本章においては、上述のシステムの代表例として、画像や動画、および 3D モデルなどの構造モデルを生成するシステムを考える。例えば、Web サイトや印刷物において閲覧者を楽しい気分にさせるような画像を生成し配置したり、映画やゲーム、アニメの構成要素となる動画や 3D モデルを生成したりする応用が考えられる。このようなシステムの例を以下に紹介する。

画像生成例 - 無指定での画像生成

何も指定せず、多様で自然な画像の生成。

参考事例（図 5.1）： A Style-Based Generator Architecture for Generative Adversarial Networks.
Tero Karras, Samuli Laine, and Timo Aila. In CVPR 2019.



図 5.1 画像生成例 - 無指定での画像生成 (Karras et al., in CVPR 2019)

画像生成例 - 種類を指定しての画像生成

画像の被写体を指定し、多様で自然な画像を生成。

参考事例（図 5.2）：Large Scale GAN Training for High Fidelity Natural Image Synthesis. Andrew Brock, Jeff Donahue, and Karen Simonyan. In ICLR 2019.



Figure 6: Additional samples generated by our model at 512×512 resolution.

図 5.2 画像生成例 - 種類を指定しての画像生成 (Brock et al., in ICLR 2019)

画像生成例 - 背景見本とレイアウトを指定しての背景生成

美術画像、レイアウト指定（物体の領域配置）を与え、美術画像の各領域の詳細テクスチャを反映し、指定レイアウトでの背景を生成。

参考事例（図 5.3）： Anime Generation with AI. DeNA, 2020. <https://www.slideshare.net/hamadakoichi/anime-generation-ai>

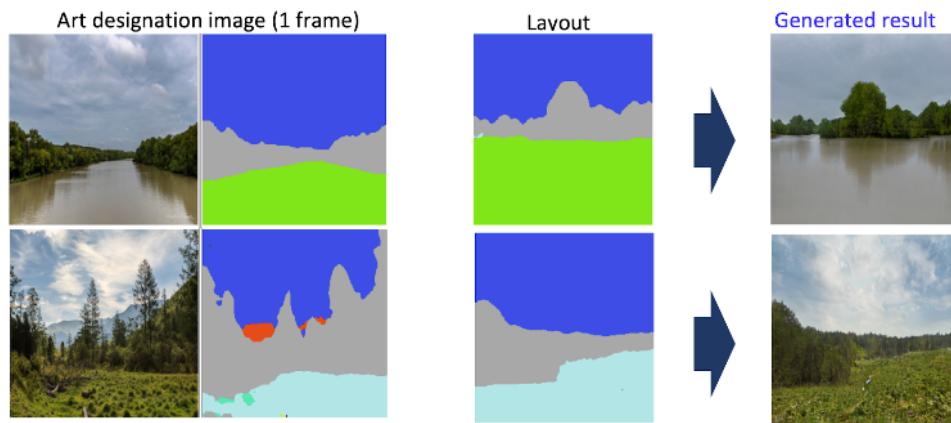


図 5.3 画像生成例 - 背景見本とレイアウトを指定しての背景生成 (DeNA, 2020)

画像生成例 - 色見本を指定しての線画彩色

色見本、線画、ラフ部位領域を指定し、各部位の色パターン・線詳細を厳密に反映した彩色。

参考事例（図 5.4）： Anime Generation with AI. DeNA, 2020. <https://www.slideshare.net/hamadakoichi/anime-generation-ai>, <https://youtu.be/X9j1fwexK2c?t=191>

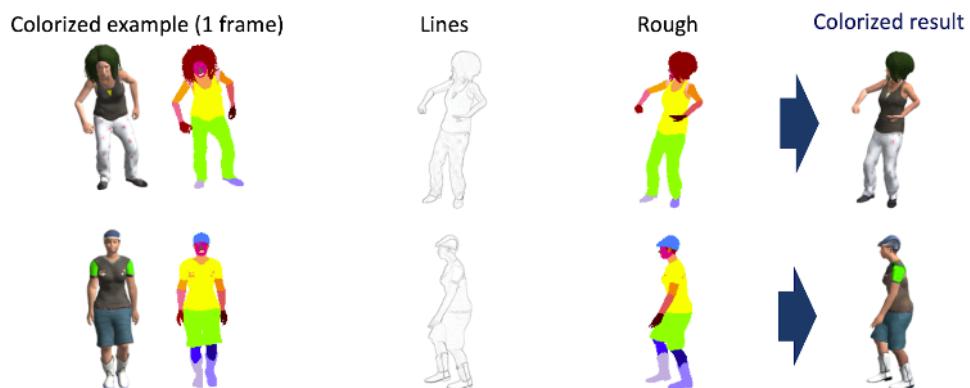


図 5.4 画像生成例 - 色見本を指定しての線画彩色 (DeNA, 2020)

動画生成例 - キャラクター画像と構造系列を指定しての動画生成

キャラクターにとらせたい構造を表す、構造情報の系列を与え、動画を生成。

キャラクターにとらせたい姿勢を表す座標モデルの系列（姿勢のリスト）を与え、多様なキャラ

クターが、指定された姿勢を順にとって動いていく自然な動画を生成。

参考事例（図 5.5）：Full-body High-resolution Anime Generation with Progressive Structure-conditional Generative Adversarial Networks. Koichi Hamada, Kentaro Tachibana, Tianqi Li, Hiroto Honda, and Yusuke Uchida. In ECCV Workshop 2018. <https://youtu.be/bIi5gSITK0E> <https://youtu.be/0LQ1fkvQ30k>

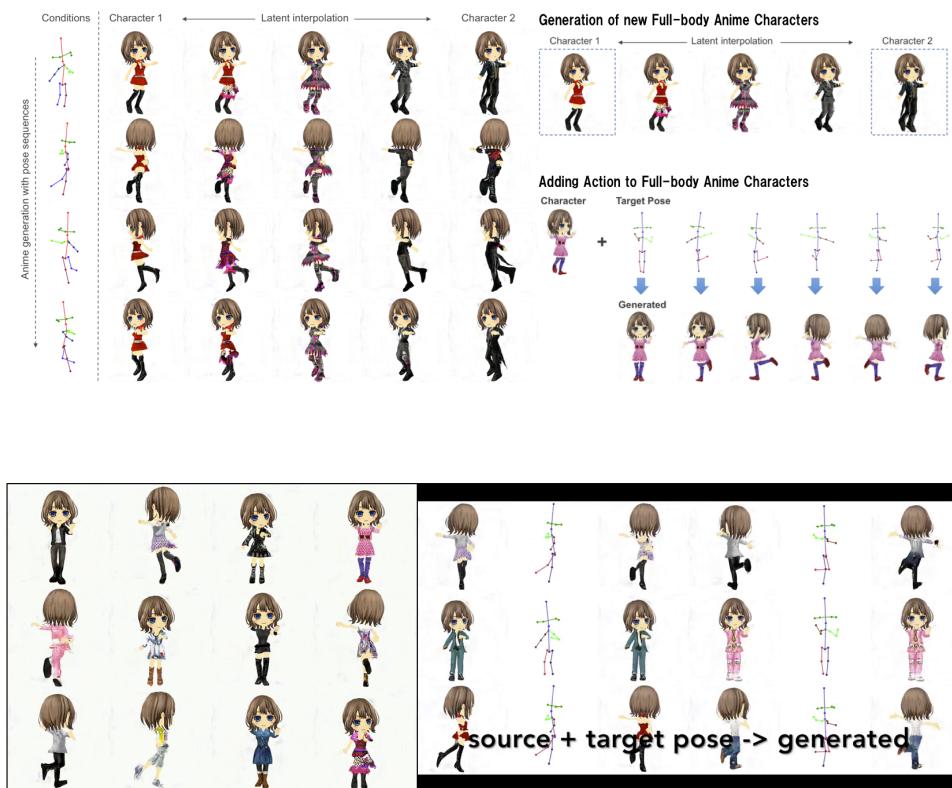


図 5.5 動画生成例 - キャラクター画像と構造系列を指定しての動画生成 (Hamada at el., in ECCV Workshop 2018)

キャラクターを描きたい各部位形状の系列（領域情報のリスト）を与え、多様なキャラクターが、指定された各部位の形状・姿勢を順にとて動いていく自然な動画を生成。

参考事例（図 5.6）：Anime Generation with AI. DeNA, 2020. <https://www.slideshare.net/hamadakoichi/anime-generation-ai> <https://youtu.be/X9j1fwexK2c?t=166>

動画生成例 - キーフレームを指定しての中割生成

キーフレームとして始点画像と終点画像を与える、それらの画像の間を「補間」するような自然で多様な動画を生成。

参考事例（図 5.7）：AI によるアニメ生成の挑戦. 濱田晃一, 李天琦. In DeNA TechCon

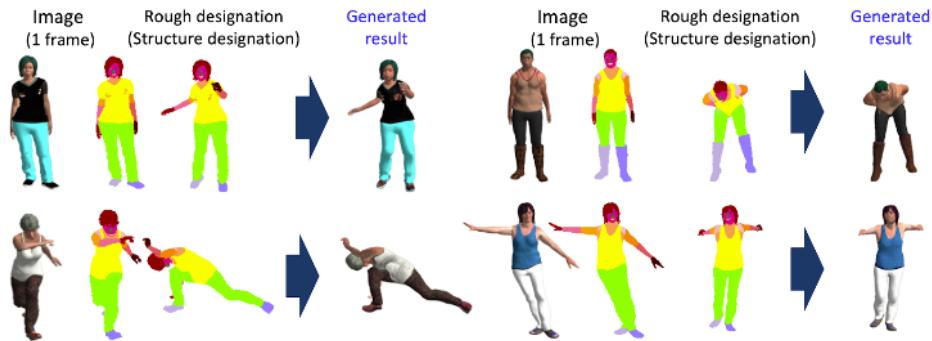


図 5.6 動画生成例 - キャラクター画像と構造系列を指定しての動画生成 (DeNA, 2020)

2019. <https://www.slideshare.net/hamadakoichi/anime-generation> <https://www.slideshare.net/hamadakoichi/anime-generation-ai> https://youtu.be/t0ZW_KWb8b0 <https://youtu.be/X9j1fwexK2c>



図 5.7 動画生成例 - キャラクター画像と構造系列を指定しての動画生成 (DeNA, 2019)

5.1.2 背景 - 近年の AI による生成モデルの進展 -

近年、深層生成モデルの各種方法論の発展により、AI による生成の可能性が大きく拓けてきている。特に、Generative Adversarial Networks (GANs) [Goodfellow+14] の 2018 年以降の進展 [Karras+18,Brock+19,Karras+19] により、本物と見間違うほどの高品質な生成が行われるようにな

なってきている。また従来、人の全身のような複雑な構造での生成を行うことは難しく、高品質生成は、車両・部屋などの剛体や、顔のみの生成等の身体でも一部の部位のみの生成に限られていた [Karras+18,Brock+19,Karras+19] が、最近の高品質生成と構造整合性を両立させる技術進展 [Hamada+18] により、キャラクター全体といった複雑な構造での高品質生成が可能になり、多様なキャラクター全体でのイラスト・アニメーションの生成といった、アニメやゲームの実産業への応用可能性が大きく拓けてきている [Hamada+18, Hamada+19]。

GANs [Goodfellow+14] は Generator (生成器) と Discriminator (識別器) と戦わせ、生成品質を向上させていく生成モデルである。識別器は本物データと生成器による生成データを識別し、生成器は生成データを識別器に本物データと誤識別させようと生成品質を向上させていく。識別器と生成器を相互に向上させ均衡点（ナッシュ均衡）に到達すると、生成器による生成データの生成分布が本物データと一致することが保証されている。しかしながら GANs の学習は難しく、特に高解像度・高品質の生成を多様な対象に対する学習を安定化する方法論が重要である。

Karras ら [Karras+18] は、その課題を解消する方法として 4x4 の低解像度画像で生成器・識別器を戦わせることから始め、進歩的に解像度をあげ、生成器・識別器を成長させることにより、1024x1024 解像度で高品質で多様な顔画像の生成を実現した。

生成対象の種類を指定する生成に関し、ImageNet [Russakovsky+15] の 1000 クラスを指定しての生成においても生成品質が大きく向上してきている [Odena+17,Miyato+18,Zhang+18,Brock+19]。Brock ら [Brock+19] は各種学習安定化の方法論を用い大規模バッチサイズ・チャンネルでの学習により、512x512 解像度での 1000 クラス指定の高品質な生成を実現している。

また産業応用上重要な課題が残されていた、人の全身のような複雑な構造での生成 [Karras+18,Brock+19,Karras+19] に関しても進展している [Hamada+18, Hamada+19,Hamada+20]。濱田ら [Hamada+18] は、構造条件づけられた GANs を 4x4 解像度から進歩的に学習していくことにより、高品質生成と構造整合性を両立させ、1024x1024 解像度でのキャラクター全身といった複雑な構造での高品質生成を実現している。キャラクターにとらせたい姿勢を表す座標を指定しての画像生成、生成系列を指定しての動画生成、等、多様なキャラクター・アニメが生成される。またアニメ制作で行われる、始点となる画像と終点となる画像を与え画像間を「補間」するような自然で多様な動画を生成においても、構造条件づけられた生成学習を用いることにより、従来難しかった構造変化の大きい画像間の補間生成を実現している [Hamada+19,Hamada+20]。上記進展により、多様なキャラクター全体でのイラスト・アニメーションの生成といった、アニメやゲームの実産業への応用 [Hamada+18, Hamada+19,Hamada+20] 等、生成モデルの各種実用化へ向けた可能性が大きく拓けてきている。

5.1.3 議論に用いるユースケース

本ガイドラインにおける議論のための例として、アニメやゲームの産業応用を想定したイラスト・アニメーションの生成アプリケーションに関するユースケースを考える。

本例題の議論においては、生成する画像・動画は、実写ではなくイラストであり、人間のキャラクターである場合に限る。また、システムが自動的に訓練データを獲得し、モデルの更新、配備を行うことは考えない。すなわち、Web 上など制御の効かない不特定多数のデータから学習し続けることは考えず、訓練データの追加やモデルの更新は、開発者による指示の下行われ、開発者による品質評価が行われることを想定している。

生成モデルの開発においては、自然に存在するものに近いコンテンツを生成することが第一の目標となる。これから直接実現できる機能は、自然なコンテンツを多数、多様に生成するというものである。一方で産業応用においては、生成するコンテンツに制約を加えたいと考えられる。例えば、特定アニメキャラクターがある程度決まった動きをする動画を生成したい、といった場合である。このため、本章における例題においては、「どのような画像・動画を生成するか」という指定の有無や、指定の仕方に関し異なるケースを検討する。

以下では、本例題で検討する 5 つのユースケースについて述べる。UC-0 は、生成モデルの活用として最も基本的、直接的なものである。これに対し、UC-1A/2A においてはそれぞれ画像・動画におけるキャラクターの構造（姿勢）について指示に従うことを求めている。UC-1B/2B においては、それぞれ動画・画像におけるキャラクターの属性について指示に従うことを求めている（正確には、UC-2B には姿勢の大まかな指示も含まれている）。これらのユースケースについては、構造、キャラクターそれについて指定に沿うような出力を行う必要があり、UC-0 で扱う品質特性に加えて、それら指定に応じた特性を扱う必要がある。また、UC-2A/2B の動画生成においては、UC-1A/1B の画像生成で扱った品質特性を考慮するよう拡張して扱う必要がある。

【ユースケース UC-0：無指定・画像】

何も指定せず、多様で自然な画像を生成する

【ユースケース UC-1A：構造指定・画像】

人物にとらせたい姿勢を表す情報として、各器官点（関節など主要な部位）の座標を与え、多様なキャラクターの指定姿勢の画像を生成する

【ユースケース UC-1B：属性指定・画像】

生成対象の属性（性別や服の色など）を与え、多様で自然な画像を生成する



図 5.8 器官点座標からの画像生成

【ユースケース UC-2A：構造系列指定・動画】

人物にとらせたい姿勢を表す情報であり各器官点の座標を、複数並べたリストを与え、多様なキャラクターが、指定された姿勢を順にとって動いていく自然な動画を生成する。

【ユースケース UC-2B：端画像指定・動画】

始点となる画像と、終点となる画像を与え、それらの画像の間を「補間」するような自然で多様な動画を生成する。

入出力

制約のない UC-0 を除き、それぞれのユースケースにおける入力について以下に示す。この入力された制約の指定に従いつつ、指定されていないことについては自然で多様な出力を出すことが、各ユースケースにおける該当機能の目的となる。なお、動画のフレームレートなど詳細な情報は割愛する。

- 画像
 - UC-1A：構造（姿勢）を表す、各器官点の座標（図 5.8 参照）
 - UC-1B：性別や服の色などの属性を選択肢の中から指定したもの
- 動画
 - UC-2A：構造（姿勢）を表す各器官点の座標を、リストとして並べたもの
 - UC-2B：それぞれ動画の始点と終点となる 2 つの画像

5.2 特有の課題

例題に代表されるコンテンツ生成システムにおいては、品質について特に以下に述べる点が特徴的である。このシステムにおいては、「新しい画像や動画をつくり出す」といった、創造的とも言える高度な機能を実現している。このため、その達成すべき品質特性は、「自然である」「多様である」「入力画像のテイスト（画風等）が維持されている」など、非常に抽象的に表現されるものとなる。このため究極的には人が評価をすることが必要であるが、現実的にはどのような自動判断基準ならば実現できるかということが問題となる。

5.3 期待される品質特性

以下では例題に触れつつ、画像や動画の生成システムにおいて期待される品質特性について述べる。ISO 25010 (SQuaRE) では主に「適切性」や「満足性」に分類されるような特性となる。

5.3.1 全ユースケースに共通する品質特性

【品質特性 QC01：自然さ】

生成された画像、あるいは動画の各コマが自然である。例えば、すでに存在するイラストと比べて違和感がない、人が描いたと言われても違和感がない。

【品質特性 QC02：鮮明さ】

生成された画像あるいは動画が鮮明である。例えば、高解像度で閲覧や印刷をしたときにも、形状や色合いの崩れやノイズがない。

【品質特性 QC03：多様さ】

多様な画像あるいは動画が生成される。例えば、キャラクターの服装や姿勢が常に同じになってしまっているといったことがない。これは、生成モデルが分布内の限られた点だけを近似してしまうという技術課題（Mode Collapse と呼ばれる）が起きていないことの確認ともなる。

【品質特性 QC04：社会的適切さ】

生成された画像あるいは動画が、社会的に不適切なものとなっていない。例えば、裸に見えるようなもの、残虐性を感じさせ生理的嫌悪感を発生させるようなものが生成されない。

5.3.2 コンテンツの指定に関する品質特性

【品質特性 QC05：指定構造との合致】

生成された画像が指定された構造と合致している。ユースケース UC-1A における例としては、指定構造において右腕を上げるようになっていたならば、生成された画像で右腕が上がっている。別の例としては、指定構造として画像内のどの位置にどのオブジェクトを置くかの領域配置を指定した場合に、その指定にあった形での画像が生成されている。

【品質特性 QC06：指定属性との合致】

生成された画像が指定された属性と合致している。ユースケース UC-1B における例としては、指定として性別、服装、表情、肌の色、髪型、体型、アクセサリが指定されたものになっている。別の例としては、指定属性としてテクスチャやスタイルを示す参照画像を与えた場合に、その指定を反映された画像が生成されている。

5.3.3 動画に関する品質特性

【品質特性 QC07：動画としての自然さ】

生成された動画が自然である。例えば、すでに存在する動画と比べ、画像列として違和感がない、本物の動画と言われても違和感がない。

【品質特性 QC08：動画のなめらかさ】

生成された動画がなめらかである。例えば、色合いや姿勢に連続性がない部分がない。

【品質特性 QC09：構造系列としての自然さ】

生成された動画の構造に描かれている対象の構造変化が自然である。例えば、走り方・ジャンプの仕方、等、各動作に関する身体構造（器官点）の系列が、既存の動画と比べとして違和感がない、本物の動画と言われても違和感がない。

5.4 品質評価・保証のための技術アプローチ

4.3 で挙げた品質特性の大半は、その充足度合いを直接評価するような客観的指標が存在せず、よって充足度合いの評価を自動化することも難しい。人による検査は非常に大きな工数がかかるとして、技術的・現実的に実現可能で、部分的であっても有効であるような品質評価・保証の手法を検討することが重要となる。これは研究開発レベルの大きな課題を投げかけることになる。以下では、まだ類似の実現例が報告されていないようなものも含めて、どのような考え方で品質評価・保証の手法を検討すべきかについて論じる。

品質評価・保証においては、訓練データや訓練アルゴリズム、得られた訓練済みモデルなど、異なる成果物に対する多くの活動が含まれる。例えば訓練データをイラストレーターへの発注により作成する場合を考えると、その際にも例えば鮮明さや多様さについて仕様として言及する必要がある。一方で、そのような訓練データを用いても、得られた訓練済みモデルからの出力が鮮明かつ多様である保証はない。このため、以下では主にテスト、すなわち、構築したコンテンツ生成システムからの出力を多数検査することで、品質特性が満たされているかどうかを確認する手法を論じる。

以下に挙げる品質評価の手法のほとんどが、厳密化できない品質特性を近似的に評価するものとなる。このため、手法により「品質が悪い」と評価されるものには、実際に人が見たときに「問題がない」、あるいは「品質がよい」ものさえあるかもしれない。しかし、ランダムサンプリングで得られた少数の出力を人が確認するだけでなく、効率的に問題を発見することができるツールを追求することが重要である。そのような効果があることを実験、仮説検定を通して確認することが望ましい。

なお、対象とする品質特性のうち、例題のコンテンツ生成システム固有でなく一般性が高いものについては、評価手法が確立しており、ライブラリやソフトウェアアプリケーションとして利用可能である場合がある。この場合、そういった既存の手法、その実装を活用して、品質特性の評価を行うことが考えられる。

5.4.1 指標による評価

対象とする品質特性を、そのものでなくとも近似的に表すような定量指標を定義することにより評価する。

例：品質特性 QC01：自然さ / 品質特性 QC02：鮮明さ / 品質特性 QC03：多様さ

画像や動画の多様さについては、コンテンツ生成システムからの多数の出力に対し、鮮明・多様であるということを評価すればよい。

これは、生成対象の画像・動画での学習済識別モデルを用い、識別結果分布・特微量分布を用いて評価できる。評価指標として例えば、Inception Score [Salimans+16]、Fréchet Inception Distance [Heusel+17]、Kernel Inception Distance [Bińkowski+18]、Learned Perceptual Image Patch Similarity [Zhang+18] といった生成モデルの鮮明さ・多様さの評価指標の個別品質評価への応用を検討する。Inception Score は画像分類の学習済識別モデルを用い、各生成画像が識別モデルでどのクラスか明瞭であったり（ピークが 1 クラスのみにたつ）、画像全体では幅広いクラスに分布したりすると値が高くなる。ImageNet の 1000 クラスの多様な生成を行う場合、ImageNet の学習済みモデル、クラスが用いられる。Fréchet Inception Distance や Kernel Inception Distance は、学習済み識別モデルから算出される中間特微量分布を用い算出される、データ集合間の距離である。本物のデータ（学習データ）集合と生成データ集合の特微量分布の距離を算出し、生成データが学習データ同等に鮮明・多

様になっているかを評価する。画像評価とともに動画の評価では、動画を入力して行動種類を出力する行動分類（Action Recognition）タスクの学習済み識別モデルを用い、同等な距離算出も行われている。Learned Perceptual Image Patch Similarity は、Perceptual loss を利用した学習済み識別モデルを用い、画像パッチに対し抽出された特徴量の重み付き L2 距離で、画像パッチに対する人間の知覚的な類似性の評価に近いとされている距離指標である。この距離指標は、生成データの多様さの評価や、生成データが本物データ同等に自然・鮮明になっているかの評価に用いられる。

各個別の産業応用を行う場合、これらのスキームを応用する。画像や動画の特徴制御を行う変数に対する識別モデルを学習し活用する。例えば、生成制御したいイラスト画像・動画に対して、その種類を判別する識別モデルを学習する。その学習済みモデルを用いた Inception Score の算出により、制御対象で多様な画像・動画が生成されている評価を行うことが出来る。識別モデルの学習時に学習データ全体から学習データの選定の仕方を、部分的な選定、N 種類での選定をすることにより、モデルのアンサンブルによる出力評価により指標評価の向上も期待できる。識別モデルの学習データとしては構築された生成対象ドメイン画像を活用する。例えば、イラストレーターにより描かれた画像により構成される。あるいは、自然に Web 上などで収集できるイラストと比較することも考えられる。

例：品質特性 QC08：動画のなめらかさ

画像間の Optical Flow の統計値を算出し、動画として自然な変化量の閾値を超えた量を評価する。システムが出力した動画内の変化量が、その閾値を超えていないかチェックすることで、不自然ななめらかさを持つ動画を検出することができる。

例：品質特性 QC09：構造系列としての自然さ

身体構造上接続している器官点間の相対距離の変化量を数値化し、構造系列として自然な変化量の閾値を超えた量を評価する。システムが出力した動画内の構造系列の変化量が、その閾値を超えていないかチェックすることで、不自然な構造系列を持つ動画を検出することができる。

5.4.2 機械学習による品質評価 AI の構築

自然さなどは言葉にできない（演繹的なルールでのソフトウェア実装はできない）基準であるが、訓練データがあれば出力結果に対する品質評価 AI を実装できる可能性がある。

ドメインの差異により、既存の学習済み識別モデルを利用できないケースでは、独自に品質評価 AI を構築する。例えば、人物の姿勢推定に関しても、写実画像の姿勢推定とイラスト画像の姿勢推定では大きく異なる。もともとのコンテンツ生成システムを構築するために用いた訓練データを活用し、品質評価 AI を構築することが考えられる。姿勢・属性の評価についても、意識して事前にラベル付けをしておけば、画像や動画から姿勢・属性を推定し品質評価する AI を構築することができる。

きる。

以降ではこのように固有の訓練データセットを用いて評価 AI を構築することを議論する。なお、技術が進展しているマルチモーダルな生成 AI を活用して、画像に関する判定をより低成本で行わせることができる可能性もある。

例：品質特性 QC01：自然さ / 品質特性 QC02：鮮明さ

コンテンツ生成システムを構築するときに集めたデータは、自然に存在する画像や動画をしたものになっているはずである。別のアーキテクチャの生成モデルで、集めたデータの一部を用い、画像として自然かどうかを判断するような識別器を作る。

例えば自然さの評価 AI として、GANs 学習により、本物のデータであるかを判別する識別器を構築する、その識別器のスコアを用い画像の自然さ・鮮明さを評価する。または、画像から画像の潜在ベクトルへのエンコード・デコードに画像復元を学習したモデルを用い、生成データの復元スコアにより、自然さ・鮮明さを評価することもできる。

また、今回対象となるイラストについては、Web から収集することができる。これらに対して、明らかに問題となるであろう大きさの崩れやノイズを付加することにより、不自然な画像や動画を表すデータを作ることができる。これらの自然・不自然 2 種のデータを用いることにより、想定する不自然さを検出するような品質評価 AI を構築することができる。同様に、鮮明な訓練データに対して形状や色合いの崩れをノイズとして付加したり、解像度を落としたりした画像データを作成して訓練データとし、鮮明な画像データと分類するモデルを構築することで、不鮮明な画像を検出することも考えられる。

例：品質特性 QC04：社会的適切さ

社会的に不適切なデータを識別する AI を構築する。それを用いて、訓練データから不適切な画像や動画を排除したり、コンテンツ生成システムの多数の出力から不適切なものを検出したりすることができる。なお、同様の技術は検索エンジンなどで確立されている。

例：品質特性 QC05：指定構造との合致 / 品質特性 QC06：指定属性との合致

画像や動画に写る人物の姿勢を推定する AI の研究開発は盛んに行われてきた [Sun+19, Pavllo+19, Kocabas+19]。ユースケース UC-1A/2A においては、そのような技術を用いて、コンテンツ生成システムからの出力を評価することが考えられる。具体的には、出力における姿勢を推定し、それと入力にて指示した姿勢との距離などを測定することにより、指定された姿勢との合致度合いを評価することができる。Text-to-Image においては、プロンプトと呼ばれるテキスト形式においてこのような構造や属性の指示ができるようになっている。

例：品質特性 QC07：動画としての自然さ

QC01 と同じように、動画から数枚の画像を抜き出したときに、それが画像例として自然かどうかを判断するような識別器 [Wang+18] を作る。

例：品質特性 QC08：動画のなめらかさ

QC01 と同じように、動画から数枚の画像を抜き出したときに、画像間の Optical Flow の分布が自然かどうかを判断する識別器を作る。

例：品質特性 QC09：構造系列としての自然さ

QC01 と同じように、構造列が自然かどうかを判断するような識別モデルを活用する [Cai+18, Barsoum+18, Kundu+19]。動画から数枚の画像を抜き出したときに、その画像から抽出される構造列が自然か否かを判断する識別器を作る。

5.4.3 ルールベースの AI など他実装との比較

多少性能が悪くとも同じ機能を実装したもの、あるいは対象とする品質評価を実装したものが得られるのであれば、それらを用いることで擬似的に品質評価を行うことができる（疑似オラクル）。

例：品質特性 QC06：指定属性との合致

服の色など簡単な属性については、画像処理ライブラリを活用して画像から主要色を抽出するなど簡単な実装を行うことができる。これは必ずしも実際の「服の色」と合致しない場合があるかもしれないが、こういった簡易実装と比較することで、検査を実装することができる。

5.5 品質保証レベル

本章でここまでに挙げた品質特性は、システム全体の出力に関するものであり、ユーザが直接体験するものであるが、ほぼ直接モデルからの出力そのままがユーザに表示されることになる。これは画像や動画については、ルールベースのプログラムによる後処理でフィルタリングなどを行うことが難しいためである。このため、本章で挙げた品質特性は、2章における品質保証レベルの中で Model Robustness を対象としたものとなっている。

また、2章では、顧客の期待に対応して、データやモデル、システム、そしてプロセスの品質保証を適切に行う必要があると述べられている。本例題においては、例えば以下のように異なる利用状況に対応した品質保証レベルが考えられる。

- レベル1：システム内部挙動を知っている人の利用のために保証すべき水準。例えば、動画作成を請け負うサービスを運営し、コンテンツ生成システムの開発チームも交えた組織にて

動画を作成する場合。

- レベル 2：システム内部挙動を知らない、外部の人による利用のために保証すべき水準。例えば、コンテンツ生成システムをアニメーション制作会社に納品して利用させるような場合。
- レベル 3：悪意のある人も含めて多種多様なユーザに利用させるときに保証すべき水準。コンテンツ生成システムを不特定多数のユーザが利用できるサービスとして Web 上にて公開し、利用させるような場合。

レベル 1 とレベル 2 に対する要件の違いは、個々の出力が安定して「良い」という安定性の指向である。レベル 1 のように内部で用いる場合、出力を利用者に送る前に人手で確認をし、複数の出力から最もよいものを選んだり、出力を出し直したりするような運用方式もとりやすい。このため、各品質特性に対する評価を行う際に、複数出力において最もよいもの（最大値等）だけを評価し、品質のばらつき（分散値等）や最悪ケースはさほど問題としないということが考えられる。レベル 2 以上のように外部の利用者が用いる場合、個別の出力が安定して「良い」、少なくとも「ひどい」ものが出てない、といった安定性がより重視される。

レベル 2 とレベル 3 における考え方の違いは、仕様、特に適用範囲の定義、制限、合意のしやすさである。レベル 2 は B2B の想定であり、例えばユースケース UC-2B で画像が入力となる際に、特定イラストレーターによる画像だけが入力になるといった適用範囲を定めることが考えられる。すると、その想定の下でのみ品質保証を考えればよい。レベル 3 は B2C の想定であり、サービスの魅力の観点から適用範囲の限定が難しく、また細かい適用範囲を定めたとしても、熟読や徹底せずを利用し不満を抱くような場合が想定される。このため、より多様な入力を想定して品質保証の活動を行う必要がある。

これらのレベルに応じ、4.1.4 節で例示した各品質特性に対する評価の活動をどのように実施するかのプロセスを表 4.1 に示す。ここでは、モデルの設計・構築に伴う評価活動をフェーズ 1、品質保証のための評価活動をフェーズ 2 と分けている。それぞれは以下のように役割を位置づける。

フェーズ 1 では、モデルの設計・構築の際に評価活動を行う。すなわち、訓練・最適化を通してモデルを構築するための目的関数として、あるいはアーキテクチャや訓練手法等の評価や改善の基準として、品質特性の評価を行う。例えば、QC01-03（自然さ/鮮明さ/多様さ）は、すべてのユースケースにおいて重要な品質特性である。さらに、QC01-03 の評価指標（4.1.4.1）として挙げた Inception Score 等については、一般的に生成モデルによるコンテンツ生成の評価指標として用いられている。このため、「QC01-03 の指標による評価」という活動は、モデルの設計・構築を行う開発者が、モデルの品質を確認しながら、一定の品質が達成されるまで設計の改善や訓練・最適化を反復するために行うものと位置づけることができる。

フェーズ 2 では、ある程度のモデルが構築された後に、さらなる品質の確認、改善のための活動を行う。すなわち、モデルの設計・構築を行った開発者とは別の観点からの評価を行う。このため、モデルの本質的な目標となるような品質特性だけでなく、付加的な品質特性を評価したり、評価用

表 5.1 コンテンツ生成システムにおける品質保証のプロセスとレベル

	指針	フェーズ 1：構築（設計・訓練・最適化）を駆動するための品質	フェーズ 2：異なる観点・データから評価する品質、追加の機構を構築して評価する品質（QA・第三者含む）
レベル 1	複数出力からの選出可能性を考慮して評価	生成モデルの改善を駆動するにおける基本評価（QC01～03, 07～08 の指標評価）。 ユースケース実現の核となる機能の評価（QC05, 06 の既存 AI を用いた評価）。	付加的な品質特性に対する評価（QC04）。
レベル 2	安定性を重視して評価		フェーズ 1 で扱った品質指標について、ドメイン固有の追加データセットや、評価手段実装も用いた追加評価。
レベル 3	多様な入力を想定して評価	一般的な指標や AI 実装で可能なものを用いた評価。	

の AI や異なる評価データセットを準備して評価を行ったりする。例えば、QC01-03（自然さ/鮮明さ/多様さ）についても、対象ドメイン固有の観点を反映するように、訓練データセットとは別に評価データセットを構築したり、独自の評価 AI を構築（4.1.4.2）したりすることが考えられる。具体的には、イラストの適切な線の描かれ方、画像や動画に人物のみが含まれるか、人物と背景が含まれるか、人物の服装にどれだけの多様さを想定するか、といったドメイン固有の点を考慮した取り組みが必要となる。

以上のように、フェーズ 1 では、ユースケース上機能の根幹にかかり、ゆえにモデル設計・構築を駆動するために用いられる品質特性、一般的な指標や AI で評価できるような品質特性を扱う。フェーズ 2 では、付加的な品質特性、固有の指標や AI の構築が必要となるような品質特性を扱う。

5.6 テスト設計事例

ここまで述べた品質評価に対応するテスト設計を、画像生成システムに対して適用した事例を示す。対象となるシステムとしては、SPADE [Park+19]^{*1}を用いる。

5.6.1 対象システムの概要

SPADE は以下の 2 つの入力を受け取り、その入力に応じた画像を生成する。

ラベルマップ 画像のうちこの部分は「空」、この部分は「木」など、領域とその領域にある物体の

*1 <https://nvlabs.github.io/SPADE/>

対応を表したもので、生成される画像内のオブジェクトの配置を指定する。

スタイルガイド画像 参考となる画像により、生成する画像のスタイルを指定する。例えば夕焼けの画像を指定すると、生成画像においては空に赤い太陽が低く写っていたり、木や地面は暗く影になって見えたりする。

図 5.9 に SPADE からの出力例を示す。列がラベルマップに、行がスタイルガイド画像にそれぞれ対応しており、それらの組み合わせにより異なる画像が生成されている。

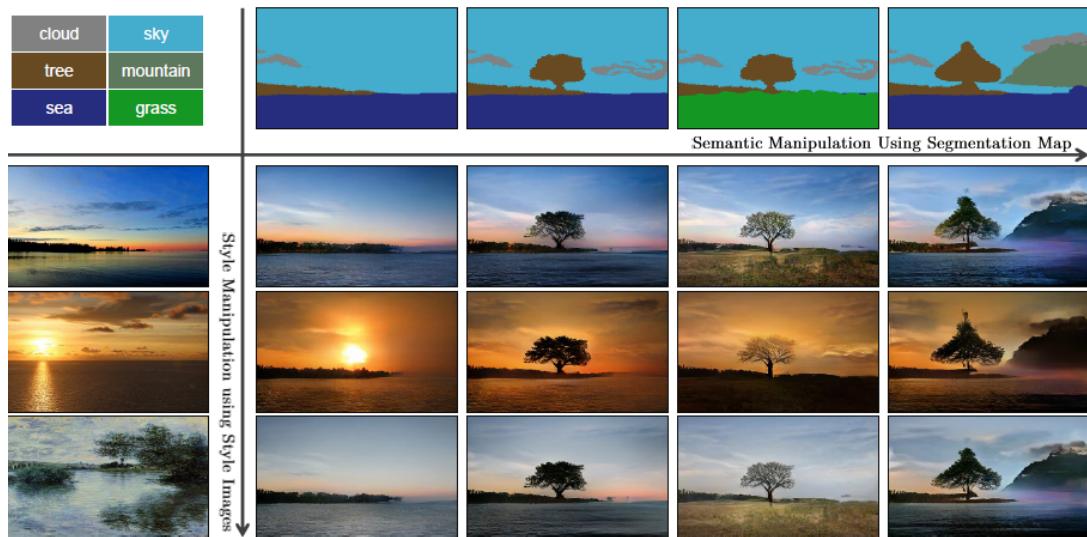


図 5.9 SPADE の出力例 [Park+19]

5.6.2 対象となる品質特性

5.3 節では、画像や動画の生成を行うシステムに関する品質特性を論じた。QC01（自然さ）をはじめとして、これらの品質特性の多くは、画像・動画生成システム全般に共通する一般的なものである。また QC05（指定構造との合致）や QC06（指定属性との合致）については、「構造」や「属性」が何を指すかについて対象システムごとの具体化が必要となる。

SPADE に対するテスト設計事例においては、まず 5.3 節で定めたものと同じものとして、QC01（自然さ）および QC02（鮮明さ）を扱う。次に SPADE に特化し具体化した品質特性として以下が挙げられる。

QC05'（指定ラベルマップとの合致） ラベルマップで指定されたオブジェクト配置を遵守している。

QC06'（指定スタイルガイド画像との合致） スタイルガイド画像で指定されたスタイルを反映している。

ここで QC01（自然さ）および QC02（鮮明さ）に対する具体的な問い合わせとして、ラベルマップおよびスタイルガイド画像からなる入力のうち、想定される多様な入力に対して、高い品質が安定して実現されるかという問い合わせられる。例えば、運転シーンに関する画像を SPADE で生成し、自動運転に関する AI の訓練あるいはテストに利用することを考えてみる。このとき、入力となるラベルマップの指定によって、歩行者がとりうる位置を系統的に網羅するような画像一式を生成することが考えられる。同様に、スタイルガイド画像によって、様々な天候状況や時間帯に対応し明るさや色合いが異なる画像一式を生成することも考えられる。こういったユースケースを想定すると、SPADE についても入力がある基準で網羅的に変えながら系統的にテストを実施することが重要であろう。また、識別モデルに対して敵対的サンプルとして盛んに議論されているように、入力の微小な変化で出力が大きく変化する可能性も否定できない点からも、入力を系統的に変化させてのテストは有効であると考えられる。

5.6.3 テスト設計

SPADE に対するテストを実施するためのアーキテクチャ例を図 5.10 に示す。SPADE の入力となるラベルマップあるいはスタイルガイド画像に対して系統的に一定のバリエーションを用意し、それらに対して SPADE の出力を評価する。

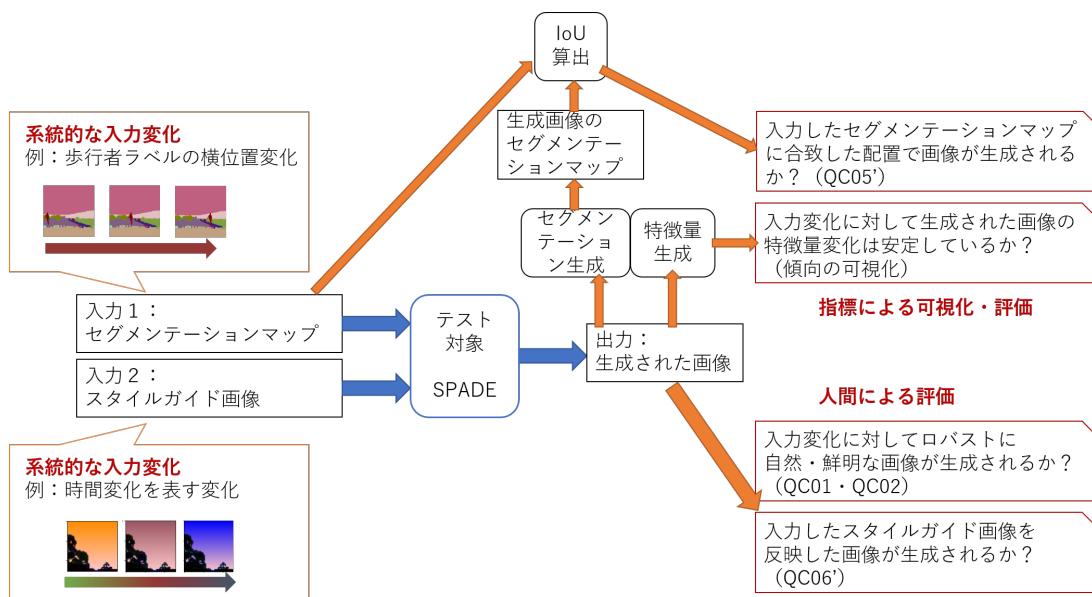


図 5.10 SPADE に対するテストのためのアーキテクチャ設計

一連の入力に対して生成された画像列を、QC01（自然さ）および QC02（鮮明さ）の観点から評価する。基本的には人手での目視確認を行うが、補助的な指標として、一般的な画像特徴量を取得し、入力画像の変化に対して大きな変化がある点を把握できるようにもする。QC05'（指定ラベル

マップとの合致)については、SPADEにより生成された画像からラベルマップを生成し、これを入力のラベルマップと IoU 指標 (Intersection over Union) を用いて比較する。IoU 指標は、2つのラベルマップ上の各クラスに対して、領域の重なり度合いを評価する指標である。QC05' に関するこの比較手法は、SPADE だけでなく、セマンティックセグメンテーションツール自体の影響も受けることに留意する。QC06' (指定スタイルガイド画像との合致)については、スタイルの合致を判断するような処理の自動化も検討しうるが、ここでは目視確認を行うことを想定している。

テストの入力のうちラベルマップについては、人間の手で描画すると多数生成することができない。すでにラベルマップを含むデータセットを用いるか、既存画像からラベルマップを生成するかのいずれかにより元となるラベルマップを得て、それらを系統的に加工する（例えば歩行者の位置を変化させる）ことが現実的であろう。スタイルガイド画像については、既存画像を朝焼けや逆光、曇りなど分類することになる（自動運転ドメインではしばしば議論される属性である）。空の色や明るさを人工的に加工することで、少しずつ暗くなり夜になるといった画像列を構築することも考えられる。

5.6.4 実験

5.6.3 節で述べた設計に従い、SPADE のテストを実施した。SPADE のバージョンとしては 2020 年 12 月に公開されていたもの（2019 年 10 月 18 日のバージョン 1a687ba)^{*2}を用いた。セマンティックセグメンテーション手法の実装としては、DeepLab v2 の訓練済みモデル^{*3}を用いた。

以下では、2つの入力のバリエーションに関するテスト実施結果について述べる。前述したような SPADE の品質をあらゆる観点から完全に評価することを実施したわけではなく、テストの有効性を確認したいいくつかの事例についてのみ示す。

第一に、入力となるラベルマップ内において、歩行者の位置を系統的に横にずらしていく場合の実施結果例を図 5.11 に示す。図の上段には、入力となるラベルマップにおいて、歩行者の位置を左右に系統的に動かしたものを見ている。下段には、これらの入力に対して SPADE が生成した画像を見ている。生成された画像は、位置によらずに歩行者であることが十分認識できるため一定の自然さ (QC01) と指定構造との合致 (QC05) を保っていると言ってよいと言える。歩行者の服装変化などは起きているが自然な範囲である。また背景等の他オブジェクトと比較しても同等の解像度となっているため、鮮明さ (QC02) も保っていることがわかる。IoU 指標についても測定を行ったが、位置によって大きく変化するようなことは見られず安定して一定の値を保っていた。オブジェクト同士が重なり合って配置される場合に IoU 値がやや低下したが、これはセマンティックセグメンテーション手法の影響であると考えられる。これらの結果により、この入力画像について歩行者の位置に関する一定の頑健性を示すことができた。実際には大量の画像について同様な評価

^{*2} url`https://github.com/nvlabs/spade/`

^{*3} `https://github.com/open-cv/deeplab-v2`

を行うことになるが、ここで試行としてはそこまでは扱わない。他にも車道の自転車の位置を画像上で上下に動かす（自車から見て奥や手前に動かす）ことも実施したが、結果は同様であったために詳細は割愛する。

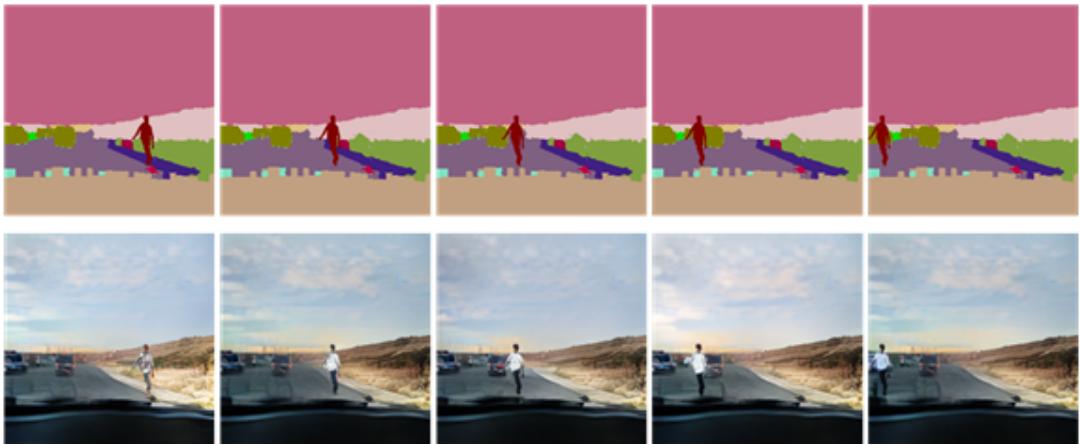


図 5.11 SPADE に対するテスト実施結果例（上段におけるラベルマップの系統的な変化に対して生成された画像が下段）

第二に、入力となるスタイルガイド画像において、時間帯が変わっていき空の色が変わっていく場合の評価を示す。ここでスタイルガイド画像では、元の画像から空の部分を人工的に着色し、その色を昼らしい色から夜らしい色に変えていった。図 5.12 に、人工的に与えたスタイルガイド画像の変化と、それに対する出力画像の変化の例を示す。出力画像においては太陽の位置が変化しており、時間帯の変化を反映させることができている。このようにスタイルガイド画像を系統的に変えていくテストを行った結果、ある色合いの付近で生成画像が大きく崩れる場合があった。この付近のスタイルガイド画像と生成画像を図 5.13 に示す。スタイルガイド画像の変化は人間の目には見て取れないほど微少なものもあるにもかかわらず、左から 4 列目、6 列目において生成画像が崩れている。この不安定な挙動はこの先も続いている。このことから、本テスト設計事例で扱う 4 品質特性 (QC01, QC02, QC05, QC06) すべてを満たしていないと言える。なお、画像の特徴量を監視することでも、該当するスタイルガイド画像付近では、明らかに特徴量が不安定に変化していた (OpenCV ライブラリで実装された SIFT, AKAZE, ORB の三指標で確認)。系統的なテストを行った結果、スタイルガイド画像の変化に対して頑健ではない事例を検出することができた。

5.6.5 テスト設計事例のまとめ

ここでは SPADE を対象として系統的にテストを行った事例を示した。入力が画像やラベルマップなど複雑な情報をもつデータである場合、ある観点を系統的に網羅するように既存データを集めることは困難であることが多い。このため既存データの加工も交えて対応している。系統的にテス

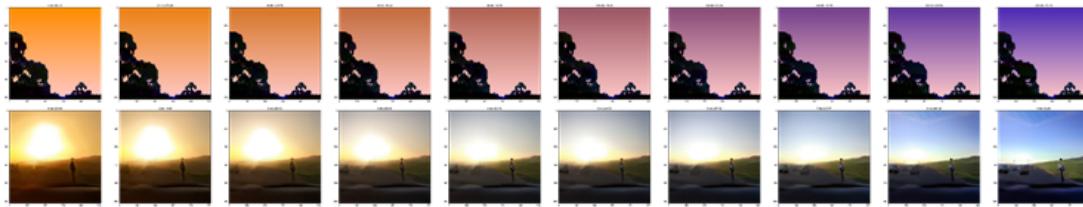


図 5.12 スタイルガイド画像に対する SPADE の出力変化例（上段におけるスタイルガイド画像の系統的な変化に対して生成された画像が下段）

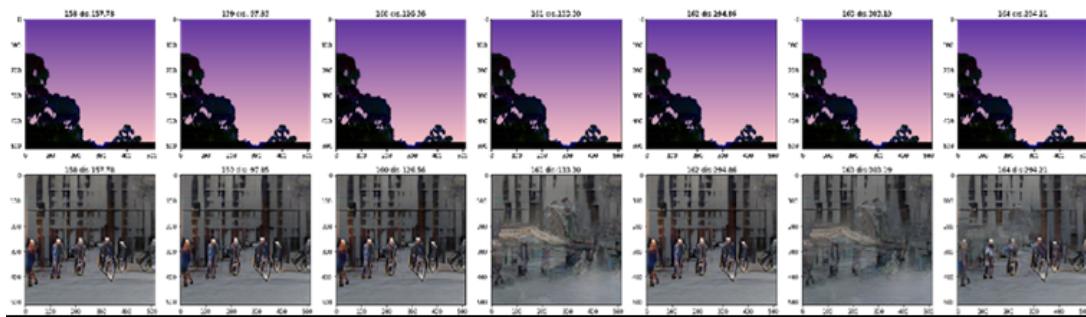


図 5.13 SPADE に対するテスト実施結果例（上段におけるスタイルガイド画像の系統的な変化に対して生成された画像が下段）

トを実施してみると、頑健性について一定の確認ができることもあれば、具体的な問題が見つかる場合もある。問題の修正をどのように行うかは、GAN による生成モデルに限らず深層学習全般において大きな課題であるが、具体的な問題やその傾向を把握することが重要であると考える。本事例ではその実例を示すことができた。

5.7 Text-to-Image に対する評価事例

謝辞

5.6 節で示した実装結果については、国立情報学研究所トップエスイープログラムにおけるソフトウェア開発実践演習において取り組んだ結果を掲載させていただきました。本ガイドラインに即した演習に取り組んでいただき、結果をご提供いただいた飯島久典様（富士通株式会社）、及川裕之様（東芝デジタルソリューションズ株式会社）、笠井栄良様（ソニー株式会社）、小御門道様（株式会社富士通研究所）、呉隆司様（株式会社 NTT データ）、鷹野翔様（株式会社デンソー）に感謝いたします。

5.8 参考文献

- [Goodfellow+14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In NIPS 2014.
- [Karras+18] Tero Karras, Timo Aila, Samuli Laine, Jaakko Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In ICLR 2018.
- [Brock+19] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In ICLR 2019.
- [Karras+19] Tero Karras, Samuli Laine, and Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. In CVPR 2019.
- [Hamada+18] Koichi Hamada, Kentaro Tachibana, Tianqi Li, Hiroto Honda, and Yusuke Uchida. Full-body High-resolution Anime Generation with Progressive Structure-conditional Generative Adversarial Networks. In ECCV Workshop 2018.
- [Hamada+19] 濱田晃一, 李天琦. AI によるアニメ生成の挑戦. In DeNA TechCon 2019. <https://www.slideshare.net/hamadakoichi/anime-generation>
- [Hamada+20] Anime Generation with AI. DeNA. 2020. <https://www.slideshare.net/hamadakoichi/anime-generation-ai> (Generated Anime: <https://youtu.be/X9j1fwexK2c>)
- [Salimans+16] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen. Improved Techniques for Training GANs. In NIPS 2016.
- [Heusel+17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Sepp Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In NIPS 2017.
- [Russakovsky+15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein. ImageNet large scale visual recognition challenge. In IJCV 2015.
- [Odena+17] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional Image Synthesis with Auxiliary Classifier GANs. In ICML 2017.
- [Miyato+18] Takeru Miyato and Masanori Koyama. cGANs with projection discriminator. In ICLR 2018.
- [Zhang+18] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In ICML 2018.
- [Wang+18] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, Bryan Catanzaro. Video-to-Video Synthesis. In NeurIPS 2018.
- [Cai+18] Haoye Cai, Chunyan Bai, Yu-Wing Tai, Chi-Keung Tang. Deep Video Generation, Predic-

tion and Completion of Human Action Sequences. In ECCV 2018.

[Barsoum+18] Emad Barsoum, John Kender, Zicheng Liu. HP-GAN: Probabilistic 3D human motion prediction via GAN. In CVPR 2018.

[Bińkowski+18] Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, Arthur Gretton. Demystifying MMD GANs. In ICLR 2018.

[Zhang+18] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In CVPR 2018.

[Kundu+19] Jogendra Nath Kundu, Maharshi Gor, R. Venkatesh Babu. BiHMP-GAN: Bidirectional 3D Human Motion Prediction GAN. In AAAI 2019.

[Sun+19] Ke Sun, Bin Xiao, Dong Liu, Jingdong Wang. Deep High-Resolution Representation Learning for Human Pose Estimation. In CVPR 2019.

[Kocabas+19] Muhammed Kocabas, Salih Karagoz, Emre Akbas. Self-Supervised Learning of 3D Human Pose using Multi-view Geometry. In CVPR 2019.

[Pavllo+19] Dario Pavllo, Christoph Feichtenhofer, David Grangier, Michael Auli. 3D human pose estimation in video with temporal convolutions and semi-supervised training. In CVPR 2019.

[Park+19] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic Image Synthesis with Spatially-Adaptive Normalization. In CVPR 2019

6. Voice User Interface (VUI)

6.1 想定するシステム

この領域では、話し手の言葉を聞き取り、聞き取った文章の意図を解釈し、話し手が意図した動作を実行するシステム、いわゆる「Voice User Interface (VUI)」を想定している。

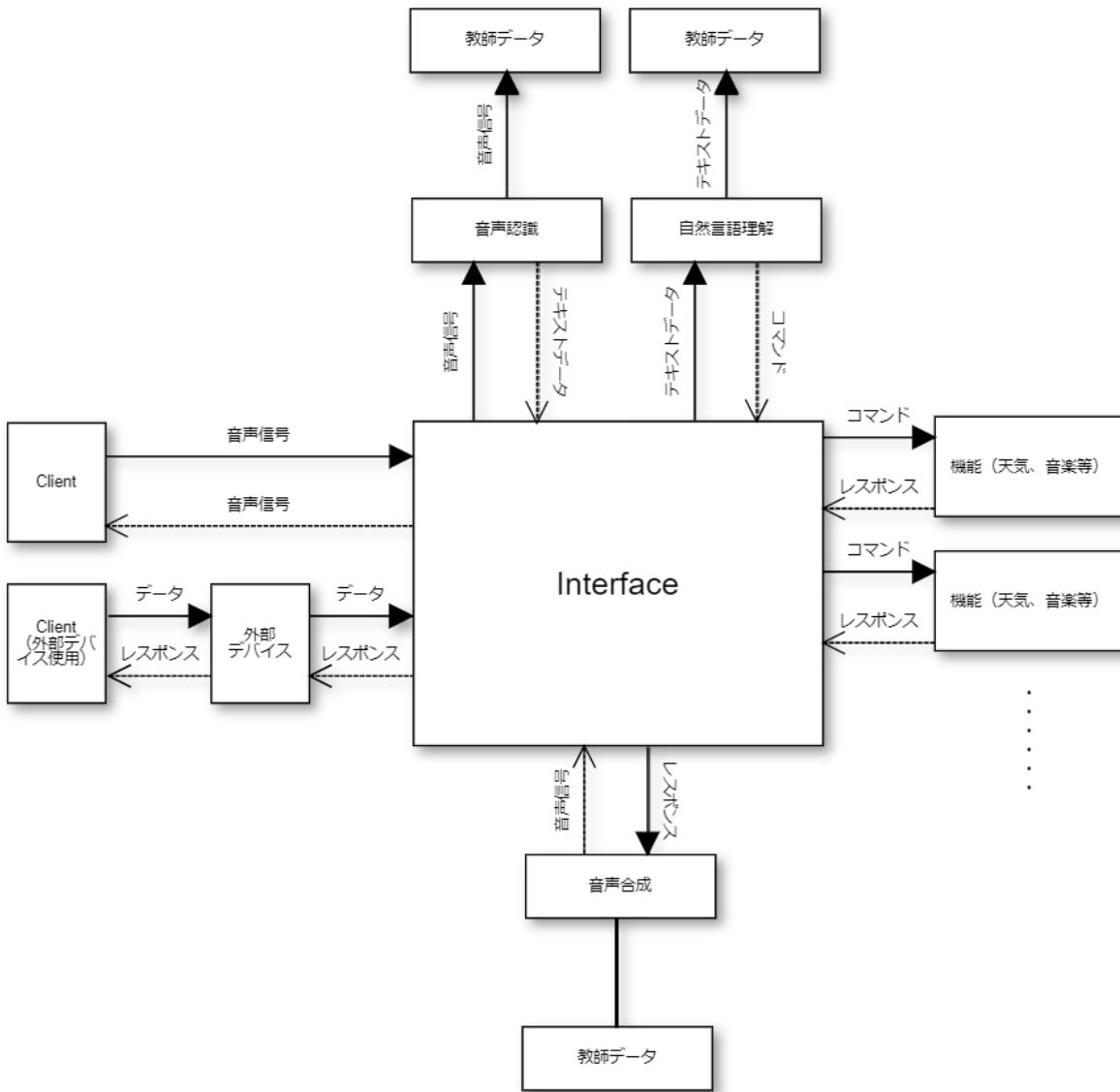
システムには以下の 3 つの機械学習が用いられていると想定する。

- マイクから入力された音声信号をテキストに変換する「音声認識」に用いられる機械学習
- 変換されたテキストが何を意図しているのか解釈し、機能動作を行うためのコマンドに変換する「自然言語理解」に用いられる機械学習
- テキストを音声信号へと変換する「音声合成」に用いられる機械学習

システム動作の流れは以下を想定している。全体像を図 6.1 に示す。

1. システムに音声信号を入力
2. 入力された音声信号が音声認識機能へ渡されテキストとして出力
3. テキストが自然言語理解機能へ渡され、目的とする機能を動作させるコマンドとして出力
4. コマンドに応じた機能（天気取得機能、音楽取得機能など）にコマンドを入力
5. 該当機能にてデータ処理
6. 該当機能からレスポンスを出力
7. 該当機能からのレスポンスの入力
 - 音声合成機能への入力の場合
 - i. 音声合成機能に該当機能からのレスポンスを入力
 - ii. 音声合成機能から音声信号として出力
 - 外部デバイスへの入力の場合
 - i. 外部デバイスに該当機能からのレスポンスを入力
 - ii. 外部デバイスによるレスポンスの処理
 - iii. 外部デバイス機能の実行

図 6.1 想定するシステム動作の流れ



6.2 VUI システムの特徴

6.2.1 音声認識

音声認識部分については「同じ言葉での音声入力だった場合、同じ文字列として変換できること」が求められる。例えば同じ「今日の天気は？」という音声入力でも以下のような条件により入力される音声信号は変化する。

- 性別（男女など）
- 年齢（高低）
- トーン（アクセント、早口、声色、関西圏/東京圏のイントネーション）
- 言葉の区切り位置（「天気は？/天気、は？」など）
- 感情（「優しい/厳しい」など）
- 母国語外である言語学習者による発音

また、音声認識については「システム利用者環境が製品保証範囲内であれば正しく動作すること」も求められる。以下の条件が考えられる。

- 環境
 - 音声環境
 - * 音声ノイズ（テレビ、話し声など）
 - * 雑音ノイズ（走行中の騒音、生活音など）
 - 設置環境
 - * 振動（足場の不安定さによる揺れなど）
 - * 閉鎖（反響による影響、窓からの反響など）
- ユースケース
 - 発話距離

音声認識についてセキュリティの面から「特定の人物以外の音声を認識してはならない」ことが求められる場合もある。その場合は以下の条件が考えられる。

- 話者認識
 - 個人の特定

6.2.2 自然言語理解

自然言語理解部分については「異なる表現でも、標準的な文章として理解できること」が求められる。ここでは日本語を取り上げ考える。以下の条件が考えられる。

- 口調（敬語、命令語、若者言葉など）
- 助詞（「てにをは」のばらつきなど）
- 文法（語順の変化、体言止めなど）
- 略語（「こいばな」といった略語、歌手名の略称、地名の略称など）
- 同音異義語（「雨/飴」、「みたい/見たい」、同音の地名など）
- 固有名詞（地名、名前など）
- 和製英語（ノートパソコン、コンセントなど）

- 流行語（ばえる、モヤるなど）
- 方言（関西弁、津軽弁など）

6.2.3 音声合成

音声合成については「システムの利用者が理解できる音声メッセージを出力すること」が求められる。ここでは日本語を取り上げ考える。以下の条件が考えられる。

- 発音（漢字の読み方。「ついたち/いちにち」、慣用句、有名人など）
- トーン（アクセント、感情、声色など）
- 話し方（文章の区切り位置など）
- 情報（速度、「長すぎる/短すぎる」など）
- 声質（男性、女性、声優など）

6.2.4 その他 - インフォテインメント

インフォテインメントはインフォメーションとエンターテインメントを組み合わせた造語であり、情報の提供と娛樂の提供を行う要素、またはシステムのことである。

VUI システムは機能実行の他に情報提供、会話といったインフォテインメントを求められることも多い。そのことから会話の多様性や、面白さといった非機能である特性も挙げられる。

6.3 特有の課題

6.3.1 システムの課題

VUI システムの共通の特徴として、音声入力のみが入り口であるため、音声認識が失敗してしまうと自然言語理解機能も正常に機能せず、意図しない出力に直結する問題がある。この問題は、例えば音声認識が失敗し、ニュアンスが近いが異なる言葉が自然言語理解機能に入力されても、ある程度予測されているような失敗であれば自然言語理解機能で処理するような対応が考えられる。

以下に VUI を用いたシステムとしてスマートスピーカー及びカーナビゲーションシステムを例に課題を挙げる。

スマートスピーカーの製品の大きな特長として、音声入力のみという「入口が一つ」に対して天気・音楽・占いといった独立した数多くの機能が実行できる点が挙げられる。それがスマートスピーカーの特徴だが、多種多様の機能があるためターゲットユーザーの特定が困難になる問題も発生する。ターゲットユーザー特定の困難さは音声認識機能の学習データ量およびその選択に関わる問題である。また、機能追加や変更の際に音声コマンドが既存機能と似通っていることにより誤認識の確率が上がる可能性もある。機能追加、変更の際は全体への影響を考慮する必要がある。

カーナビゲーションシステムの場合は車両内の利用が主となる。走行中の車両の場合、走行中の雑音が音声認識に影響を与える。よって、音声と雑音の識別の前処理が重要となる。

また車両内は閉鎖空間であるため音声の反響が発生する。結果、発言がドライバーのものなのか、後部座席の子どものものなのかといった、音声がどこから発せられたかの判別が難しい。これは「どの音声コマンドを優先すべきか」を判断するための処理が重要となる。

走行中の車両であることを考えた場合、話す側の言い間違え、音声合成側の言い間違えが事故につながる可能性がある。話す側の言い間違えについては、言い間違えを正しく解釈し、意図とは異なる場所への誘導される可能性がある。また音声合成側の言い間違えは、例えば地名「弘前（ひろさき）」を「ひろまえ」と読み上げられた場合、ドライバーが気をとられる可能性、画面へ目を向けてしまう可能性があり、事故の切っ掛けとなりかねない。

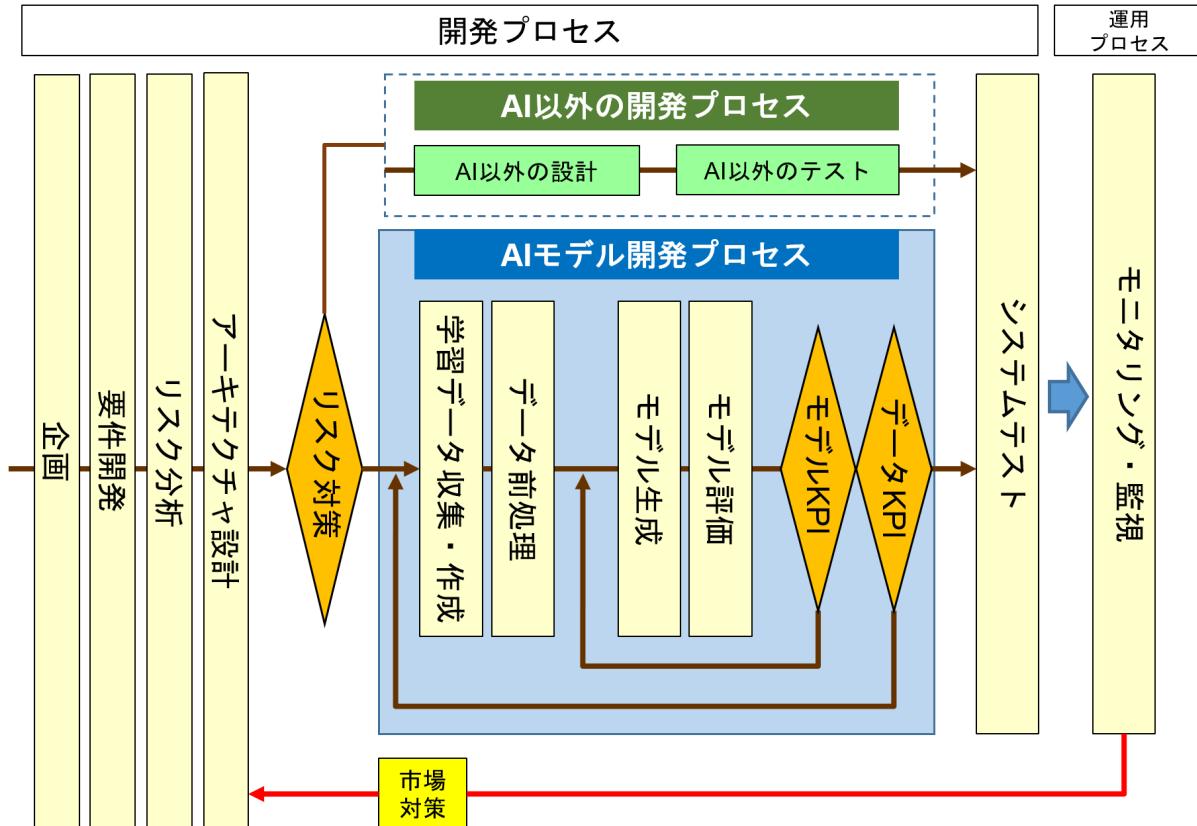
車両が移動していることもカーナビゲーションシステムの大きな特徴である。例えば「近くのコンビニエンスストアを教えて」と言われたとして、車両は移動しているため、数メートル先のコンビニエンスストアの場所を出力しても止まることは難しい。もちろん近くても進行方向と逆の方向のコンビニエンスストアを出力されても行くことが難しい。しかし前方のコンビニエンスストアまで遠い場合は U ターンした方が良い場合もある。これらは自然言語理解後のシステム側の処理で「近く」をどう処理するか対応する必要がある。

6.3.2 Process Agility の課題

スマートスピーカーにおいては、新しい言葉が出たならばその言葉に対応することや、ユーザーが使い始めた結果、よく使われる言葉に対応するといったことが頻繁に発生しうる。自動車などと異なり、頻繁なアップデートが行われることが特有の問題である。そのアップデートに対応するための開発・運用プロセスが必要であると考えられる。

図 6.2 頻繁なアップデートが必要な AI を組み込んだシステムの開発・運用例

頻繁なアップデートが必要なAIを組み込んだシステムの開発・運用例

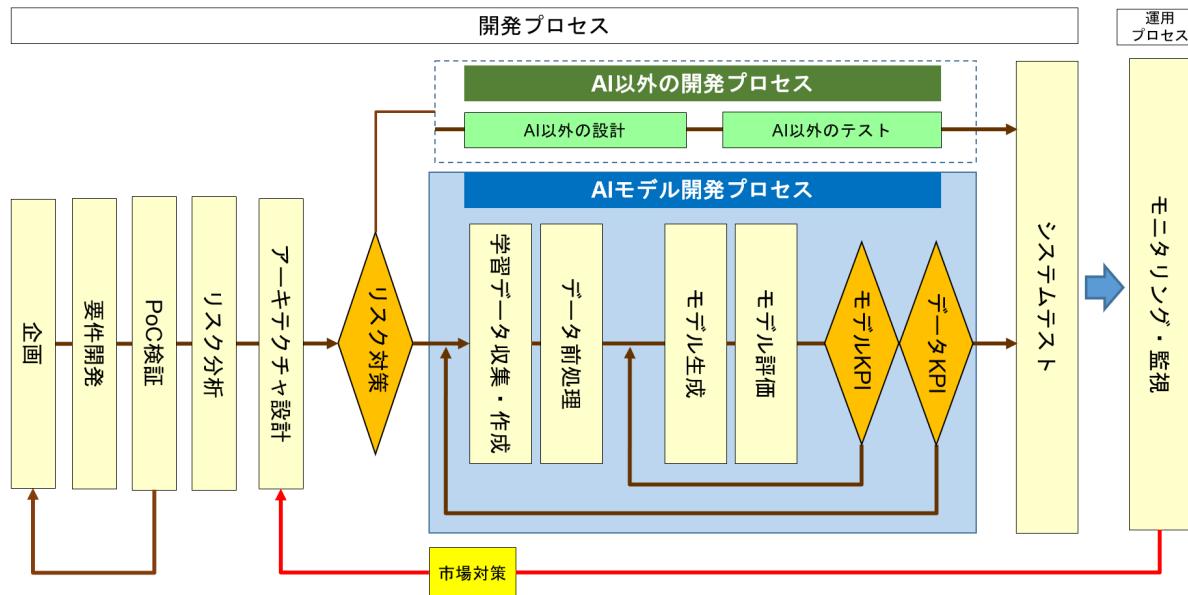


Process Agility を達成する一例として図 6.2 を挙げる。リリース後はモニタリング・監視により言葉のニーズや問題をキャッチアップする運用体制が必要不可欠である。そこからそれら問題点を拾い上げ、モデルへと反映させ、リリースする。そこからまたモニタリング・監視を行う、といった形で反映とモニタリングの循環を生み出せる体制である必要がある。

カーナビゲーションシステムのアップデートについてはスマートスピーカーと違い、頻繁に行うことは難しい。このようなシステムの場合、初期の段階において PoC 検証を行い、市場での使われ方や発話パターンを十分に検討しなければならない。その上で開発、リリースとなる。このようなシステムにおいてもリリース後のモニタリング・監視による想定した発話パターンとのずれのキャッチアップは発生する。これら問題点を拾い上げ、モデルへと反映、アップデートする運用体制を構築は考慮が必要となる。このシステムの場合の Process Agility を達成する一例を図 9.2 に挙げる。

図 6.3 頻繁なアップデートができない AI を組み込んだシステムの開発・運用例

頻繁なアップデートができないAIを組み込んだシステムの開発・運用例



6.4 VUI における機能安全

6.4.1 セーフティ

VUI を備えたシステムは話しかけることによって機能するため、気軽に、簡単に動作させることができる。便利ではあるが、その反面セーフティについて十分に注意を払う必要がある。

スマートスピーカーは発話だけで商品購入が可能となる機能を備えているものもある。この機能においては簡単に商品購入ができるが、言い間違いや音声認識の誤動作により意図とそぐわない商品や数量の購入になってしまう場合がある。またコマーシャルの声に反応してその商品が購入されてしまった事例もある。この様な機能の場合は、購入ステップを増やす、確認機能の強化といったように仕様策定の際に簡単に動作し利用者に不利益を与えないように考えなければならない。

カーナビゲーションシステムの場合は、使用される場面が運転中が主であることに注意しなければならない。例えば右折する際に、交差点の直前で情報が伝えられても右折のレーンに入ることが出来ない場合も多く、またその指示に反応し急に車線を変えようとして事故になる可能性もある。また伝える情報量も重要である。運転中に情報を一度に大量に伝えられても運転の集中が阻害されてしまう。

意図しない発話が意図しないタイミングで起きた際に安全性を優先することが品質基準となる。

6.4.2 外部機器の操作

VUI を用いて外部機器を操作する機能を持つ場合も多い。しかし、エアコンや車の各部位の操作といった品質事故に対して厳しい確認を行う必要がある製品を扱う場合は、音声の誤認識が大きな事故を引き起こす可能性がある。そのような問題を防ぐためには以下が考えられる。

- 代行手段の用意
- 誤認識を防ぐ仕様の定義

代行手段については、VUI は音声での操作のみであるため、安全面を考慮し別の制御手段の用意についてである。緊急で外部機器の操作を止めたいたとき、例えば車の窓を閉めている際に子どもが手を挟んでしまった場合など、音声認識では反応が遅れてしまうこともある。こういった機器を取り扱う際は、安全面から機器の直接制御の考慮が必要となる。

誤認識を防ぐ仕様については、よく用いられる方法は音声での命令に対し、訊き返し確認を取ることによって誤動作を防ぐ方法である。複数の同じ種類の機器の操作（例：照明 1、照明 2）や、似たようなクエリで操作をする場合は誤認識が多い。これらの場合は誤動作を防ぐため、それぞれにユニークなラベルを付け、そのラベルでの操作を行う仕様が考えられる。車のリクライニングの操作などは、走行中には操作は不能にするといった仕様が安全面で非常に重要である。

上記以外にも、VUI 機器側、接続される外部デバイスマーカー側の取り決めやガイドラインの制定も思わぬ事故を招かないための品質基準として挙げることができる。

6.5 期待される品質

6.5.1 5つの軸から見た期待される品質

VUI システム共通で期待される品質は「音声入力が話し手の意図通り認識され、意図した機能が実行されること」である。

そのためには、様々な音声で同じ動きをすることが Data integrity の観点として挙げられる。様々な音声とは 6.2.1 の音声認識機能にて挙げた、異なる性別、異なる年齢、異なるトーン、異なる感情、異なる言語学習者の発音である。同じ動きを行うためには、音声認識機能から自然言語理解機能にテキストが渡される際に適切なテキストに変換される必要がある。よって先に挙げた要素からプロダクトに必要な要素を決定し、それぞれに対して合格基準を設けることがよいだろう。

また同様に様々な表現で同じ動きをすることも Data integrity の観点として挙げられる。様々な表現とは 9.3.4 の自然言語理解機能にて挙げた口調、助詞、文法、略語、同音異義語、和製英語、流行語それぞれの要素である。言葉の表現の組み合わせは爆発的になるため、どこまでの認識を保証するか範囲を決め、その決められた範囲内で品質を確保することがよいだろう。

結果を出力する際の要件として読み間違えず音声メッセージを伝えられることも Data integrity の

観点として挙げられる。9.4.2 の音声合成機能にて挙げた発音、トーン、話し方、情報それぞれの要素である。発音に関しても全ての言葉を網羅することは現実的ではない。保証すべき範囲をプロダクトの使われ方から限定し、その範囲内で品質を確保することがよいだろう。

また日々新しい言葉が生まれている。出荷したときに学習されていない言葉も出荷後に数多く現れると考えられる。Model Robustness の観点として、これらを素早くキャッチアップしつつ早く対応、アップデートしていく体制作りも必要となるであろう。

VUI を含んだシステムは、どのような場面で動かされるか想定し、その場面での使用方法の品質を考慮することが、System Quality の観点から重要である。例えばスマートスピーカーの場合は、家庭に置かれるため家の生活している場面での使用が考えられる。そのように考えた場合考えられる要件は、ノイズによって認識が阻害されないことが挙げられる。製品を設置した箇所によってはテレビや他者の話し声があることもある。カーナビゲーションシステムの場合は走行中での使用が主となる。走行中のノイズや振動、狭い空間内の反響の影響を受けないことが挙げられる。6.2.1 の音声認識機能のシステム環境にて挙げた音声ノイズ、雑音ノイズ、置き場所による揺れ、反響の要素に対してプロダクトが必要となる要素をピックアップし、合格基準を設け確認することがよいだろう。

VUI は「声」により大きく Customer Expectation が左右される。これまでの GUI では見やすく使いやすい画面が重要とされてきた。VUI の場合は、User Interface が画面ではなく声である。この声について、VUI を使用するターゲット層や製品イメージによって求める期待値が変わってくる。男性的な声が良いのか、中性的な声が良いのか、女性的な声が良いのか、ニュースキャスターのような感情を含めない話し方が良いのか、ドラマやアニメーションのように喜怒哀楽の起伏がついた話し方が良いのか。これらもターゲット層の期待や製品イメージとかけ離れていないか品質保証の対象となる。声が製品イメージにふさわしいかどうかの確認は、音声合成の教師データ収集といった作業があるため、仕様策定から開発初期段階といった早い段階で行われ決定されていることが望ましい。

スマートスピーカーにおける Customer Expectation は、ターゲットユーザーを明確に定めない特性上、広範囲に渡っている。したがって、天気や音楽等、各機能毎にターゲットユーザーを定めて、Customer Expectation を満たす基準を検討する必要がある。Data integrity, Model Robustness, System Quality においては、スマートスピーカーの特性を考慮する必要がある。

カーナビゲーションシステムにおける Customer Expectation は、走行中の運転手の注意を不用意に逸らさずにナビゲーションすることが求められることの一つである。例えば 2 車線で違う車線に入るときのレコメンドは早めのタイミングや赤信号で停止中に行うなど、人によるナビゲーションのような自然な形で、運転中の運転手の注意力を阻害しないようなナビゲーションが行われているかが望まれる。

6.5.2 音声出力の UX

UX とは User eXperience の略称であり、ユーザーが製品やサービスを通じて得られる使いやすい・使いにくいといった印象や、楽しい・楽しくないといった感情などのユーザー体験を指す。

VUI の UX には入力部分と出力部分があるが、入力の UX についてはこれまで記載した内容と重なることが多い。よってこの項目では音声出力部分に焦点を当てる。

VUI の音声出力部分においては機械学習が用いられた音声合成部分と、それらに関連するシステムにわけられる。以下に列挙する観点により UX が左右される。また表での分類は例であり、音声合成に記載した観点について機械学習ではなく周囲のシステムで調整している場合や、その逆の場合もある。

それぞれの観点において何を適切とするかは、プロダクトの利用環境やターゲットユーザーによって左右される。社内でのユーザーテストや、実際に利用を想定しているターゲットユーザーによるユーザーテストで判断することが一例として考えられる。

表 6.1 音声合成の観点

分類	観点	概要
音声合成	流暢さ	よどみなく滑らかに話せるか
	高さ	聞き取りやすい声の高さか
	抑揚・感情表現	声の抑揚や感情表現がプロダクトの性質に合わせ適切に用いられているか

表 6.2 音声合成に付随するシステムの観点

分類	観点	概要
音声	音量	初期値の音量が適切か
	正確さ	発音が正しいか
	速さ	聞き取りやすい速度か
	句読点の待ち時間	句読点での待ち時間は適切か
内容	文章構成	情報を把握しやすい文章構成か
	長さ	メッセージが長すぎないか
性能	レスポンスタイム	音声入力してから音声が返されるまでの時間は妥当か
機能	音声の選択	利用環境や個人の好みに応じて声の選択ができるか

- 音声合成の観点
 - 流暢さ

よどみなく滑らかに話せるか。特別な理由がない限り、自然であればあるほど聞き取りやすく UX は良いものとなる。

- 高さ

出力される声は聞き取りやすい声の高さか。年齢が高いユーザーの場合、高音域が聞き取りにくい場合がある。他、高すぎた場合「キンキンした声」となりユーザーにストレスをあたえる。プロダクトが想定する利用ユーザー、利用環境に合わせた声の設定を考える必要がある。

- 抑揚・感情表現

声の抑揚や感情表現がプロダクトの性質に合わせ適切に用いられているか。これは機能がコミュニケーションに重きを置くものかどうかで考慮する。例えば家電の ON/OFF や、情報を伝えることが目的である場合、音声に抑揚や感情をのせると本来伝えたい情報のノイズとなりユーザーにストレスを与える可能性がある。コミュニケーションやエンターテインメントが目的である場合は、声が平坦では無機質な印象をユーザーに与えてしまう。こういった機能では、声に抑揚や感情をのせたほうが「人と話しているような感覚」をユーザーに与えることができる。

● 音声合成に付随するシステムの観点

- 音声

* 音量

初期状態での音量が適切か。初期値の音が大きすぎる、小さすぎる場合、音量調整は可能であるがユーザー体験を損ねる恐れがある。

* 正確さ

読み上げる単語、文章の発音が正しいか。単語の読み方が違う、「橋」「箸」のようにアクセントが違う場合、音声だけでは字が見えないため違う文意になり意図した情報が伝わらないことが起こる可能性がある。

* 速さ

文章を読み上げる速さは適切か。音声出力の場合は文字出力と異なり情報が流れてしまう。速すぎると情報を聞き洩らす恐れがある。遅すぎる場合は求める情報が出るまでに時間がかかりユーザーにストレスがかかってしまう。

* 句読点での待ち時間

文章に句読点があった際の待ち時間は適切か。句読点での待ち時間が短すぎた場合、文章の区切りがわからなくなり文意が間違って伝わってしまう場合がある。句読点での待ち時間が長すぎる場合は、違和感や、文章読み上げが止まったようにユーザーに勘違いさせてしまう。

- 内容

* 文章構成

情報を把握しやすい文章構成か。最初に結論を伝え、後に説明といったように、ユーザーが情報を把握しやすい構成で出力することが望ましい。

* 長さ

出力される文章の長さは適切か。音声出力の場合、長々としたメッセージはそれを聞いているユーザーにストレスを与える。質問に対して核となる情報を短い文章で表し出力する。またコミュニケーションが目的である場合も、一文をあまりに長くすることは好ましいとは言い難い。

- 性能

* レスポンスタイム

レスポンスタイムは入力されてから出力が始まるまでの時間である。音声入力から音声出力までの時間が長いとユーザーにストレスを与える。かといってユーザーが言い終わるか終わらないかのタイミングで返答することも望ましくはない。

- 機能

* 音声の選択

利用環境や個人の好みに応じて声の選択ができるか。プロダクトにキャラクター性がない場合は、オプショナルとして環境や好みに応じて声を選択できることを望ましい。理由は声の高さの項目で挙げたように、利用者によって聞き取りやすい声、聞き取りにくい声があるためである。また、利用環境によっても落ち着いた声の方がよいといった、利用者がふさわしいと考える声がある。

6.6 テストアーキテクチャ

VUI のソフトウェアテストについて全体的なテストアーキテクチャ例を表 6.3 に示す（ハードウェアのテストは割愛する）。機能の追加時に参考にしていただけたら幸いである。

機械学習を包括したシステムであっても、機械学習モジュール以外はこれまでと変わらないロジカルなシステムである。ここでの全体としてのテストの提案は「機械学習という言葉に惑わされず、従来通り粒度を分け段階的にテストを行う」である。それぞれのテストレベル（ユニットテスト、統合テスト、システムテスト）で段階を踏んだテストを行うことで、後の工程での問題切り分けが容易となるであろう。

システムテストの手法については 6.7 にて提案している。

また、リリース後に実際のユーザー層をターゲットとしたユーザビリティテストも今後の製品の性能向上にも役立つと考えられるため、表に記載している。

表 6.3: テストアーキテクチャ

テストレベル	テスト対象	テスト内容	説明
コンポーネント テスト	<ul style="list-style-type: none"> ・各種機能 (天気、窓を開けて等) ・機械学習部分以外のシステム部分 	各コンポーネントに対する機能テスト	<p>機械学習以外のシステム部分、各種機能（天気、窓を開けて等）部分はこれまで同様の手段でテストが可能である。コンポーネントテスト可能な部分はそのテストで確認をしておくと、機械学習部分と接続後に問題が発生した場合、切り分けが容易になり問題の早期解決につながる。</p>
	<ul style="list-style-type: none"> ・音声認識部分 ・自然言語理解部分 ・音声合成部分 	学習データ・モデルに対する性能テスト	<p>機械学習部分はそれぞれ単体での精度の確認が必要である。 表 6.4 で学習に必要なパターンの例を挙げる。 これらパターンを教師データとして参考とする、あるいは、それぞれ特徴別に精度の評価を行うことができる。</p>
	<ul style="list-style-type: none"> ・各コンポーネント 	各コンポーネントに対する構造のテスト	<p>各コンポーネントの内部構造に着目し、分岐等が設計通りに作られているかの確認を行う。</p>
		各コンポーネントに対する確認テスト	<p>各コンポーネント修正後は、修正箇所が正しく修正されているか確認を行う。</p>
統合テスト	・各種 API	API のレスポンス、DB やモジュール接続後の機能テスト	<p>システムは複数の API とのデータ受け渡しで成り立っている。 全て組み合わせたシステムテスト段階に入る前に、想定されるパターンのデータを用意し API のレスポンスを統合テスト段階で確認しておいたほうが良い。 そうしなければシステムテスト段階で問題切り分けに時間を要することとなる。</p>
		API、サーバーに対する性能テスト	<p>負荷をかけた際にデータを処理しきれるのか、サーバーのスケールアップやスケールアウトが機能するか、レスポンスタイムは許容範囲内かの確認を行う。</p>
		修正後の確認テスト、回帰テスト	<p>変更、修正を行った際は、正しく修正されているかの確認テスト、および周辺影響確認のためのリグレッションテストを行う。</p>

システムテスト	・システム全体を通しての、天気や音楽といった確認対象とする機能	仕様に基づいたスクリプトテスト	Interface 含むシステム全体の仕様に基づきテストケースを作成し、手動または自動で機能を確認するテストである。 テストケースについては「今何時」といった期待結果が明確にわかる内容と、「夏に合う曲をかけて」といった人や環境に依存する内容で分けることが良いかと考える。これら評価方法については 6.7.1 にて n 段階評価を提案する。
	互換性テスト	各社、同じ VUI を搭載した複数のバリエーションのプロダクトを販売している。開発した機能が自社のそれぞれの機種で動くか確認を行う。	
	探索的テスト	自然言語を取り扱う VUI では、特定の行動をさせるためにも複数の言い回しが存在する。よって、仕様に基づいたテストでは発見できない問題が数多く残されている。 探索的テストの時間を多めにとり、表 6.4 をテストチャータに用いる等で確認を進めることが良いであろう。	
	シナリオテスト	各機能（天気、窓を開けて等）には想定された使われ方がある。 機能ではなく、ユーザー主体であるストーリーを作成し、それに沿って確認することで問題発見に繋がる可能性がある。	
	フィールドテスト	実際に使用するロケーションでの確認を行う。 機能ごとの動作テストではなく、実用ロケーションにおける利便性などの確認を重視する。	
	ロングランテスト	音楽機能やラジオ機能は長時間使用する可能性が高い機能である。 カーナビゲーションシステムの場合は長距離ドライブも考えられる。 また長時間にわたる連続した呼びかけでエラーが発生しないかという観点もある。	
精度のテスト		コンポーネントテストレベルでの精度は出るが、実機のマイクを通した際にマイク性能等で精度が出ない問題が VUI では発生しうる。 実機を用いた精度確認を実施する。	

		VUI は人と同じように、何か伝えたときに テンポよく返答をしてもらいたいという期待がある。 ターンアラウンドタイムについて各社基準を設け、 その基準内に返答できるかの確認は必要である。
	性能テスト	
	セキュリティ テスト	個人情報を扱う場面も存在する。 その際には必ず専門知識に基づいた セキュリティテストを行う必要がある。 (脆弱性の有無の確認は、他の製品と同じく、 テストを行う)
	確認テスト、 回帰テスト	変更、修正を行った際は、 正しく修正されているかの確認テスト、 および周辺影響確認のための回帰テストを 必ず行う。
リリース前 リリース後の フィードバック	・システム全体を 通しての、天気や 音楽といった確認対象 とする機能	使用ユーザーを想定して各機能は作られるが、 実際にユーザーがどのように使って、 どこで戸惑うのか確認する必要がある。 また時間の流れとともに話し方や使われ方も 変化していく。 想定ユーザー層に集まってもらい ユーザーテストを実施し、 機能の改善を行うことが より良いと考えられる。
	ユーザーテスト	

図 6.4 テストアーキテクチャ(コンポーネントテスト)

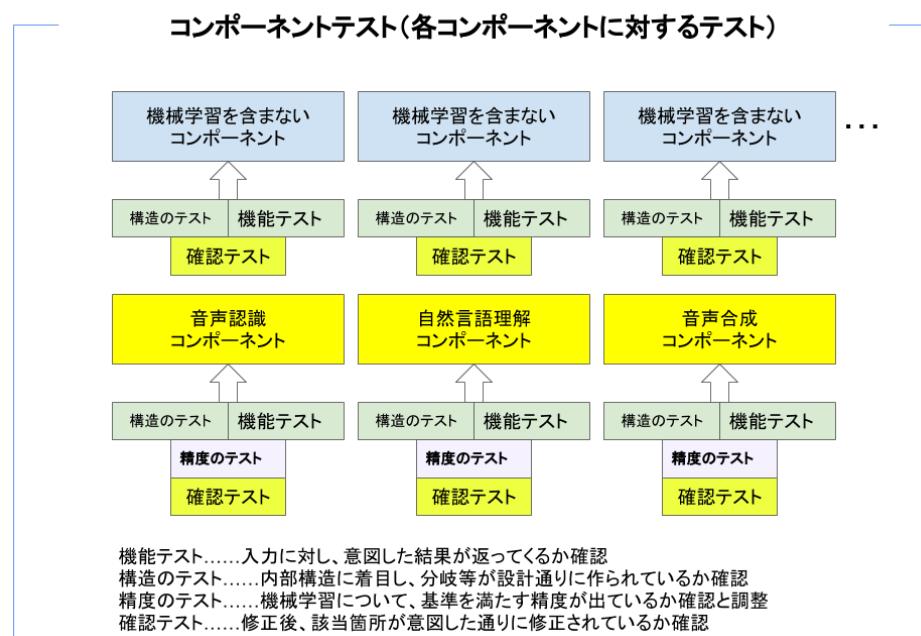


図 6.5 テストアーキテクチャ(統合テスト)

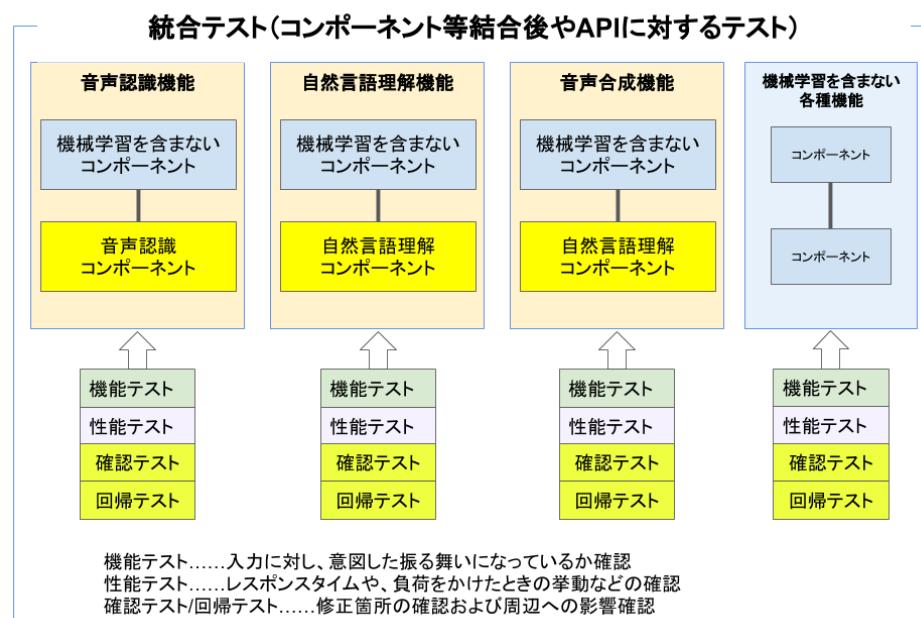


図 6.6 テストアーキテクチャ (システムテスト)

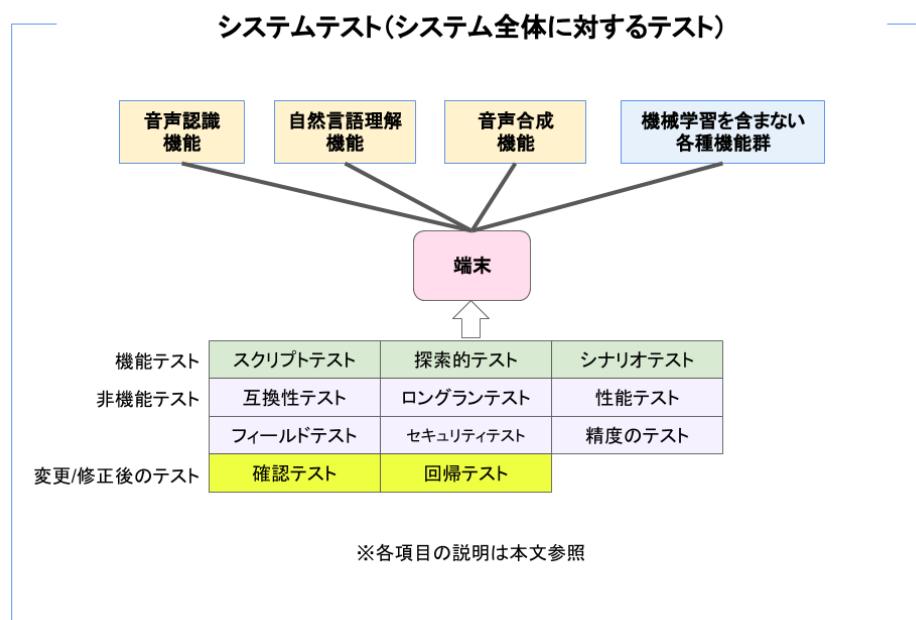


図 6.7 リリース後のテスト

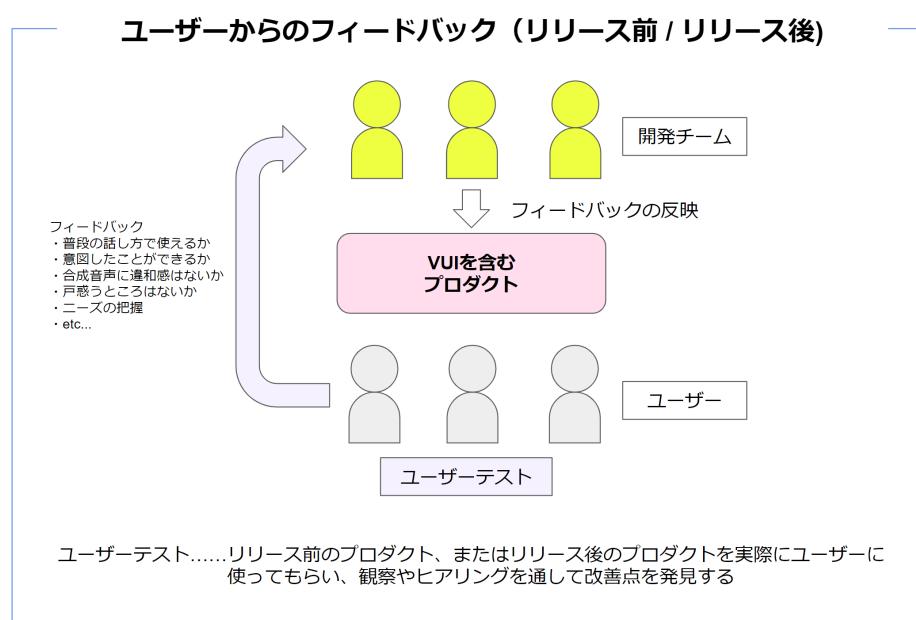


表 6.4: 学習に必要なパターンの例

同一の音声入力文字列であっても、正しく文字列として解釈できること			
音	性別	男女	-
	年齢	高い・低い	-
	トーン	アクセント	-
		早口・区切り	-
		声色	気取った声、猫なで声等
	感情	優しい・厳しい	感情を含んだ声等
	言語初心者	母音記号誤差	日本語の「ら」はl/rを区別しない等
		母国語依存	濁音など（韓国語→日本語等）
異なる表現でも、標準的な文章として理解できること			
テキスト	口調	敬語	-
		命令系	-
		若者言葉	-
	助詞	「の」「は」のばらつき	-
	文法	語順の変更	-
		体言止め	-
	略語	省略形	-
	同音異義語	表現	いどう、はし、あめ等
		曖昧表現	みたい： looks like / to watch / 語尾
		地名	「新宿」「銀座」が日本各所にある
	多言語	和製英語	ナイーブ、テンション
	流行語	-	-
システムの利用者環境が製品保証範囲であれば正しく動作すること			
環境	設置環境	音声ノイズ	テレビ、近所のおしゃべり
		雑音ノイズ	生活音、ドア開閉、扇風機の風など
	設置個所	振動	足場が悪い
		壁面	反響など
システムの利用者が理解できる音声メッセージを伝えること			
音声合成	発音	漢字の読み方	ついたち/いちにち、慣用句、有名人
	トーン	アクセント	-
		早口・区切り	-
		感情	-
		声色	-

情報量		重要度・緊急度	-
システム購入者の音声入力を正確に理解できること			
個別化	話者認識	音声の分離	-
		個人の特定	-

表 6.5: テストアーキテクチャ全体に渡って留意すべき観点の例

テストアーキテクチャ全体に渡って留意すべき観点の例			
Service	コマンド レスポンス	通信	-
		Interface 互換	-
	サービス保証	音質	Music データ通りに再生できる
	不適切なサービス アクセス	ペアレンタル コントロール	-
		卑猥	-
	不適切なサービス レスポンス	宗教	-
		報道におけるタブー	-
		資産	資産、お金周りの誤動作
		プライバシー	個人のプライバシー周りの誤動作
外部 Device を呼び出し			
家電操作	安全性	安全性に影響のある デバイスアクセス	お風呂の温度を 80 度等

6.7 有効な手法

6.7.1 n 段階評価法

期待される品質に記載した「音声入力が話し手の意図通り認識され、意図した機能が実行されること」を確認する場合、問い合わせの抽象度に応じて、期待結果を明確に Yes/No で合否を付けることが難しい場合がありうる。例えば「今何時?」という抽象度の低い問い合わせに対しては、Yes/No で判断できる出力が期待できる。しかし「夏にあう曲を書いて」「楽しい話をして」といった抽象度の高い問い合わせについて意図通りの出力かどうかの判断は難しい。これらは個人の感覚に依存する。

これらを確認する一つの手法として n 段階評価(4 段階、5 段階など)を提案する。該当機能のテス

ト項目の合否の基準値の決定、調査人数及び人選を行う。該当者にテスト項目の実施を依頼し、意図に沿っているか n 段階での評価をつけてもらう。集まった結果の中央値または平均値をとり、規定した基準と比較して「意図通りか」の検証を行う。

5 段階評価の例を以下に挙げる。

1. 意図に反する異なる機能が実行される

- 「音楽をかけて」に対して「占い」が実行される
- 「コンビニに行きたい」に対して「電話」が実行される

2. 意図した機能が実行されるが、機能内において意図と異なるコンテンツが実行される

- 「音楽を止めて」に対して「音楽の順送り」が実行される
- 「シート倒して」に対してシートが起き上がる

3. 意図した機能が実行されるが、意図したことと異なる情報/内容が返される

- 「"歌手名"の曲をかけて」に対して「他の歌手の曲」が返される
- 「"目的地"に行きたい」に対して「違う目的地」が設定される

4. 意図した機能が実行され、意図したコンテンツが返されるが、合っているとまでは言い難い

- 「夏にあう曲をかけて」に対して、曲は返されるが夏の定番曲とは言い難い
- 「"目的地"に行きたい」に対して、違う経由地の経路が返される

5. 意図した機能が実行され、意図した内容が返される

- 「夏にあう曲をかけて」に対して、「夏の定番曲」が返される
- 「"目的地"に行きたい」に対して、最短経路が返される

この際の 1～3 は機能として満たすべき「当たり前品質」部分の評価となる。使用するユーザーからしても不具合として評価される品質である。4, 5 については人や環境によって結果が左右される評価となる。

6.7.2 スモークテスト

音声認識、自然言語理解、音声合成は学習を重ねることで狙った機能の精度向上はするが、他の部分において精度が下がるといった問題が発生する。それら問題が主要ユースケース上で発生していないか早い段階で確認する必要がある。そのために主要ユースケースを通る代表的な発話を列挙し、それらが規定した精度を満たすか、または実行結果が期待結果と等しいかの検証を行う。この検証の場合、結果は成功または失敗と判別が可能で、従来の検証方法と変わらない確認が可能である。

(例) スマートスピーカー

- 今日の天気は
- 夕方の降水確率は
- 毎週平日の 8 時にアラームをセットして

- ラジオを付けて

この検証方法は音声認識、自然言語理解、音声合成の各モジュールでの確認、または全てのモジュールをシステムに搭載した後のシステムテストに適用ができる。またこの検証について各モジュールで保証するか、最終のシステムテスト段階で保証するかの合意を行う必要がある。行わない場合、重複チェックによるテスト工数増大、またはどちらかの工程でテストするだろうという考え方によるテスト漏れが発生する恐れがある。

6.7.3 音声認識の認識精度の評価方法

音声認識の精度の評価方法として、6.2.1 で挙げた要素から自プロジェクトに必要となる要素をピックアップ、それぞれの組み合わせを行い、テストクエリと合格基準を設け検証することが考えられる。

以下にわかりやすく単純化した例を挙げる。例では要素として発話距離、環境、性別、年齢を用いている。

発話距離と聞き取り環境でマトリクスを作る。発話距離は近距離、中距離、遠距離とする(距離の基準はプロダクトによる)。聞き取り環境はVUI システムの周りで音がしていない静環境、周りで音が鳴っている(dB(デシベル) の基準はプロダクトによる)騒音環境、VUI 自体が音声を発している発声環境とする。これら発話距離、聞き取り環境の組み合わせに対して、10 代、30 代、50 代の男性、女性にテストクエリを n 回読み上げてもらい検証を行う。(合格基準はプロダクトによる)

表 6.6 (例) 30 代・女性「今日の天気は」

	近距離	中距離	遠距離
静環境	○	○	○
騒音環境	○	○	×
発声環境	○	×	×

上記はあくまで単純化した例であり、例えばカーナビゲーションシステムの場合は距離が「運転席」「助手席」「後部座席」や、環境が「静環境」「走行中環境」「走行中に窓を開けた環境」といったように、プロダクトに応じて考慮すべき内容が変わるため注意されたい。

6.7.4 自然言語理解のテストケース

自然言語理解モジュールへの入力値は言い方の数だけ存在する。入力される言葉は非常に多様であるが、それは実行したいコマンドを意図している。よって自然言語理解モジュールのブラックボッ

クステストを行う際は、テストする入力値の決定と共に期待結果の用意ができる。

テストケース作成の際は、9.3.4 の自然言語理解項目で挙げた要素から自プロジェクトに必要となる要素をピックアップ、加えて該当機能で重要となるキーワード（例えば天気機能であれば天候や日付）を用いて組み合わせを行うことが考えられる。組み合わせはリスクを鑑み、一因子網羅で良い場所、二因子以上の組み合わせを求める場所について考える必要がある。

また、自然言語理解モジュールのテストの際はテストケース数が膨大になる。このテストは手動で行うことは困難であり、モジュールに対するテストは自動化することを推奨する。このテストは、初期の精度確認のテスト共に、今後のアップデートの際の回帰テストとして必要テストケースを実行する。

例として、表 6.8 にて特定日時の天気通知の機能のテストケース作成方法について記載する。

表 6.8 自然言語理解機能のテストケース作成方法例

作成方法	説明	例
(基本)	仕様などで決められたキーワードとなる要素を含んだ基本となるテストケースを用意する。	<ul style="list-style-type: none"> ・天気は？（質問） ・明日の天気は？（日+質問） ・明日の青森市の天気は？（日+場所+質問） ・明日の午後 3 時の青森市の天気は？（日+時間+場所+質問）
語順変更	テストケースの語順を変更し新たなテストケースを作成する。 日本語は語順が変わっても意味が変わらない関係にあるため、期待結果は基本のテストケースと同様になる。	<ul style="list-style-type: none"> ・青森市の明日の天気は？（場所+日+質問） ・明日の青森市の午後 3 時の天気は？（日+場所+時間+質問）
単語変更	テストケースのキーワードとなる単語を変更し新たなテストケースを作成する。 単語の選定の際は、テスト対象によって重要な単語または失敗時リスクが高い単語を考え選定する。例えば天気機能であれば、人口が多い場所を選定するなどである。	<ul style="list-style-type: none"> ・天気は？→晴れる？/雨？/雪？など ・明日→おととい/昨日/今日/明日/あさって/4 月 1 日/3 日後など ・青森市→横浜市/大阪市/名古屋市/札幌市など ・午後 3 時→3 時/午前 0 時/22 時など
語尾変更	テストケースの語尾を変更し新たなテストケースを作成する。 既にリリースされているのであれば、ログからどういった語尾が多いのか分析をする。リリース前であれば、ターゲットとなる顧客層のペルソナを作成し話し方を想定する。	<ul style="list-style-type: none"> ・明日の天気 ・明日の天気はどう？ ・明日の天気を教えて ・明日の天気って？
助詞変更	テストケースの助詞を変更または削除し新たなテストケースを作成する。 話し言葉の場合、助詞が不正確、ないことがよくある。	<ul style="list-style-type: none"> ・明日は天気は？ ・明日、青森市の天気は？

6.7.5 社内ユーザーテスト

AI プロダクトではテスト条件によって画一的に結果を確定できない部分も多い。ユーザーテストという手法は UX 面での品質向上だけにとどまらず、AI 品質に関しても有効性が高いと言える。特に AI モデルが Deep Learning の場合は、学習データで想定していないケースに遭遇する可能性が高い。

テストを行う場合は上記の理由からプロダクトで想定している利用環境に則した状況でテストを実施する必要がある。(例えば日常利用を想定しているのに社内の静音環境でテストするなどはテストの目的に合致しない)

また、本テストの実施者は、平素テストに関係しない人員を割当することで、より効果が高いと想定される。ただし、テスト経験が少ない人員であるため、どのようなテストであってどのような面を確認して欲しいかなど、テスト実施の目的に関する認識差異は埋めておく必要がある。

本テスト参加者の報告が"問題なく動作する"といったものに偏ってしまうと、テストの目的は達成されない。そのため、例えば以下のような観点を予め本テスト参加者に伝えることを推奨する。

- 音声認識の精度は高いか
- 返答が会話として自然か
- AI プロダクトが導入されたことで生活に影響を与えたか
- 新しい体験が出来たか

6.7.6 データ変更時の精度評価、モデル変更時の精度評価

データ変更時またはモデル変更時に、その変更によって精度がどのように変わったか評価が行われる。この評価の際にデータもモデルも変わってしまうとどの変更によって精度が変化したか切り分けが不可能となる。よって、データ変更による精度を確認したい場合はモデルを固定し、変更前のデータを該当モデルに通し精度の計測を行う。その後、変更後のデータを同じモデルに通し精度の計測を行う。これにより、そのモデルにとってどのようなデータがより適切であるか評価ができる。

同様に、モデル変更による精度を確認したい場合はデータを固定する。変更前のモデルにデータを通し精度の計測を行い、変更後のモデルに変更前と同じデータを通すことによりどちらのモデルの方が該当データに対して精度がより良くなるのかの評価ができる。

6.8 品質保証レベル

VUI システムのシステム全体の品質保証レベルは以下 2 段階が考えられる。図 6.8 に詳細を示す。

1. 動作担保レベル

Yes/No で答えることができる当たり前品質部分のテスト (6.7.1 の 5 段階評価の例であれば 1~3

の領域) の結果が規定した合格基準を満たしている

2. コンテンツ担保レベル

魅力品質部分のテスト (6.7.1 の 5 段階評価の例であれば 4~5 の領域) の結果が規定した合格基準を満たしている

前述の 5 段階評価例では、1~3 の領域を動作担保レベルとして定義し、品質保証のレベルとしては全てが担保されている場合にのみ分類される。このレベルで担保される品質は入力された音声情報から呼び出される機能、モジュール、情報が正しいことをそれぞれ担保する。前提となる入力される音声情報は Yes/No で判断できる情報を前提としており、その前提となる情報から意図した結果が返却されることを判断基準と置いている。

一方で、コンテンツ担保レベルでは、前述の 5 段階評価例において、4~5 の領域を定義している。そのため、前提となる音声情報は抽象度が高いものとなっており、返却される結果が意図されているものかどうかアンケート形式で評価する。コンテンツ担保レベルにおいては、アンケートの結果を評価し、各開発組織ないしプロジェクト毎に合格基準を設け、評価することを推奨する。

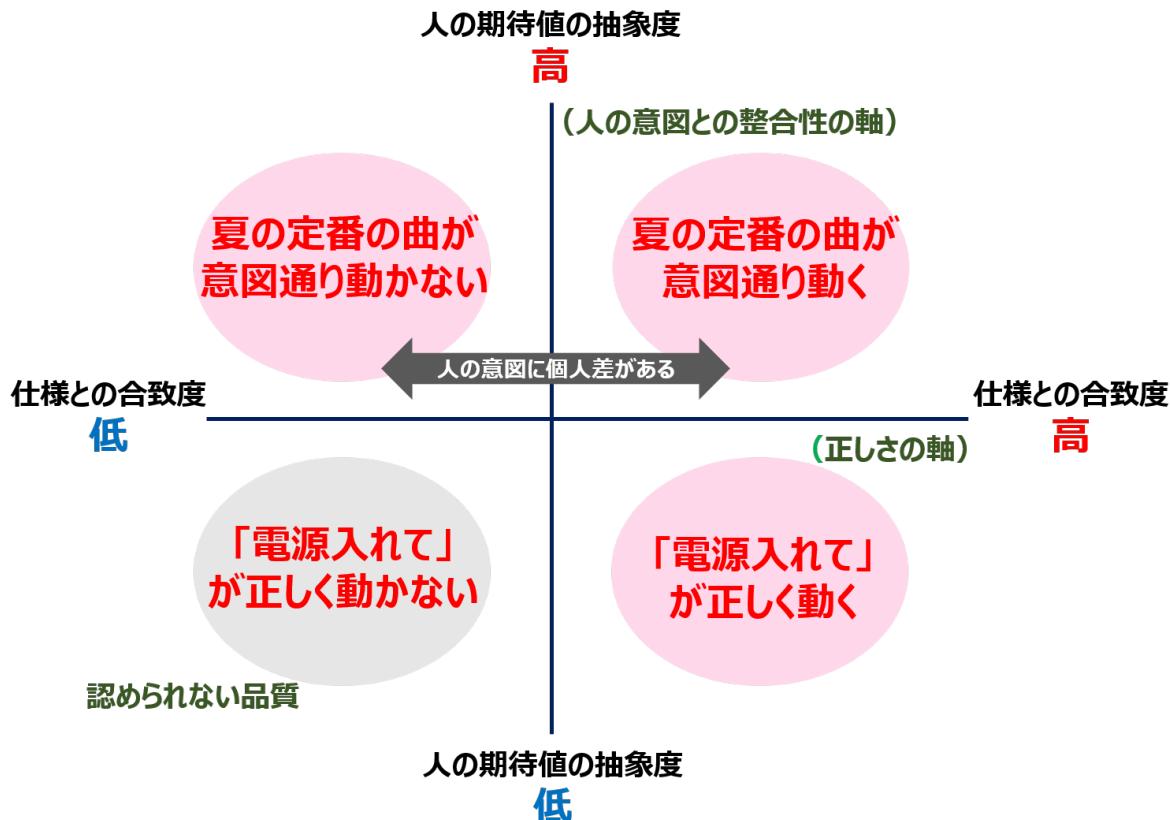
図 6.8 VUI 全体としての品質保証レベル

品質レベル	確認項目	判断基準例	
動作担保 レベル	機能呼び出しの 正常性	抽象度の低い音声入力に対して、意図と異なる機能（スキル）が実行されないこと。 例) 「音楽をかけて」に対して「占い」が実行される場合はNG	低品質
	モジュール呼び 出しの正常性	抽象度の低い音声入力に対して、機能は呼び出されるが、意図したモジュールが実行されないこと。 例) 「音楽を止めて」に対して「音楽の順送り」が実行される場合はNG	
	情報呼び出しの 正常性	抽象度の低い音声入力に対して、意図した機能/モジュールが呼び出され、意図したコンテンツ（情報）が呼び出されること。 例) 「”歌手名”の曲を書いて」に対して「他の歌手の曲」が返される場合はNG	
コンテンツ 担保レベル	呼び出された 情報の正常性	意図が抽象的な音声の入力に対して、意図した機能/モジュールが呼び出され、アンケート上、意図したコンテンツ（情報）と判断される割合が高いこと。 例) 「夏にあう曲をかけて」に対して、「夏の定番曲」が返される場合は該当	高品質

これらの品質レベルを適用する際には、抽象度の切り分けが重要となる。

抽象度の切り分けの指針としては、仕様との合致度（仕様として定義できるもの）と、人間の期待値の抽象度（期待結果が人によって変化するか否か）を基準とすることを推奨する。例を図 6.9 に示す。テスト時に利用する音声データの期待結果を下図のように切り分けを行い、実施されるテストが動作担保レベルなのか、コンテンツ担保レベルかの切り分けを行うことで、適切に品質レベルを評価することが可能となる。

図 6.9 抽象度の切り分けの指針の例



また機械学習の場合、品質を保証するためには出力されるデータのみの確認だけではなく、学習元となった教師データの品質保証も重要である。

教師データの品質保証レベルは 2 つに分けることができる。

- 特定しないデータの使用（特にカテゴライズされていないデータ）
- 6.2 にて記載した各要素において特定したデータ

6.2 に挙げた音声認識機能、自然言語理解機能、音声合成機能の各要素に関して、各組織においてどこまで保証するかの範囲と基準を設けることにより、それらのどこまでを達成できているのかより細かくレベル分けができると考える。

6.9 VUI における個人情報およびプライバシーの保護

VUI である以上音声データは必ず取り扱いが発生する。本項目は日本の法律・環境を前提とした上で、個人情報保護法遵守・プライバシー保護の観点からそれぞれの注意について記載する。

6.9.1 個人情報保護法

個人情報保護法上、個人情報の取扱いには一定の法的義務が課される。また、日本国外においてもサービスを提供する場合には、各国の法令が適用される可能性があるところ、個人情報保護法制は国ごとに異なるため、各国の法令を遵守する必要がある点に留意すべきである。本項目は、個人情報保護法の内容の解説を目的とするものではなく、音声データの取り扱いにおいて特に注目すべき点について記載するものである。

個人情報として取り扱われる可能性のあるパターン

日本法においては、個人を識別できる情報が個人情報に該当しうるところ、音声データ自体は、通常はそれにより個人を識別できないから、個人情報には該当しないことが多いと考えられる。しかし、下記の例外のように個人を識別しうるパターンが存在し、その場合は音声データも個人情報として取り扱われる場合がある。

表 6.9 個人情報として取り扱われる可能性のあるパターン

パターン	例
音声の内容に個人情報が含まれる場合	音声の内容に氏名や生年月日等が含まれるケース
音声データと個人情報が紐づいている場合や、個人情報と容易に紐づけ可能な場合	音声データと個人情報を持つアカウントが紐づけられるケース
音声データを解析して個人を識別できるデータにした場合	音声認証システム等

個人情報に該当する音声データの利用・保管について

個人情報に該当する音声データを利用する場合、利用目的を通知又は公表しなければならない。音声データをサービスの提供だけでなくサービスの開発にも用いる場合、このような利用はユーザーが予期していないことも考えられるため、特に分かりやすく利用目的を伝える必要がある。また、音声データが個人情報に該当する場合、取得した情報の開示等に応じる義務が発生する場合もある。

個人情報を自社のみで利用する場合には、利用目的の通知又は公表を行えば足りるため基本的に本人の同意を得ることまでは求められないが、複数社で共同開発する場合のように、自社以外でも利用する場合には、音声提供者から同意を得るなどの措置が必要なこともある。

音声データの収集・処理における注意点

VUI のサービスは、音声データの取得が長期間にわたる場合もあり、その場合には音声の内容に個人情報が含まれる可能性も高まるため、個人情報に該当するものとして取り扱うことが求められ

る場合も多い。

また、クライアント側（エッジ側）で音声データを処理しているとしても、その時点で個人情報の取得として扱われる場合もあるため、注意が必要である。

VUI のサービスは、その性質上、いつ音声データを収集しているかが分かりにくい場合もあるため、利用目的の通知・公表の際には、音声データの収集タイミング等についても説明をすることが望ましい。

6.9.2 プライバシー保護

前項はあくまで個人情報を保護するための要件である。実際にユーザーが安心・安全に VUI を利用するためには音声データに対する個人情報保護法の遵守の他、プライバシーへの配慮も必要となる。本項目では音声データを取り扱う際のプライバシーへの配慮の観点について記載する。

音声データで考慮すべき主な事項

プライバシーへの配慮のために必要な措置は個別のケースによって異なるが、以下のような観点からの検討が有用である。

表 6.10 音声データで考慮すべき主な事項

カテゴリ	例
取得状況	<ul style="list-style-type: none"> ● 取得空間 <ul style="list-style-type: none"> - 公共の場の音声 - プライバシー空間の音声 ● 取得範囲 <ul style="list-style-type: none"> - 常時（すべての音声） - 限定的（電話内容の録音等）
利用目的	<ul style="list-style-type: none"> ● サービス提供のみ ● サービスの品質向上（機械学習） ● サービス以外の利用（マーケティング等）
連動するデータ	<ul style="list-style-type: none"> ● 個人のアカウント（購買履歴等） ● 音声解析データ（声から解析した年齢・性別等） ● デバイスの識別子
保持期間	<ul style="list-style-type: none"> ● サービス提供として必要な期間のみ保持しているか ● 利用目的として達成後も保持を続けていないか
ユーザーの意思尊重	<ul style="list-style-type: none"> ● オプトイン/オプトアウト ● 削除請求への対応の容易さ
取得場面のユーザー認識	<ul style="list-style-type: none"> ● ウェイクアップワードの取得 ● 誤認識時の録音
音声データへのアクセス権	<ul style="list-style-type: none"> ● 役職/雇用形態 ● 委託先への提供等
プライバシー方針の提供	<ul style="list-style-type: none"> ● ユーザー目線でのわかりやすさ ● 利用目的や保持期間、利用を望まない場合の手段等の説明
データ解析時の処理方法	<ul style="list-style-type: none"> ● 個人を特定できない形に加工する

プライバシーへの配慮には絶対的な基準はない。本項目で挙げた内容をすべて配慮したとしても、配慮不足は起こり得る。その為、ユーザーが安心・安全に利用するための観点というものを常に考慮し続けることが VUI としても必要である。

VUI はその性質からプロダクトの利用層に子供が含まれることも少なくない。そのためプライバシー配慮が不足している仕組みでもプライバシーポリシーへ記載している・同意しているから問題ないという考え方を良しとするのではなく、充分なリテラシーを持ち合わせていないユーザー層が利用しても安心・安全だと言えるようなプライバシー配慮がなされた仕組みが必要であり、その安

心・安全な状態をプライバシーポリシーに明記するという考え方が重要である。

6.9.3 その他音声データの取り扱いに関して考慮できる内容

本項目では音声データの取り扱いに関して実際に存在する事例について記載する。個人情報保護法を遵守し、プライバシー面のリスク回避やプライバシーに配慮した運用を行うことでユーザーにとって安心・安全な状態を提供できる。

事例概要	事例ポイント	考慮ポイント
録音データで面白い内容があつた場合、従業員間で共有されていた	<ul style="list-style-type: none"> ・本来の目的とは関係ない共有理由 ・録音データがアカウント番号、ファーストネーム、デバイスのシリアル番号と紐づいていたとされる 	システム的な問題ではなく、運用する側の意識における問題。データの取り扱いに関する指導もプライバシー配慮においては重要。
殺人事件の証拠としてスマートスピーカーの録音データが使用された	政府から企業へ、録音データの開示要求があった	<ul style="list-style-type: none"> ・法令による開示要求または公的機関からの開示要求があった場合の、利用者データの開示条項が利用規約に記載されているか ・何のデータが、どういった目的で、どこに保存されているのかが明確になっているか

事例概要	事例ポイント	考慮ポイント
スマートスピーカーと連携するサービスの Web サイトに脆弱性があり、音声履歴がハッキングされた	<ul style="list-style-type: none"> ・脆弱性のある Web サイトはスマートスピーカーとは関係のない同社サービスの Web サイトであった ・自宅の住所以外にスキルやアプリなどのプロフィール情報も流出していた可能性があった ・攻撃者が悪意あるスキルをインストールし、さらにデータを抜き出すことも可能だった ・自衛手段として履歴削除は出来る 	VUI を使ったプロダクト単体だけではなく、関連するサービス、アカウント側も安全性が充分に担保されているか考慮する。いつ起こり得るかわからない事象に対しては、履歴削除のみでは充分な防衛手段とはならない。
他人のプライバシー情報を音声操作により収集できる	個人アカウントと紐づいた設定がされているスマートスピーカーに対して、その個人アカウントの保持者以外の人物が発話した場合、音声コマンドの種類によってはプライバシー情報が引き出される可能性がある	発話者特定機能の実装により、情報流出のリスクを低減できる
近くに人がいるかどうかの情報や、位置情報が傍受される可能性	使う人が近くにいることにより在室であるといったプライバシー情報が発生し、悪意の第三者でその情報を知りたい人がいれば、有効なプライバシー情報になる（ストーカーなど）	通信路やデータ保存のセキュリティを確保することにより、情報を傍受されるリスクを低減できる

事例概要	事例ポイント	考慮ポイント
音声データを利用した学習を機械自身が自動で行うようにした	<ul style="list-style-type: none"> ・教師あり学習の場合「ラベル付け」の作業が必要であり、音声データを人間が聞く必要があることで、取り扱い上の問題が発生する懸念がある ・ラベルのない教師なしデータから自動学習を進めることでプライバシー配慮としても利がある 	プロダクトの機能やシステムにおけるプライバシー配慮だけではなく、運用面でも配慮することができる

7. 産業用プロセス

7.1 検討の前提と対象

本章では、制御システムを例とする産業用システムでの AI の品質保証を対象とする。産業用システムでは、品質安定化・生産性向上を目的として、統計的品質管理やフィードバック制御等の制御技術が発展してきた。近年、画像を代表とする自動識別や、設備の予知保全を目的とした異常検知・変化点検知の導入など、多方面での機械学習技術の応用・実用化が進められている。図 7.1 では産業用システムアプリケーション例と、AI 技術を適用した機能の例を示す。



図 7.1 産業用システムアプリケーション例と AI 技術の適用例

このような多様な産業用システムでの品質保証を検討するにあたり、プラント制御を主な題材とし、次の順で検討を進めた。

まず、7.2 項では、産業用システムに AI 技術を適用するにあたり、考慮すべき重点課題を明確にする。次に、7.3 項にて、AI 技術を組み込んだ産業用システムの基本的なアーキテクチャを示し、その構成要素（コンポーネント）ごとに、7.2 項で示した重点課題に対する考慮事項を示す。7.4 項では、想定するステークホルダーを示す。また、7.5 項にて、AI 技術を適用した産業用システムの PoC から運用に至る工程と細部プロセスを示し、その工程ごとに、7.2 項で示した重点課題に対する考慮事項を示す。7.2 項～7.5 項の検討を踏まえて、7.6 項では、産業用システムに対する品質保証上の考慮事項を 5 つの指標 (Data Integrity, Model Robustness, System Quality, Process Agility, Customer Expectation) に基づいて具体化する。7.6 項の具体化された産業用システムにおける 5 つの指標に対する工程との対応を 7.7 項、AI 技術の適用を具体的なイメージで公開された事例に基づく品質保証検討例を 7.8 項に示す。以上を踏まえて、AI 技術を適用した産業用システムに対する品質保証の検討の流れを。図 7.2 に示す。

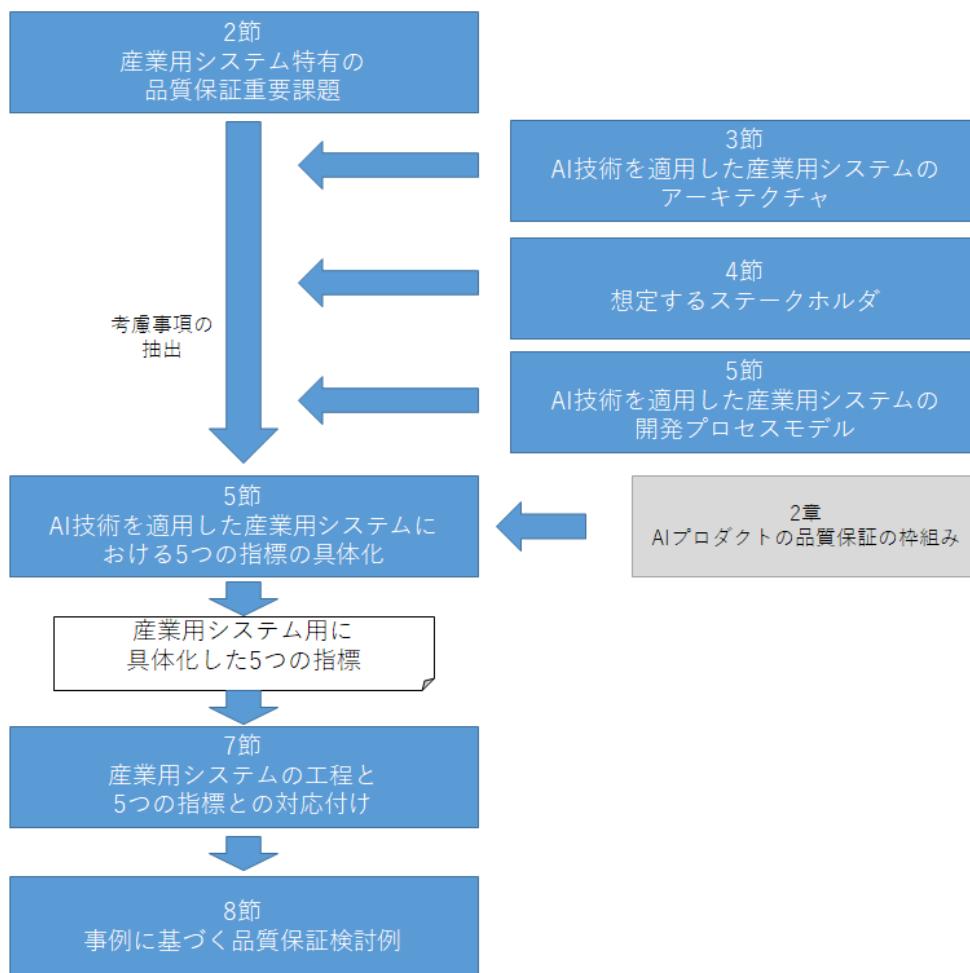


図 7.2 本ガイドラインでの産業用システム品質保証検討の流れ

7.1.1 本章で用いる用語定義

本章では、各用語を下記に示す意味で使用する。用語の定義は AI・データの利用に関する契約ガイドライン [4] を参考にしている。

表 7.1: 用語定義

用語	意味
AI	人間が知能を使って行うことを機械にさせることであり、とくに「機械学習」を用いて実現する場合を指す。
機械学習	あるデータの中から一定の規則を発見し、その規則に基づいて未知のデータに対する推論や予測等を機械的に行う手法の総称。
AI システム	AI を利用して目的を達成しようとするシステム。 AI コンポーネントだけでなく、周りの非 AI コンポーネントも含むシステム全体を指す。AI プロダクトと同義。7.8 項の例では、包装機が AI システムとなる。
AI プロダクト	AI システムと同義。
AI コンポーネント	AI システムを構成する部品のうち、AI を用いて何らかの処理を行うもの。7.8 項の例では、フィルム蛇行検知部である。
AI アルゴリズム	機械学習を行う方法や手順。代表的な AI アルゴリズムにニューラルネットワークがある。AI アルゴリズムに訓練データを適用して AI モデルを生成する。
AI モデル	AI コンポーネント内で、入力されたデータから推論、予測などを行う部分。ニューラルネットワークや重みパラメータなどで構成される。 機械学習により生成され、結果は重みパラメータなどに反映される。
生データ	センサや提供元などから一次的に取得されたデータで、データベースに読み込むことができるよう変換・加工処理されたもの。 (欠損値や外れ値を含む等、そのままでは学習を行うのに適していない場合が多い)
学習用データセット	生データに対して、欠損値や外れ値の除去などの前処理、ラベル情報の付与等の二次加工を行ったデータの集まり。 訓練データとテストデータで構成される。
訓練データ	学習用データセットに含まれるデータのうち、AI モデルに対する機械学習を行うために使用されるデータ。
テストデータ	学習用データセットに含まれるデータのうち、学習済みの AI モデルの

	汎化性能等の確認のために使用されるデータ。
運用データ	実際の運用時に AI モデルに入力されるデータ
入力データ	推論・判別を行うために AI モデルに入力するデータ
出力データ	AI モデルに入力データを与えた際の出力
学習用プログラム	機械学習を行うプログラム。
AI プログラム	学習用プログラム及び AI コンポーネントで用いるプログラムの総称。
ハイパーパラメータ	学習率、学習回数（エポック）や Drop Out のパラメータ等、機械学習の枠組みを規定するために用いられるパラメータであり、主として人為的に決定されるパラメータのこと。
再学習	AI モデル生成後に、学習用プログラム自体はそのままとして、再度、学習用データセットやハイパーパラメータを変更して、最初から学習しなおすこと。
追加学習	既存の AI モデルに、異なる学習用データセットを適用して、更なる機械学習を行うこと。
蒸留	既存の AI モデルへの入力及び出力データを学習用データセットとして利用し、小規模化やシンプル化した新たな AI モデルを生成すること。

7.2 産業用システムへの AI 技術適用にあたっての重点課題

産業用システムでは、AI 技術自身の特性だけでなく、「信頼性」「安全性」といった対象システムの品質目標や、システムの特性・制約を踏まえた課題解決が品質保証に必要である。

これを踏まえて、産業用システムの品質保証の重点課題として、次の 3 つが挙げられる。

1. ステークホルダー多様性：大規模・複雑なシステムであることから、複数事業者が契約に基づいてサブシステムの構築・運用がなされる形態が多い。個々のデータの整合性や、権利保護、システム全体を検証する必要がある。
2. 環境依存性：システムは 5M+E の変化に晒されており、多様なデータや再現性の異なるデータを前提とした保証が必要である。
3. 説明容易性：システムが妥当であることを保証するプロセス・規格が複数存在し、顧客への説明責任と納得を引き出す必要がある。

そこで、産業用プロセス WG では、QA4AI ガイドに基づく 5 つの指標によるバランスチャートでの観点と、AI プロダクトの工程（PoC/開発/運用）を整理することで、課題に対するコンセンサス向上を目的としたガイドライン策定を進めた。

産業用プロセス WG は、ガイドライン利用者が本検討をもとに對象 AI プロダクトの要件や特性

に基づいて拡張し、関係者との計画や実施での合意形成を促進することを期待する。

* 5M+E とは： Man, Machine, Method, Material ,Measurement+ Environment の略称。機械加工や工場の品質管理で利用される用語。品質の変化要因である、人、機械設備、方法、材料、検査・測定、環境といった代表的な管理項目のこと。

7.3 参照システムアーキテクチャ

産業用プロセス WG では、多様なシステムの品質保証活動を類型化するにあたり、システムアーキテクチャを抽象化した参照システムを用意した（図 7.3）。

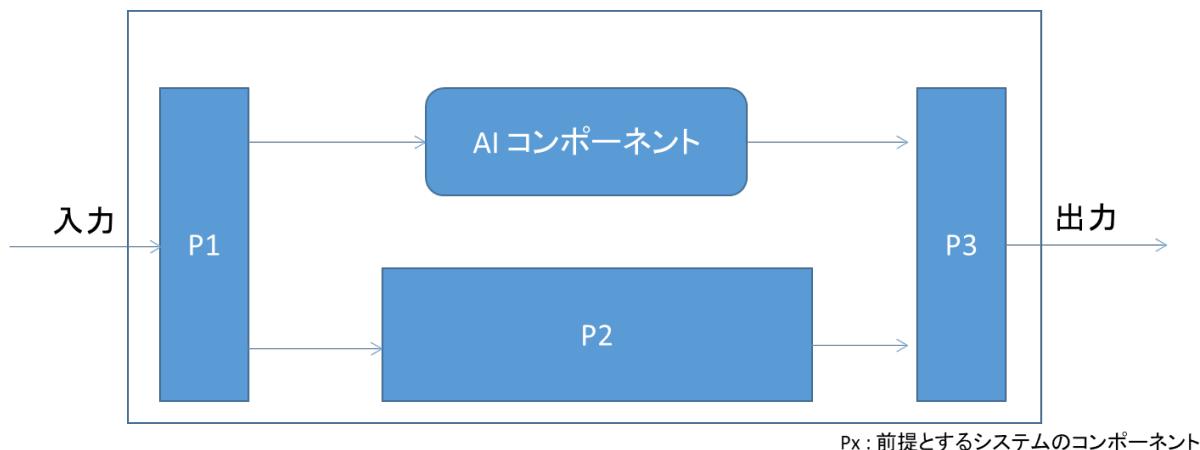


図 7.3 参照システムアーキテクチャ

現実の産業用システムは、この参照システムを連結・階層化して構成する。P1 から P3 は、AI コンポーネントを追加する前提となる既存システムのサブシステムである。AI コンポーネントにとつて入力部が P1、出力部が P3 である。P2 は AI コンポーネントにとつては代替となる制御処理部である。

AI コンポーネントの確率的なふるまいに対して、以下の拡張が必要となる。

- P1 : AI コンポーネントが必要とするデータの前提・精度・ラベルの妥当性の保証
 ステークホルダー多様性への対応 データ整合性の保証
 環境依存性への対応 データ区間・範囲の保証
 説明容易性への対応 AI コンポーネントの振る舞いを説明するためのデータ記録

P1 は画像検査を例とした場合、適切な輝度・照明角度・焦点といった保証を担い、必要に応じて対象画像データを保存する責任を持つ。

プラント制御では対象材料のセンサ値が正当であることを保証する。

- P2 : AI コンポーネントの代替系・監視

環境依存性への対応 AI コンポーネントでの動作前提対象外となりえる事象の監視・観測

説明容易性への対応 AI コンポーネントの動作性能の比較・保証

P2 は画像検査を例とした場合、次のような例がある。

例 1：人がサンプル検査を担うことで環境依存性に対する評価を担い、AI コンポーネントの監視を担う。

例 2：画像検査以外の異なるルールベースの指標（例えば、位置、タイミング、検査装置等）によるモニタリング。

プラント制御での異常検知を例とした場合。P2 はセンサや制御データをルールに基づき異常監視を行う一方で、AI コンポーネントは機械学習で異常監視を行う動作となる。P3 が異常発生を予見し出力する動作では、AI コンポーネントが異常と推定し、P2 が正常と判断するといった動作の組み合わせで判断することもある。

- P3 : AI コンポーネントが出力するデータの結果・精度に対する出力の保証

ステークホルダー多様性への対応 P3 の出力に対する AI コンポーネントの出力値の寄与のさせ方の明確化

環境依存性への対応 環境依存性に対する P2 の情報に基づく AI コンポーネント出力のフィルターや判断

説明容易性への対応 AI コンポーネントの出力と判断結果の記録

P3 は画像検査を例とした場合、AI コンポーネントは例えば「良品とする集合に属する確率 x %」、「不良品とする集合に属する確率 y %」といった結果を出力する。この結果をシステム全体としてどのように利用するかをルールに基づき確定する。（例えば、x、y それぞれに対して閾値を用意する）

プラント制御を例とした場合。AI コンポーネントが認識したシステムの状態をもとに、P3 がシステムを最適状態に推移するための補正指令を出力するケースがあり得る。多様なステークホルダーに妥当な補正指令であることを示すには、AI コンポーネントが認識した状態、環境依存性、システム品質の関係性を、何かしらの実証手段に基づき説明する必要がある。

本ガイドラインの作成時点では、これらの問題に対し、機械学習技術を利用したシステムの保証をアーキテクチャで担保する方法はない。品質保証上の留意点をもとに個々のシステムごとに解決する必要がある。そこで本 WG では、ガイドライン利用者が実システムでの品質保証活動で活用するため、品質保証活動の 5 つの軸での留意点を整理した。

7.4 想定ステークホルダー

産業用プロセス WG では、AI プロダクト開発に関わる関係者をステークホルダーとして定義する。ステークホルダーは AI のシステムを開発するメーカー側と AI を導入するお客様側に分かれます。お客様側は主に工場であり、生産性向上に AI を適用することを想定している。ステークホルダーの相関図を図 7.4 に示し、それぞれの役割を表 7.2 に記す。

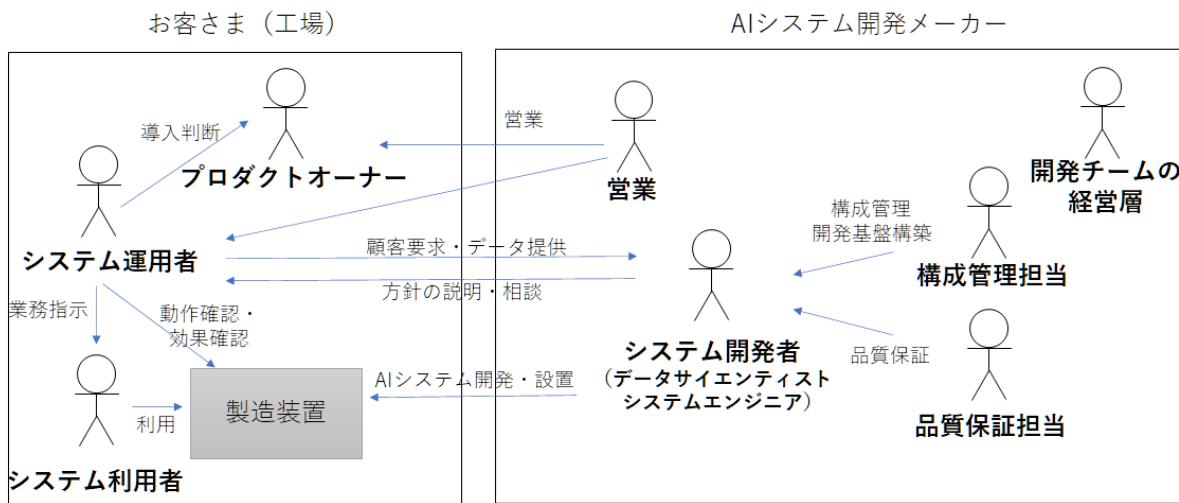


図 7.4 ステークホルダー相関図

上記ではシステム運用をお客さま側で実施するものとして記載した。システムの運用をメーカー側が行い、お客様側はサービスだけを受けるという形態も考えられる。この場合、システム運用者はメーカー側にいる。一方、お客様側にはサービス利用部門があり、その中に要求を持っているサービス利用部門責任者や、実際にサービスを利用するサービス利用者がいると考えられる。

表 7.2 ステークホルダーの役割

ステークホルダー名	主な役割
開発チームの経営層	開発チームを管理する。
営業	お客様に対して、AI システムを使った生産性向上の企画・説明を行い、AI システム導入を推進する。
システム開発者	AI システムを設計・開発する。図ではシステム開発者としか配置していないが、実際にはアーキテクトやデータサイエンティストなどの専門家や、システム統合を行うシステムエンジニア、実装を担当する協力会社など、多くのメンバで構成されることが多く、前節のステークホルダー多様性を考慮する必要がある。
プロダクトオーナー	AI システムで導入に対して権限と責任を持つ。システム運用者の管理者である。
システム運用者	工場のライン管理と、AI システム導入および運用を担当する。導入に対する要求などを持っており、生産性の改善が業務となる。メーカー側から見ると顧客である。
システム利用者	実際に導入した AI システムを利用する人である。
構成管理担当	AI およびシステムの構成管理やインフラ構築・メンテナンスを担当する。
品質保証担当	AI およびシステムのプロダクト・プロセスの品質保証を担当する。

7.5 品質保証活動

産業用システムでは、要求品質を満たすために、システム開発の工程をプロセスで分解し、個々の過程が適切であることを計画的に検証することで品質を保証している（プロセスコントロール）。実システムでもプロセス品質の考え方を前提としたうえで、AI プロダクトの品質保証を工程ごとに 5 つの指標と関連づけて達成することが、品質保証上有用であると考える。

本 WG では SQuBOK®[2] のプロセス形態を参考に、開発工程の全体像を定義した産業用システムの全体工程を IXI（Intelligent eXperimental Integration）モデルとして図 7.5 のとおり定義する。本工程は、大きく PoC、開発（機械学習プログラムの開発工程含む）、運用の 3 工程に分割される。「PoC」では開発・運用での主要リスクの確認や検証を実施する。PoC の検討結果を受けて、「開発」では、5M+E の特性を考慮に入れながら AI 技術を適用した産業用システムを開発する。「運用」では、開発したシステムの出力を監視するとともに、出力された結果に対してステークホルダーへの説明に必要なデータの収集や運用時に初めて確認されたデータを用いて評価を行い必要に応じてモ

デルの更新を行う。なお、図 7.5 の薄橙色背景の枠囲みは AI を使ったプロダクトの開発で特に重要な活動を示す。

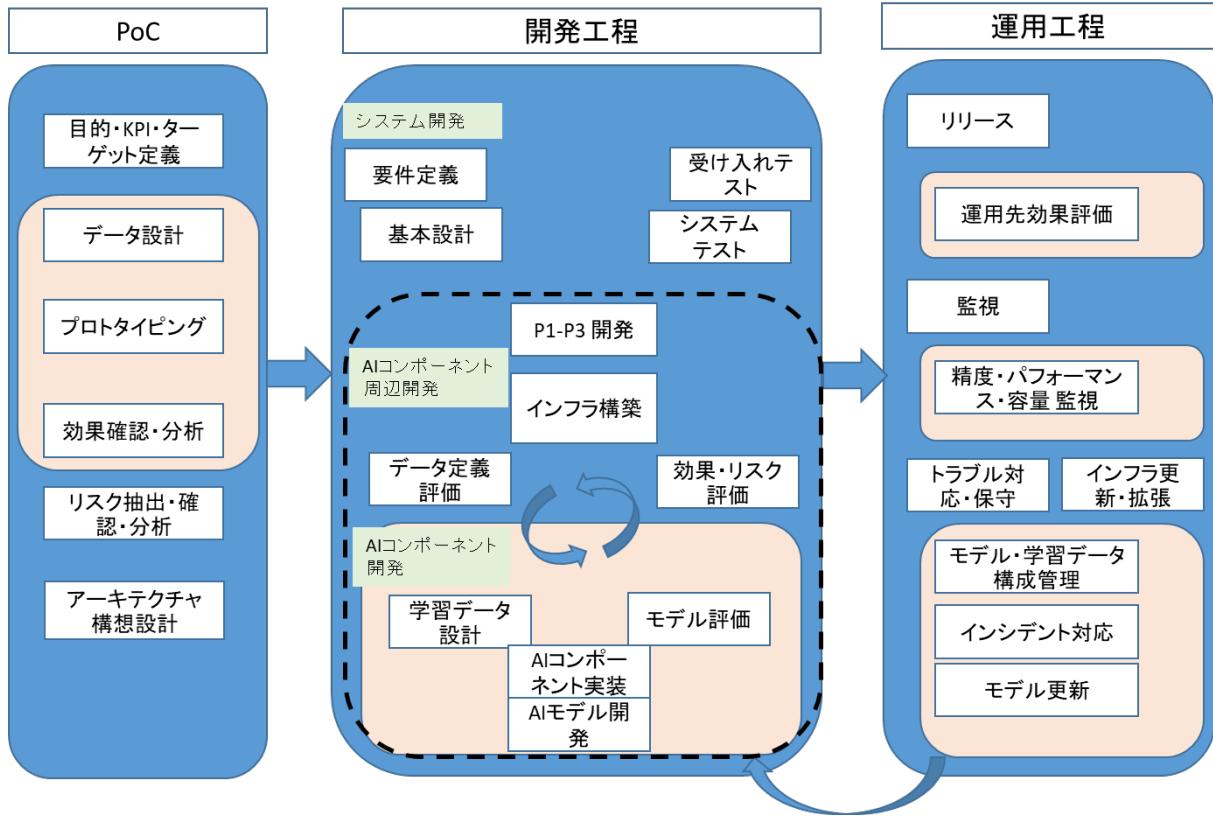


図 7.5 工程の全体像 : IXI: Intelligent eXperimental Integration モデル

これら 3 つの工程により、7.2 項で示す 3 つの重要課題を整理することにより、産業用システムにおける考慮事項が表 7.3 のとおり抽出される。この分類により、PoC ではステークホルダー多様性、開発では環境依存性、運用では説明容易性がそれぞれ重点課題となることがわかる。

IXI モデルを実際に適用する場合、たとえば、PoC ではステークホルダー多様性に対応する比重が高い場合、5 つの指標のうち Customer Expectation と、Process Agility が高くなる。また、開発においては、運用中でしか発見できない事象を除いて、環境依存性を考慮しながらシステム開発を進めることで、Data Integrity および Model Robustness を高め、System Quality を本番相当まで高くする。運用に至り、実現場でのデータやモデルの獲得や、現場運用に適合することで、全ての 5 つの指標による品質保証のバランスチャートが整い、提供する AI 技術を適用した産業用システムの品質保証度合いを明示することが可能となる。

品質保証のバランスチャートの変化の一例を図 7.6 に示す。

具体的なレベルの計測法・尺度は、本ガイドラインを参考に具体化する必要がある。開発プロセ

表 7.3 3つの工程による重要課題の整理

工程	PoC	開発	運用
目的	達成可能性、実現可能性を確認し、多様なステークホルダーと開発合意に至る	PoC 結果や開発中の結果に基づき、環境依存性への保証事項・方法を確立し、運用合意に至る	AI システムを現場で動作し、性能や発生事象を評価・対応し運用を安定させる
ステークホルダー多様性	関係者との目標・リスク等の合意とプロトタイプによる実証	インターフェース・API 設計や、データ一貫性への対応	現場運用要件・変更管理要件等への対応
環境依存性	一部の環境条件に基づく目標・リスク評価	システムが対象とする環境条件の明確化と、システムの開発・検証	仕様外の環境の監視とデータ収集・評価
説明容易性	説明要件の洗い出しとシステム要件への反映	構成管理された環境依存データによるモデル評価やコード品質の説明	事象発生時、性能変化時のデータ・モデル・構成に基づく理由と対応策の説明

スでの個々の活動の関連付け (7.7 項) や検討例 (7.8 項) を参考にしていただきたい。

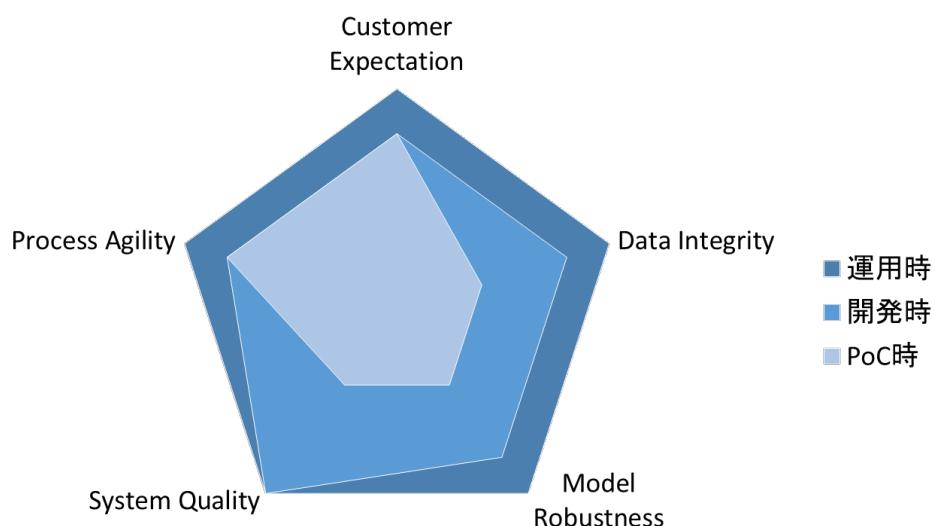


図 7.6 品質保証レベルのバランスチャート例

7.6 産業用システムにおける 5 つの指標の具体化

7.6.1 QA4AI 品質保証観点の解釈

産業用システムは、PoC(目標設定・プロトタイピング・効果確認/分析)、開発(設計・製造・据付)、運用(運転・保守)といったシーンでの実施責任者・関係者が異なり、QA4AIで規定した5軸の解釈の重みづけが異なる。そこで、本WGでは5軸の品質保証活動の留意点(解釈方法)を示すと共に、各留意点毎に対象となるシーン(PoC、開発、運用)を明示する。

また、具体的な品質保証活動を想起できるよう、各留意点をかみ砕いた内容で説明し、留意事項を怠った場合の影響を併せて記載する。

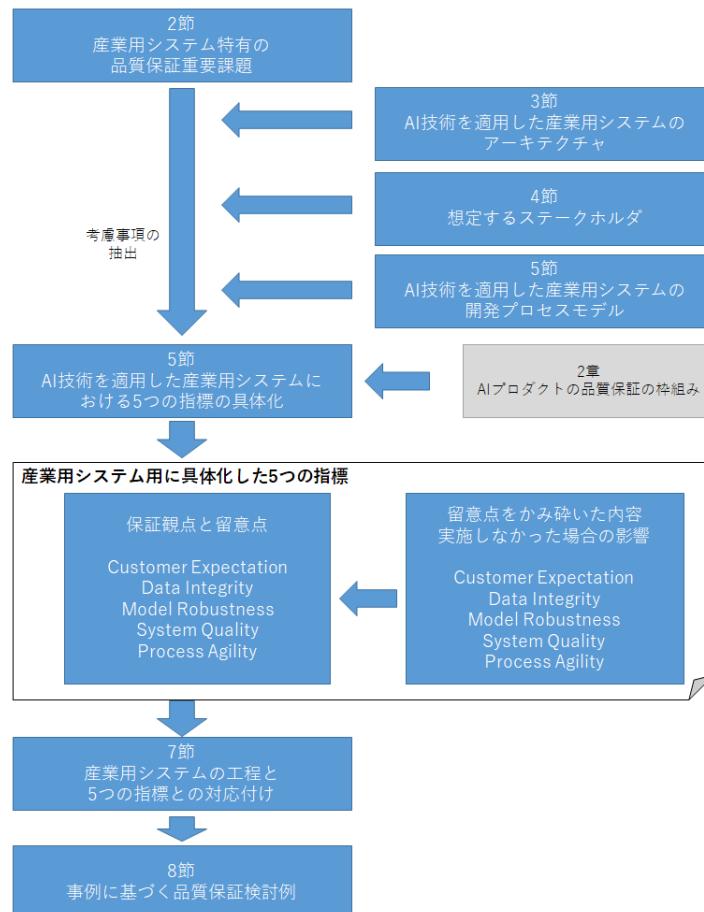


図 7.7 産業用システムに具体化した 5 つの指標

7.6.2 Customer Expectation

産業用システムは、ステークホルダー多様性により、複雑な利害関係がある。結果、AI システムに対する期待の齟齬だけでなく、データ・モデルといった機械学習に必要な無形資産の権利・共用性にも調整が必要となる。

表 7.4 Customer Expectation の留意事項

ID	観点	留意事項	PoC	開発	運用
CE-1	顧客側期待の高さ 狙っているのが「人間並み」なのかどうか	● AI によって解決したいビジネス課題の明確化 ・顧客のビジネス課題が明確になっているか。(AI の適用が目的になっていないか。)	<input type="radio"/>	<input type="radio"/>	
		● AI によるビジネス課題の解決可能性 ・AI によって顧客のビジネス課題は解決可能か。	<input type="radio"/>	<input type="radio"/>	
		● AI によって解決したいビジネス課題解決の効果 ・AI を利用することによる顧客の期待効果（目標性能等）は、明確になっているか。 ・「人間」と同等以上の効果を期待しているか。	<input type="radio"/>	<input type="radio"/>	
		● AI の性能維持に対する継続改善の理解度 ・AI の性能を維持するためには、継続的に学習を重ねて改善を図る必要があることを顧客に理解されているか。	<input type="radio"/>		<input type="radio"/>
		● AI によるビジネス課題解決の満足度 ・顧客は、AI による課題解決結果に満足しているか。			<input type="radio"/>
CE-2	確率的動作という考え方の非受容 顧客側でのリスク・副作用の無理解、容易な需要による対策不備	● 確率的動作で出力される AI への理解度 ・AI の出力結果が、確率的動作（確率的にもっともらしい解）により出力されることを顧客に理解されているか。	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
		● 確率的動作で出力された結果のリスク許容度 ・確率的動作（確率的にもっともらしい解）により出力された結果を許容することは可能か。	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

表 7.5 Customer Expectation の留意事項（続き）

ID	観点	留意事項	PoC	開発	運用
CE-3	継続的実運用への近さ PoC やβリリースという概念を理解していない度合	● アジャイル型ソフトウェア開発への理解度 ・ 従来の演繹的なウォータフォール型開発と異なる帰納的なアジャイル型開発で行うことを顧客に理解されているか。	○	○	
		● AI システムの性能維持に対する継続改善の理解度 ・ AI システムの性能を維持するためには、継続的に学習を重ねて改善を図る必要があることを顧客に理解されているか。			○
		● 運用後の改善における検証方法への理解度 ・ 運用後の改善については、プログラムの修正ではなく、学習用データセットの追加学習／再学習やハイパーパラメータの調整等で行うこと顧客に理解されているか。			○
CE-4	データの量や質に対する認識の甘さ	● 顧客のビジネス課題に合致した学習用データセットの必要性に対する理解度 ・ 顧客は、AI モデルの性能向上に、顧客のビジネス課題に合致した学習用データセットが必要なことを理解しているか。また、そのような学習用データセットを保有しているか。	○	○	
		● AI モデルの学習に必要な学習用データセットの質、量への理解度 ・ AI システムによるビジネス課題解決には、多くの学習用データセットを網羅的に用意する必要があることを顧客に理解されているか。	○	○	
		● 運用中の入力データ傾向の変化に関する顧客の理解 ・ 運用中、学習時と異なる入力データ傾向になった場合、既存の AI モデルで正しく推論できなくなる可能性があることを顧客と合意し、理解しているか。また、運用中に入力データの傾向を常に監視し、顧客と情報を共有できているか。	○	○	○

表 7.6 Customer Expectation の留意事項（続き）

ID	観点	留意事項	PoC	開発	運用
CE-5	AI プロダクトの利用における法令上、倫理上の問題の有無、第三者のプライバシー等への配慮の必要性、AI プロダクトの利用が社会的に受容されている度合	● AI システムの著作権をはじめとする知的財産権等に関する契約での取り決め ・ AI システムに対する著作権をはじめとする知的財産権等の権利帰属及び利用条件について、顧客と契約で取り決めが行われているか。	○	○	
		● AI システムの著作権をはじめとする知的財産権等に関する理解度 ・ AI システムの著作権をはじめとする知的財産権等の権利帰属及び利用条件について、顧客に理解がされているか。	○	○	
		● AI システムに使用する顧客データのセキュリティの高さ、情報開示範囲及び取扱制限の明確化 ・ AI システムに使用するデータのセキュリティの高さや情報開示範囲、取扱制限は明確になっているか。	○	○	
		● AI システムに含まれるデータの権利に関する取り決め 学習用データセット、テストデータ、運用時の入力データ、出力データなど含めて AI システムに関わる直接的、間接的にすべてのデータについて、データの権利や利用ルールなどが明確になっているか。	○	○	
		● 運用後に改善した AI システムの著作権をはじめとする知的財産権等に関する契約での取り決め ・ 運用後に行われた AI モデルの変化（追加学習／再学習等）に対して、導入時の契約における権利帰属及び利用条件に抵触しないか。（例えば、学習モデルを開発・提供元が有する知的財産権が含まれる場合、利用者側でそのモデルを更新する場合の考慮がなされているか。） ・ 再学習等により運用後に改善した AI システムの知的財産等の権利帰属及び利用条件について、契約で取り決めが行われているか。 ・ 運用中に入力される顧客・ユーザーからのデータについて、権利帰属及び利用条件が明確になっているか。			○

表 7.7 Customer Expectation の留意事項（続き）

ID	観点	留意事項	PoC	開発	運用
CE-6	“合理的”説明を求める度合、“外挿”や“予測”をしたがる度合 “原因”や“責任(者)”を求めたがる度合 納得感を共感する風土や雰囲気、仕事の進め方の少なさ	<ul style="list-style-type: none"> ● AI モデルの説明困難性に対する理解度 <ul style="list-style-type: none"> ・AI モデルの処理した結果に対する説明の難しさを顧客に理解されているか。 	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
		<ul style="list-style-type: none"> ● AI モデルの説明困難性と精度に対する理解度 <ul style="list-style-type: none"> ・AI アルゴリズムの種類によって出力結果の説明根拠を示すことができるもの(二分木等)と、困難なもの(DeepLearning 等)があり、説明可能性は AI モデルの精度に反比例することがあることを顧客に理解されているか。 	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
		<ul style="list-style-type: none"> ● AI モデルの選択に対する理解度 <ul style="list-style-type: none"> [PoC 時][開発時]に選択した AI アルゴリズムおよび AI モデルに対して、その選択で良いと言うことを顧客側と合意しているか。その際、選択の根拠などの説明をして顧客が理解ができるか。 	<input type="radio"/>	<input type="radio"/>	
CE-7	責任の所在が明確か（開発側が事故の責任を負うとした契約）	<ul style="list-style-type: none"> ● 顧客・ステークホルダーの明確化 <ul style="list-style-type: none"> AI システムの [PoC 時][開発時][運用時] に必要となるステークホルダーを洗い出し、それぞれが AI システムにどのように関与するかを明確にしているか。 	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
		<ul style="list-style-type: none"> ● AI システムの出力に対する責任の明確化 <ul style="list-style-type: none"> ・確率的動作（確率的にもっともらしい解）の AI システムが出した結果により人的被害等が発生した場合、[PoC 時][開発時][運用時] の責任所在は明確になっているか。 	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

表 7.8 Customer Expectation の留意事項（続き）

ID	観点	留意事項	PoC	開発	運用
CE-8	顧客の協力・関与の度合いは高いか	● AI システムの開発に対する顧客の協力度及び関与度 ・ [PoC 時][開発時] に顧客の保有しているデータの提供や検証において、顧客は協力的か。	○	○	
		● 運用中の継続した顧客の協力・関与 ・ 運用中も継続して顧客と意見交換しながら進められているか。顧客からフィードバックを都度得られているか。			○

CE-1 の留意事項に対する説明

● AI によって解決したいビジネス課題の明確化< PoC/開発 >

< PoC >

以下のような技法を用いてビジネス課題を明確にする。

※ビジネス課題とは、市場不具合、工程内不良等のロスコスト、検査業務のコスト等、定量的に把握すること。

※直接的には観測できない場合、ブレイクダウンし、間接的に観測できるかも検討すること。

- ゴール指向分析
- マーケティング分析
 - 4C 分析（顧客価値/顧客コスト/コミュニケーション/利便性）
 - 3C 分析（市場・顧客/競合/自社）
 - SWOT 分析（強み/弱み/機会/脅威）
 - STP 分析（市場細分化/狙う市場/自社の立ち位置）
 - PEST 分析（政治/経済/社会/技術）
 - 4P 分析（製品/価格/流通/プロモーション）
- QC7 つ道具
 - パレート図、ヒストグラム、管理図、散布図、特性要因図、チェックシート、グラフ

<開発>

- 解決したい課題の内容が変化していないか監視をする。

留意事項を怠ったことによる影響

< PoC >

- PoC から次のフェーズ移行出来ない。
- 開発の中断。

<開発>

- 開発コストの増大。
- 課題が変化したことにより目的が達成できないシステムになってしまう。

● AI によるビジネス課題の解決可能性 <PoC/開発>

<PoC>

※顧客の目指すゴール設定が難しい場合、顧客/受託側双方のリスク回避のため、[PoC]、[開発時]の契約を分けることも検討すること。

※ AI でなくとも従来方法で(より安価に) 解決可能な場合がある。

- 顧客の抱えている課題とその解決までの道筋が妥当かを確認する。AI システムへの入力データおよび AI システムからの出力データは妥当か、適用先やターゲットの規模について検討は十分かを検討し、解決できない可能性があるリスクについて顧客に説明し合意を得る。ビジネス課題の解決のために、文献や適用事例などを評価し、顧客に説明を行う。

<開発>

- PoC の結果をふまえて AI を使わずに課題が解決できるかを検討する。(生データや訓練データを集めた結果、ルールベースでの対応でも課題を解決できる可能性がある)

留意事項を怠ったことによる影響

<PoC>

- PoC から次のフェーズ移行出来ない。
- 開発の中断。

<開発>

- 開発コストの増大。
- 課題が変化したことにより目的が達成できないシステムになってしまう。

● AI によって解決したいビジネス課題解決の効果 <PoC/開発>

<PoC>

- 「費用対効果」を考慮しながら「システムにどこまでやって欲しいのか」を定義する。
- AI システムの導入により自動化が期待できるレベルと、ビジネス課題の解決のために顧客が期待する自動化のレベルの認識を合わせる。(参考：自動運転における人と機械の協調 [3])

<開発>

- 費用対効果が変化していないか監視をする。
- 自動化のレベルが変化していないか監視をする。

留意事項を怠ったことによる影響

<PoC>

- PoC から次のフェーズ移行出来ない。
- 開発の中止。

<開発>

- 開発コストの増大。
- 課題が変化したことにより目的が達成できないシステムになってしまう。

● AI の性能維持に対する継続改善の理解度<PoC/運用>

<PoC>

- 性能維持のための契約はしているか。
- AI の性能維持に必要となる作業やコストを明確にする。反復的に再学習の実施、推論精度および環境依存性の維持・向上に必要なコストは考慮しているか。
- AI の性能維持における判断基準を明確にする。開発後(導入時)の AI 性能を基準にし、運用後はその基準値にて性能維持・向上を判断しているか。
- AI の期待効果に対して、各フェーズでターゲットとするメトリクスを明確にする。例えば、「不良削減 20%」という目標があっても、PoC でそれを指標にして検証することは難しいため、各フェーズのターゲットとするメトリクスを明確にする必要がある。
- 継続して改善しなければいけない用途で AI を使用するのか。
- 性能を維持するための仕組みは定義されているか。

<運用>

- AI を導入したことで期待した結果となっているか監視をする。

留意事項を怠ったことによる影響

<運用>

- 運用コストの増大。
- 運用がうまくいかず期待する結果からかけ離れていってしまう。

● AI によるビジネス課題解決の満足度<運用>

- 顧客満足度調査を実施する。※満足度の調査は、運用時に実施となるが、[PoC 時]、[開発時]にビジネス課題と活動の乖離しないようにすること。

留意事項を怠ったことによる影響

<運用>

- 目的に合致しないシステムが出来上がる。(顧客が満足しない)
- 以降の受注の機会を失う可能性がある。
- 運用コストの増大。
- 運用がうまくいかず期待する結果からかけ離れていってしまう。

CE-2 の留意事項に対する説明

● 確率的動作で出力される AI への理解度< PoC/開発/運用 >

< PoC >

- 確率的動作であるため、以下のような懸念があることを顧客に説明し、合意を得る。
 - 確率的判断を行うためには大量の学習用データセットが必要であり、収集にはコストがかかる。また、収集した学習用データセットから期待した結果が出ない場合もあり、その結果収集した学習用データセットが無駄になる可能性がある。
- 入力データやハイパーパラメータの調整にトライ＆エラーの取り組みとなる場合がある。調整に伴って、徐々に改善するという保証はないため、調整してこのまま続けるか、それとも中止とするかの判断が必要となる場合がある。
- 期待した結果が出ない場合の分析・解析・改善が難しい場合がある※ AI は人間と異なる思考で出力を出すため、人間には判別が容易なことも AI では難しかったり、その逆が起こりうる。
- AI は良品を不良品と判定したり不良品を良品と判定することがあり、その対策をシステム全体で行う必要があることを顧客に説明し、合意を得る。

<開発>

- 確率的動作であるため、入力データがテストデータと少しでも異なった場合の動作について幾らかのリスクがあるということを顧客に説明し、合意を得る。
- AI が良品と不良品の判定を誤る具体的なケースを顧客に説明し、合意を得る。

<運用>

- 人の判断と異なる意図しない判断をする可能性があるということ、また AI モデルが複雑であ

るほど、その原因特定は困難であるということについて顧客に説明し、合意を得る。

- AI モデルの更新の結果、従来の判定より悪くなる部分がある可能性があることを顧客に説明し、合意を得る。

留意事項を怠ったことによる影響

< PoC/開発 >

- 実現不可能な性能要求を受け、性能未達による開発中断等が発生する。※例、x x x の推論の正解率は 100% であること。
- 推論誤り時の対策が次々と必要になり開発費が膨れ上がる。

< 運用 >

- 想定外の問題は、莫大な損害となる。
- 推論誤りが発生した場合、その責任を問われる可能性がある。

● 確率的動作で出力された結果のリスク許容度 < PoC/開発/運用 >

< PoC/開発 >

- 確率的動作による出力結果のリスク（想定外・判定/推論の誤り）が許容できるかを検証＆合意を得る。必要であれば、確率的動作により発生しうる問題のリスク評価を行い対策を検討する。対策の内容によっては、システム実装に組み込む必要がある。
- システム全体の構成（AI が担う範囲（自動化のレベル）の明確化）と AI が誤った場合のハザード分析（FMEA 等を活用）を実施し、許容できるかを検証＆合意を得る。（例）以下のよ うな手法を活用する。
 - FMEA
 - FTA
 - HAZOP
- リスクを識別する。
- 優先度を決める。（優先度 = 発生確率 × 損害額）
- リスク対策（保有、回避、増加、共有、リスク源の除去、起こりやすさを変える、結果を変える）を立てる。

< 運用 >

- [PoC 時][開発時] のリスク対策が上手くいっているかフォローする。
- リスク対策が上手くいかず、顕在化した場合の対策（受容、回避、転嫁、軽減）を立てる。
- リスク対策していない問題が顕在化した場合の対策を立てる。
- 運用時においても、リスクが存在しないかリスク分析を行う。※分析の手順は [PoC 時]、[開

発時]と同じ。

- 運用時は気温等の環境面の変化も影響するためリスク分析を行う。

留意事項を怠ったことによる影響

<PoC/開発>

- 実現不可能な性能要求を受け、性能未達による開発中断等が発生する。※例、xxxの推論の正解率は100%であること。
- 推論誤り時の対策が次々と必要になり開発費が膨れ上がる。

<運用>

- 想定外の問題は、莫大な損害となる。
- 推論誤りが発生した場合、その責任を問われる可能性がある。

CE-3 の留意事項に対する説明

● アジャイル型ソフトウェア開発への理解度<PoC/開発>

<PoC>

※アジャイル型ソフトウェア開発については、PAも参照のこと。

- PoC時の目標(例えば正解率など)に対して、アジャイル的に開発するプロセスになることを顧客と合意を得る。

※この時、期間内に目標未達の場合どうするかも合意しておく。

- シンプルなV字プロセスではなく、試行→評価を繰り返し実施する必要があることについて顧客と合意を得る。
- PoC時の目標(例えば正解率など)を達成するためには新たな訓練データ収集や訓練データの質を上げる必要が出てくることを顧客と合意する。
- PoC時に何度もトライ＆エラーを繰り返しながら訓練データを増やしていき段階的に精度を上げていくことを顧客と合意する。

<開発>

※アジャイル型ソフトウェア開発については、PAも参照のこと。

- 開発時の目標に対して、PoC時と同じく顧客と合意する。
- AIコンポーネントのリリース時期やその性能について顧客と合意する。※特にAIコンポーネントと周辺の開発が別会社の場合。
- 運用時の事を考慮して、AIシステム実装に組み込む。※ログの取り方、AIコンポーネントの

更新の仕方、等。

留意事項を怠ったことによる影響

- 繰り返し実施しても正解率が上がらず、次のステップに進めない。その結果、計画の遅延や開発コストが増大する。

● AI システムの性能維持に対する継続改善の理解度<運用>

● 運用後の改善における検証方法への理解度<運用>

- 運用時に得られる推論対象の訓練データは、経年劣化、設置環境の変化、世情の変化等で推論の性能（正解率など）が劣化する可能性がある。このため、適宜、AI モデルを改善する必要がある。
- 運用時の改善の検証は、運用時のテストデータで行う必要がある。
- 性能維持に対する継続改善および検証を行うに当たり、以下を合意しておく。
 - 運用時の改善について、客先に説明＆合意および契約を行う。
 - 運用時の訓練データ収集の扱いについても契約上明記しておく。

留意事項を怠ったことによる影響

<PoC/開発>

- 実施した結果、目標（正解率など）に届かず開発の中止が発生する。
- 学習用データセット収集のコスト増により開発中止となる。
- 繰り返し実施しても正解率が上がらず、次のステップに進めない。その結果、計画の遅延や開発コストが増大する。

<運用>

- 客先での生データを元にした訓練データを性能改善に使えない場合、性能改善の要求を満たせない。

CE-4 の留意事項に対する説明

● 顧客のビジネス課題に合致した訓練データの必要性に対する理解度<PoC/開発>

- ビジネス課題と集めた生データの間に関係性が必要であることを理解してもらう。
 - ゴール指向分析
 - 特性要因分析
 - 散布図、相関係数などから統計的に有意であるか（無相関検定）

留意事項を怠ったことによる影響

- 十分な訓練データが用意できないと、(特に正解率の) 性能に対して未達となる。

● AI モデルの学習に必要な訓練データの質、量への理解度< PoC/開発 >

※ DI-1、DI-2 参照

留意事項を怠ったことによる影響

- 十分な訓練データが用意できないと、(特に正解率の) 性能に対して未達となる。

● 運用中の入力データ傾向の変化に関する顧客の理解< PoC/開発/運用 >

< PoC/開発 >

- 事例を紹介し、5M+E(人/設備/方法/材料/検査・測定+環境) により入力データの傾向が変わることを理解してもらう。
- 各説明変数にて推定される傾向変化の因果関係を整理し、その可能性を説明する。※設置環境の変化、経年劣化、等。

< 運用 >

- 運用中の入力データ内容を監視し、学習時とかけ離れた入力データとなっていないか監視する仕組みを構築する（必要か要議論）。

留意事項を怠ったことによる影響

< PoC/開発/運用 >

- 十分な訓練データが用意できないと、(特に正解率の) 性能に対して未達となる。

< 運用 >

- 学習時の異なる傾向の入力データで推論をし、結果が異なる場合、係争が発生する可能性がある。

CE-5 の留意事項に対する説明

- AI システムの著作権をはじめとする知的財産権等に関する契約での取り決め、
- AI システムの著作権をはじめとする知的財産権等に関する理解度、
- AI システムに使用する顧客データのセキュリティの高さ、情報開示範囲及び取扱制限の明確化
- AI システムに含まれるデータの権利に関する取り決め< PoC/開発 >

※法令の改正や解釈の変更等によりデータを利用可能な範囲等に変更が生じる可能性もあり、法改正等の動向を適宜確認する必要がある。

※データに所有権は生じないが、著作権等の権利が生じる場合がある（もっとも、著作権法 30 条の 4 第 2 号により、第三者の著作物であっても学習のための利用が認められる）。また、契約や法

令（個人情報保護法等）の規制によりデータの利用に一定の制約がかかりうる。データの利用は、かかる制約をふまえて適切に検討し、顧客と一定の合意等を行っておく必要がある。

※また、国外のデータを扱う場合等には、国外の法令が適用される可能性があることにも留意する必要がある。

- PoC 時に顧客から提供された生データおよび学習用データセットの権利は誰に帰属するか、またその範囲を定義する。※例えば
 - PoC 後も、AI モデルの性能改善のために生データおよび学習用データセットを使用して良いか。
 - 新たな提案活動において生データおよび学習用データセットを使用して良いか。
- 推論用の入力データとして使用する情報は提供者の許諾を得ているか。また、推論結果の出力データは倫理的に問題ないかについて確認し、顧客と合意を得る。
- データの管理方法、保持期間、廃棄方法について顧客との認識を共有する。
- 以下観点で問題の有無を共有＆対策方針の策定
 - セキュリティ確保：AI システムの頑健性及び信頼性を確保すること。
 - 安全保護の原則：AI システムが利用者及び第三者の生命・身体の安全に危害を及ぼさないように配慮すること。
 - プライバシー保護の原則：AI システムが利用者及び第三者のプライバシーを侵害しないように配慮すること。
 - 倫理の原則：人間の尊厳と個人の自律を尊重すること。
 - アカウンタビリティの原則：利用者等関係ステークホルダーへのアカウンタビリティを果たすこと。

留意事項を怠ったことによる影響

- 以下のような影響が考えられる。
 - 第三者の知的財産権を侵害することによる紛争の発生
 - 個人情報保護法に違反したことを理由とした行政指導等の不利益
 - 社会的な批判、ブランド失墜の可能性等

●運用後に改善した AI システムの著作権をはじめとする知的財産権等に関する契約での取り決め<運用>

※法令の改正や解釈の変更等によりデータを利用可能な範囲等に変更が生じる可能性もあり、法改正等の動向を適宜確認する必要がある。

※データに所有権は生じないが、著作権等の権利が生じる場合がある（もっとも、著作権法 30 条の 4 第 2 号により、第三者の著作物であっても学習のための利用が認められうる）。また、契約や法令（個人情報保護法等）の規制によりデータの利用に一定の制約がかかりうる。データの利用は、か

かる制約をふまえて適切に検討し、顧客と一定の合意等を行っておく必要がある。

※また、国外のデータを扱う場合等には、国外の法令が適用される可能性があることにも留意する必要がある。

- AI システムはリリース後の AI モデル更新の事を含めて権利は誰に帰属するか、またその範囲を定義する。
- 運用時に収集されたデータおよび更新した AI モデルの権利は誰に帰属するか、またその範囲を定義する。※例えは、
 - 運用時にモニタリングする必要がある場合、AI コンポーネントへの入力データおよび出力データを閲覧、加工しても良いか。
 - AI モデルの更新は誰が行うのか？更新した AI モデルは誰が責任を負うのか。
 - データの管理方法、保持期間、廃棄方法について顧客との認識を共有する。

留意事項を怠ったことによる影響

- 以下のような影響が考えられる。
 - 第三者の知的財産権を侵害することによる紛争の発生
 - 個人情報保護法に違反したことを理由とした行政指導等の不利益
 - 社会的な批判、ブランド失墜の可能性等

CE-6 の留意事項に対する説明

- AI モデルの説明困難性に対する理解度
 - AI モデルの説明困難性と精度に対する理解度 < PoC/開発/運用 >
- < PoC/開発 >

- 性能と説明し易さのトレードオフ表等を作成し、顧客に説明する。※説明のしづらい事例などを示す。(行列計算式など。)
- 作成した AI モデルで推論した結果、その過程の説明が困難であるケースがあることを顧客に説明する。そのようなケースの場合にどう対応するのか決めておく。

<運用>

- 説明に必要なデータが収集できていることを監視する。

留意事項を怠ったことによる影響

< PoC/開発/運用 >

- 説明可能性について、合意を得ていない場合、顧客より説明要求が発生した場合、回答出来ない場合がある。※回答しても顧客が納得しない可能性がある。

● AI モデルの選択に対する理解度< PoC/開発 >

- AI モデルの選択肢と選択した根拠を定義し顧客と合意を得る。

留意事項を怠ったことによる影響

- AI モデルに起因して十分な効果が得られないなど、AI モデルの再選定が必要になった場合、責任の所在を巡って係争に発展する可能性がある。

CE-7 の留意事項に対する説明

● 顧客・ステークホルダーの明確化< PoC/開発/運用 >

※ [PoC 時]、[開発時]、[運用時] の各契約時に決めておく必要がある。

- システム全体の構成 (AI が担う範囲と既存システムとの接続部分) を定義し、ステークホルダーの特定と関与の度合いを明確化する。必要があれば、リスク管理を行う。※ CE-2 参照。

留意事項を怠ったことによる影響

< PoC/開発/運用 >

- ステークホルダーを特定していない場合、以下のような事象が (1つまたは複数) 発生し、手戻りが発生し、PoC・開発・運用の中止が起こりうる。
 - ビジネス課題の定義誤り。
 - 使用するデータの誤り、または精度不足。
 - 使いたいデータが使えない。など
- AI システムの出力に対する責任の明確化をしていない場合、係争に発展する。

● AI システムの出力に対する責任の明確化< PoC/開発/運用 >

- 責任所在を明確にし、関係するステークホルダーと合意する。※ AI コンポーネントへの入力データの責任、推論結果である出力データに対する判断の責任、再学習用データセットの準備の責任など。
- また、推定される事象に対し、リスク管理を行う。※ CE-2 参照。
- AI システムの品質目標設定または PoC の完了条件の設定が難しい場合は、顧客、開発双方のリスク回避のため、PoC、開発および運用の契約を分け、各契約のゴールを明確にする。※例えば SOW (Statement Of Work) を作成し、合意を得る。

留意事項を怠ったことによる影響

- ステークホルダーを特定していない場合、以下のような事象が (1つまたは複数) 発生し、手

戻りが発生し、PoC・開発・運用の中止が起こりうる。

- ビジネス課題の定義誤り。
- 使用するデータの誤り、または精度不足。
- 使いたいデータが使えない。など
- AI システムの出力に対する責任の明確化をしていない場合、係争に発展する。

CE-8 の留意事項に対する説明

● AI システムの開発に対する顧客の協力度及び関与度< PoC/開発 >

< PoC >

- 各フェーズの目的、顧客が用意する必要のあるデータ（生データまたは学習用データセット）および懸念点についてプロジェクト計画書に記載し、顧客と合意を得た上で進める。
- 期待した結果がでないこと、期待したことがないことに対して、顧客とリスクの共有とその対策の合意を得る。※リスク分析、対策立案については CE-2 参照。
- PoC 目的（顧客提供データの実用性判断）を顧客と共有し合意を得る。
- AI 学習に必要となる顧客保有データを継続的に提供してもらえることを確認する。※ CE-4 にて認識あわせしたデータの量や質の継続提供。

< 開発 >

以下について顧客に説明し合意を得る。

- 顧客提供データ（PoC 実験データ）に対して実際の運用データは想定の範囲内か確認する。
- AI モデル変更が発生する場合の与える影響について確認する。
- 運用時の AI モデル更新手順や、事前チェックの方法について顧客に確認する。（顧客提供データでの結果を持ってリリースするなど）
- AI 学習に必要となる顧客保有データを継続的に提供してもらえることを確認する。※ CE-4 にて認識あわせしたデータの量や質の継続提供。

留意事項を怠ったことによる影響

< PoC >

- 以下のような事象（1つまたは複数）の発生による、手戻りや PoC の中止が起こりうる。
 - PoC 目的の定義誤り。
 - リスクに対する認識の齟齬。
 - 使いたいデータが使えない。など

< 開発 >

- AI をターゲットに実装できない。所望の性能を達成できない。要件定義、PoC からのやり直しとなる。

●運用中の継続した顧客の協力・関与<運用>

- 運用結果のフィードバック方法や、例外データの取得方法について、事前に共有しておく。また急なパッチ対応は内容(頻度、優先度等)を定義し、顧客と共有しておく。
- 再学習のためのラベル付を現場の人員が行う必要があるか、確認の上、顧客と(追加の費用が発生することの)合意を得る。
- AI 学習に必要となる顧客保有データを継続的に提供してもらえることを確認する。※ CE-4 にて認識あわせしたデータの量や質の継続提供。

留意事項を怠ったことによる影響

- AI の精度が低下することにより、ビジネス課題に対する効果が減衰する。

7.6.3 Data Integrity

産業用システムは、環境依存性により、工場などのクローズドなシステムであっても、データの変動要因が多数存在する。どのように変更管理を厳密に実施したとしても、設備の自然劣化、調達材料の変化、自然環境天候変化といった予期せぬ(未経験の)変動要因に対し、PoC や開発時にすべてを仕様化し扱うべきデータすべてを分類・検証することは不可能である。そのため、段階的に対象を事前に定義し、データのバリエーションや内容確認が必要となる。

表 7.9 Data Integrity の留意事項

ID	観点	留意事項	PoC	開発	運用
DI-1	学習データの量の十分性	● AI の学習に必要なデータ量の確保 ・課題解決のための検証 (PoC) に必要なデータの量は揃っているか。	○		
		・運用に向けた開発に必要な学習用データセットの量は揃っているか。		○	
		● 交差検証や汎化性能等に使用するデータの確保 ・学習だけでなく交差検証や汎化性能等が確かめられるデータ量か。	○	○	
		● 「かさ増し」したデータに対する評価 ・かさ増しの手法や追加されたデータが適切であったか評価してあるか。	○	○	
		● 運用時に得られたデータを使ったデータの「かさ増し」に対する評価 ・「かさ増し」に対する開発時の仮定に対し、運用時に得られる追加データの分布やラベリングに対して適切であったかを評価しているか。			○
DI-2	学習データの妥当性 学習データの要件適合性 学習データの適正性 学習データの複雑性 学習データの性質の考慮 学習データの値域の妥当性 学習データの法的適合性	● 学習に使用するデータのビジネス課題との整合度 ・課題解決に繋がるデータが顧客より提供されているか。もしくは生成、獲得することが可能か。	○	○	
		● 学習用データセットの質の確保 ・想定する母集団を定義した上で、網羅的にサンプルデータが取得され、偏り等はないか。	○	○	
		● 訓練データの特性評価 ・データに特性がある場合、選択バイアス、情報バイアス、交絡の問題・リスクを評価したか。外れ値や欠損値の除去・訂正の根拠、措置方法について、受容・排除などのポリシーにもとづいて行っているか。	○	○	

表 7.10 Data Integrity の留意事項（続き）

ID	観点	留意事項	PoC	開発	運用
DI-2(続き)		<ul style="list-style-type: none"> ●運用データの必要性 <ul style="list-style-type: none"> ・運用時は、開発時にはなかったデータが得られる可能性があるため、実運用でしか収集できないデータを記録する仕組みを構築しているか。 ・運用で発見したエラーや多様性に対応したデータを確保したか。 			<input type="radio"/>
		<ul style="list-style-type: none"> ●入力データの特性評価 <ul style="list-style-type: none"> ・運用中のデータには、導入時と異なる偏りが存在するか。また、その背景を解析しているか。 ・外れ値や欠損値の除去・訂正の根拠、措置方法について、受容・排除などのポリシーにもとづいて行っているか。 システム維持を想定できているか。 			<input type="radio"/>
		<ul style="list-style-type: none"> ●ビジネス課題（現象）に対するデータ定義（想定するモデル）の複雑さ <ul style="list-style-type: none"> ・ビジネス課題のモデル化に際して、学習用データセットの説明変数の数・因果関係の数が複雑過ぎないか、もしくは単純すぎないか。また、多重共線性は考慮しているか。 	<input type="radio"/>	<input type="radio"/>	
		<ul style="list-style-type: none"> ●学習用データセットの入手ルート及び管理の妥当性 <ul style="list-style-type: none"> ・学習用データセットの入手・取得ルートは明確になっているか、データの管理方法に不備はないか。 	<input type="radio"/>	<input type="radio"/>	
		<ul style="list-style-type: none"> ●ラベルや正解値が正しくつけられているか <ul style="list-style-type: none"> ・学習用として、正解値も含めて妥当なデータセットになっているか。（識別問題ならラベル、回帰問題なら値など問題によってつけるべき正解値は変わる） 	<input type="radio"/>	<input type="radio"/>	

表 7.11 Data Integrity の留意事項（続き）

ID	観点	留意事項	PoC	開発	運用
DI-3	検証用データの妥当性	●交差検証や汎化性能等に使用する学習用データセットの独立性 ・交差検証や汎化性能等に使用する訓練データとテストデータを独立して分離・管理しているか。	○	○	
		●再学習での交差検証や汎化性能等に使用する学習用データセットの独立性 ・再学習、追加学習時時に交差検証や汎化性能等に使用する訓練データとテストデータを独立して分離・管理しているか。			○
DI-4	オンライン学習の影響の考慮	●外れ値に対する学習除外の仕組みの実現 ・運用時にオンライン学習を行い、モデルをインクリメンタルに追加・置き換える場合は、信頼できるデータ区間を定義し、予期せぬデータによる学習を防ぐ仕組みを構築しているか。		○	
		●外れ値の監視 ・モデルを更新するデータが想定したデータ区間を外れているかを監視するなど、入力データの質をパトロールしているか。			○
DI-5	データ処理プログラムの妥当性	●データの前処理等に利用するプログラムの妥当性確認 ・データ処理を行うプログラムが妥当であることを確認しているか。また確認した結果が残っており、示せるか。	○	○	

DI-1 の留意事項に対する説明

● AI の学習に必要なデータ量の確保 < PoC/開発 >

一般的に、学習に使用するデータが多いほど母集団に近づくが、学習によっては、訓練データ量をいくら増やしても性能が飽和状態に陥り、検証効率が低下する場合がある。このため、PoC フェーズでは、開発フェーズでの検証にかかる費用や時間の抑制を目的に、期待する学習性能を獲得するために必要なデータ量を評価することも重要である。また、解決したい課題によってアルゴリズムが変わると、その際に必要な学習用データの量も異なる（参考：scikit-learn algorithm cheat sheet [1]）。

留意事項を怠ったことによる影響

学習用データセットが少ないと、期待するモデルが構築できず、運用時に正しい分類や回帰の結果が得られない。一方学習用データが多すぎると、学習に時間を要し、データ量に比例して記録容

量も必要になる。

● 交差検証や汎化性能等に使用するデータの確保 < PoC/開発 >

多くの場合、学習用データセットを訓練用とテスト用に分けて利用するため、データをすべて訓練に使えるわけではない。このため、最良のモデルを構築するための適切なデータ量を確保することが必要となる。

留意事項を怠ったことによる影響

学習のデータが少ない場合と同様、期待するモデルが構築できず、運用時に正しい分類や再起の結果が得られない。

● 「かさ増し」したデータに対する評価 < PoC/開発 >

PoC の学習用データセットが不十分な場合、かさ増しをしてデータ量を増やすことがある。このときにかさ増しの手法や増やしたデータが妥当なものかを確認することが重要である。

留意事項を怠ったことによる影響

想定する母集団と異なる傾向のデータをかさ増しで生成すると、汎化性能が悪くなることがある。

● 運用時に得られたデータを使ったデータの「かさ増し」に対する評価 < 運用 >

学習用データセットが少ない場合、データを少し加工して「かさ増し」をすることがある。かさ増しとはデータをずらす、反転する、ノイズを加えるなどの加工を加えたものである。このようなかさ増しをしたデータでモデルが構築可能かを検討する。また、かさ増しにおいては、画像検査システムを例にとると、良品と不良品のサンプル画像の他、カメラの状態や周辺状況など、AIへの入力データに影響を及ぼす要因を想定して行うとよい。なお、かさ増しには以下のようない法がある。
画像：Data Augmentation、信号：各種ノイズ付与

留意事項を怠ったことによる影響

誤ったかさ増し方法を適用すると、返って汎化性能が悪くなることがある。「かさ増し」したデータが適切であっても、追加学習により、必ずしも性能が改善されることは限らず、その場合には、再学習による評価を検討する。

DI-2 の留意事項に対する説明

● 学習に使用するデータのビジネス課題との整合度 < PoC/開発 >

訓練データの質を高めるためにデータの追加や変更を行う場合は、試行錯誤的なやり方となることが多いため、ロールバックや再現ができるように、変更履歴を管理することが重要である。(PA-3)

も参照)「課題解決につながるデータ」の定義は難しい。AI の利用を検討するにあたって明確化されているはずの「解決すべき課題」に対して、学習と評価の繰り返しによって「課題」を解決可能なデータを試行錯誤的に特定していくアプローチが現実的である。

留意事項を怠ったことによる影響

訓練データに存在しない場合、判定が困難になる。

●学習用データセットの質の確保 < PoC/開発 >

精度の高いモデル構築のために、正例だけでなく負例も訓練データとして扱うと良い。訓練データの外れ値や欠損値、揺らぎ、重複等が多い場合、正しく学習できない可能性も高いため、訓練データの前処理（クレンジング）の実施が必要となる。同様に、偏りの正規化のために数値の対数化や正規化なども検討するとよい。

留意事項を怠ったことによる影響

負例が少ない場合や外れ値などがあると、モデルの精度が低下する。

●訓練データの特性評価 < PoC/開発 >

ほとんどの場合、母集団のすべてを入力とした学習用データセットを揃えることは不可能である。そのため、母集団を想定して、その中のサンプルデータを利用する。サンプルデータに偏りがある場合、学習したモデルも偏り※に影響される。想定した母集団を適切に表すサンプルデータを利用する。(※データの偏りは、ヒストグラムや主成分分析、t-SNE 分析、散布図などで確認できる。)

留意事項を怠ったことによる影響

訓練データの偏りがある場合、未学習のデータが多いモデルとなるため、汎化性能が悪くなることが考えられる。

●運用データの必要性 < 運用 >

学習できていない入力データに対しては、出力が定義できない。例えば、部品 A、部品 B、部品 C を判別するシステムで部品 D が入力された場合に出力が A, B, C のいずれになるかが予測できない。部品 A が入力されたときも、側面や背面の訓練データがなかった場合、正しく A が出力されないかもしれません。したがって、訓練データは、運用時の入力データに対する網羅性を持つことが重要である。ただし、完全な網羅性を持つことは現実的に困難であり、運用時に未学習の入力データが入力されることもあるので、その場合の対処法も考慮する。また、どれにも当てはまらない Unknown クラスを定義することも検討した方が良い。

留意事項を怠ったことによる影響

訓練データに存在しない場合、判定が困難になる。

●入力データの特性評価 <運用>

運用を進めると、学習したときと異なる入力データの傾向になる場合がある。その場合、モデルを更新しないと期待した結果が得られない可能性が高い。

留意事項を怠ったことによる影響

運用を進めると、学習したときと異なる入力データの傾向になる場合がある。その場合、モデルを更新しないと期待した結果が得られない可能性が高い。

●ビジネス課題（現象）に対するデータ定義（想定するモデル）の複雑さ < PoC/開発>

学習に使用するデータの種類(因子)が多く存在する場合、学習結果への影響度が小さい因子や因子間の相関性が高いデータが含まれている可能性がある。このようなデータを用いて学習を行った場合、学習に使用するデータ量が増加し、検証コストも増加する。このため、PoC 時では、シミュレーションや多変量解析等を利用して、学習結果への影響度が高い因子を調べることで、データの絞り込みを行い、検証コストの抑制を検討する。

留意事項を怠ったことによる影響

すべての因子を使った場合、学習時間が増大したり、メモリ等のリソースを使いすぎる場合がある。また、検証にも時間がかかる。

●学習用データセットの入手ルート及び管理の妥当性 < PoC/開発>

学習用データの入手や管理にも気を配る必要がある。顧客の同意無しに得られたデータではないか、プライバシーに関わるデータを適切に管理しているか等である。

留意事項を怠ったことによる影響

学習用データセットの入手方法や管理方法が不適切の場合、社会的な問題や倫理的な問題に繋がる恐れがある。

●ラベルや正解値が正しくつけられているか < PoC/開発>

学習用データセットに正解ラベルを付与する場合、正しいラベルになっていなければならない。ラベルが誤っていると、モデルの誤りに繋がる。

留意事項を怠ったことによる影響

ラベルに誤りが多い場合も汎化性能が悪くなる。

DI-3 の留意事項に対する説明

●交差検証や汎化性能等に使用する学習用データセットの独立性 < PoC/開発>

機械学習では、学習用データセットの数やバリエーションが少ない場合や 1 訓練データの学習回

数が長すぎた場合、モデルが過学習に陥る可能性がある。過学習を適切に評価するため、学習に使用するデータと交差検証や汎化性能に使用するデータを完全に分離・管理しておく。データを分けるために、データの管理方法も明確にしておくことが大切である。

留意事項を怠ったことによる影響

データの管理方法が適切でないと、訓練データと検証データが混じり、モデルの汎化能力を適切に評価できない。

●再学習での交差検証や汎化性能等に使用する学習用データセットの独立性 <運用>

再学習／追加学習においても、訓練データと検証データを適切に分離しておく必要がある。

留意事項を怠ったことによる影響

データの管理方法が適切でないと、訓練データと検証データが混じり、モデルの汎化能力を適切に評価できない。

DI-4 の留意事項に対する説明

●外れ値に対する学習除外の仕組みの実現 <開発>

データを1件ずつ更新するオンライン学習を運用時に取り入れる場合、異常なデータでモデルが劣化することを防ぐため、期待するデータ区間を定義し、予期せぬデータを学習しないなど仕組みを構築する必要がある。同時に、データの傾向を確認できる仕組みを用意し、どのようなデータでモデルを更新したかを記録できるようにしておく。

留意事項を怠ったことによる影響

仕組みがないと、運用中に異常なデータが大量にあっても異常なデータに気づかず、異常なデータでモデルが更新されてしまう。

●外れ値の監視 <運用>

運用時にモデルを更新する際、更新に使うデータの傾向を監視する。データの傾向が変わる場合やノイズやその他の理由で異常なデータが入力される場合もある。このような場合にサービス利用者に通知し、学習を継続するか、止めるかなど対応を選択できるようにする。

留意事項を怠ったことによる影響

運用中に異常なデータが大量にあった場合、そのデータによってモデルが劣化する可能性がある。

DI-5 の留意事項に対する説明

●データの前処理等に利用するプログラムの妥当性確認 < PoC/開発 >

データの前処理や作成・過去を行うプログラムを利用することがあるが、このプログラムが正しく動くことを確認しておく。一般的には前処理プログラムはルールベースのプログラムであるため、一般のソフトウェア開発でおこうなうテストを用いて妥当性を確認する。また、確認した結果は記録し、後から確認できるように残しておく。

留意事項を怠ったことによる影響

AI モデルの学習時に誤った学習データセットを使うこととなり、正しく学習できない。データ処理プログラムの妥当性を事前に確認しておかないと、モデルが意図しない動作・性能をした際の原因切り分けが困難になる。

7.6.4 Model Robustness

Data Integrity で述べた制約条件から、モデルのロバスト性能の評価に必要な網羅性を収集したデータのみで数理的に担保することは難しい。そこで、多様な収集条件（システムとして備えなければならない外乱・パラメータなどの多様性条件）下でデータ収集し、直接利用ないし加工データで学習やロバスト性評価を実施することになる。直接データや加工データを利用する妥当性は、System Quality に基づいた評価を伴う。（例えば、外乱によるデータのばらつきを加工して得る場合、加工結果による System Quality への影響範囲やリスクにもとづき、加工方法の実証の厳密さを決める）

表 7.12 Model Robustness の留意事項

ID	観点	留意事項	PoC	開発	運用
MR-1	モデルの精度の十分性 ・正答率、適合率、再現率、F 値といった推論性能に関する評価指標の値は、要求に対して十分か	<ul style="list-style-type: none"> ●指標値の仮仕様定義 <ul style="list-style-type: none"> ・正答率・適合率などの指標値の説明を顧客に行っているか。PoC 終了までに顧客の要求と最適な指標値が整理できているか。 ・PoC 時点での顧客の指標値要求（正答率は**%以上 etc）をヒアリングできているか。 ・PoC 終了までに、顧客の要求を整理できているか。 	○		

表 7.13 Model Robustness の留意事項（続き）

ID	観点	留意事項	PoC	開発	運用
MR-1(続き)		<ul style="list-style-type: none"> ● 学習結果の妥当性 <ul style="list-style-type: none"> ・学習後の正答率、損失関数の残差は、十分に収束しているか。 ・適合率、再現率、F 値は目標に達しているか。 		<input type="radio"/>	
		<ul style="list-style-type: none"> ● 運用後の AI システム動作の妥当性 <ul style="list-style-type: none"> ・運用後、性能に影響を与える要因を抽出し、マージンを持たせた性能目標としているか。 ・性能劣化の検出を人間もしくは AI システムが判断する設計になっているか。 			
MR-2	モデルの汎化性能の十分性、モデルの評価の十分性、モデルの検証の十分性 <ul style="list-style-type: none"> ・汎化性能は確保されているか ・(AUROC といった) 精度以外のモデルのよさを表す指標についても適切な指標を選定し十分に評価したか ・十分に交差検証などを行ったか 	<ul style="list-style-type: none"> ● 汎化性能の調査 <ul style="list-style-type: none"> ・どのような汎化性能の測定が適切か、顧客と議論・整合が取れているか。 	<input type="radio"/>		
		<ul style="list-style-type: none"> ● 汎化性能の目標 <ul style="list-style-type: none"> ・汎化性能の目標値を明確に定めているか。 ・学習後の AI モデルの汎化性能は、学習時の正答率と比較して著しく劣化していないか。 		<input type="radio"/>	
		<ul style="list-style-type: none"> ● 汎化性能を測定する方法 <ul style="list-style-type: none"> ・汎化性能を測定する方法を決めているか。交差検証を利用する際、利用する学習用データセットのバリエーションを確保しているか。 		<input type="radio"/>	
		<ul style="list-style-type: none"> ● 運用時の交差検証方法を定義 <ul style="list-style-type: none"> ・学習用データセットのバリエーションが増えた際でも検証ができるように、交差検証の方法を決めているか。 			<input type="radio"/>
MR-3	学習過程の妥当性 <ul style="list-style-type: none"> ・学習は適切に進行したか ・局所最適に陥っていないか 	<ul style="list-style-type: none"> ● 学習過程の妥当性 <ul style="list-style-type: none"> ・学習後の正答率、損失関数の残差は、十分に収束しているか。 ・学習過程の正答率及び損失関数の残差は、異常な変化を示していないか。 	<input type="radio"/>	<input type="radio"/>	
		<ul style="list-style-type: none"> ● 再学習時の学習過程の妥当性 <ul style="list-style-type: none"> ・学習後の正答率、損失関数の残差は、十分に収束しているか。 ・学習過程の正答率及び損失関数の残差は、異常な変化を示していないか。 			<input type="radio"/>

表 7.14 Model Robustness の留意事項（続き）

ID	観点	留意事項	PoC	開発	運用
MR-4	モデル構造の妥当性 ・適切なアルゴリズムかどうかの検討は行ったか ・適切なハイパーサーバーパラメータかどうかの検討は行ったか	<ul style="list-style-type: none"> ● AI モデルの構造の妥当性 <ul style="list-style-type: none"> ・選択した AI アルゴリズムおよび蒸留有無の選択根拠、ハイパーサーバーパラメータの設定根拠は明確になっているか。顧客にアルゴリズムの選択根拠を説明・合意できているか。 	<input type="radio"/>	<input type="radio"/>	
		<ul style="list-style-type: none"> ● ハイパーサーバーパラメータの記録 <ul style="list-style-type: none"> ・どのようなハイパーサーバーパラメータを設定して検証を行ったか記録を残しているか。ハイパーサーバーパラメータごとの AI モデルの性能の差について顧客に説明できているか。 ・ハイパーサーバーパラメータの設定が AI モデルの性能に影響を及ぼすことを顧客が理解しているか。 			
		<ul style="list-style-type: none"> ● 再学習の際のハイパーサーバーパラメータの記録 <ul style="list-style-type: none"> ・どのようなハイパーサーバーパラメータを設定して再学習を行ったか記録を残しているか。 			<input type="radio"/>
MR-5	モデルの頑健性 ・ノイズに対して頑健か	<ul style="list-style-type: none"> ● AI モデルに影響を及ぼすノイズの洗い出し <ul style="list-style-type: none"> ・AI に影響を与えるノイズ候補の洗い出しを行っているか。具体的には、誤差因子の選定とそれとの与える影響解析を行っているか。 	<input type="radio"/>		
		<ul style="list-style-type: none"> ● AI モデルのノイズ耐性（ロバスト性）の妥当性 <ul style="list-style-type: none"> ・ノイズ候補により、AI モデルの性能が著しく劣化することはないか。 			<input type="radio"/>
MR-6	モデル更新に対する検証の十分性、モデル陳腐化への考慮	<ul style="list-style-type: none"> ● 再学習における性能劣化の許容度 <ul style="list-style-type: none"> ・訓練データの特性変化や出力の追加等により再学習を行った結果、再学習前の性能に対する劣化は許容可能か。 			<input type="radio"/>
		<ul style="list-style-type: none"> ● AI モデルの自動更新・配備の検査内容の十分性 <ul style="list-style-type: none"> ・AI モデルの更新を手動ではなく自動で実施する際に、AI モデルの特性変化や性能変化が許容範囲であることを十分検査できるか。 			<input type="radio"/>
MR-7	プログラムとしてのモデルの適切性	<ul style="list-style-type: none"> ● 外部ライブラリのテスト <ul style="list-style-type: none"> ・システムを評価するときに、外部ライブラリに対する単体テストやシステムテストを実施しているか。 		<input type="radio"/>	
		<ul style="list-style-type: none"> ● 外部ライブラリの責任範囲 <ul style="list-style-type: none"> ・ライブラリのサプライヤとの間で、不具合に対する責任範囲は明確になっているか。 			

MR-1 の留意事項に対する説明

● 指標値の仮仕様定義 < PoC >

- 指標値の定義を顧客に説明しているか。

留意事項を怠ったことによる影響

- 開発した AI システムの性能と、顧客の要求との不整合が発生する。

● 学習結果の妥当性 < 開発 >

- 学習時に、精度を図る指標(適合率・再現率・F 値)を算出する仕組みを用意する。作成された AI システムは、要求する精度を担保しているか。

留意事項を怠ったことによる影響

- 見過ぎ・見逃しが、顧客の要求に沿わない AI システムを作成してしまう。

● 運用後の AI システム動作の妥当性 < 運用 >

- 様々な特性の変化を定期的に監視し、その都度精度の指標を算出し直すことで、AI システムの妥当性を確認する運用を行っているか。

留意事項を怠ったことによる影響

- AI システムの性能が劣化していることに気付かないまま、運用を続けてしまう。

MR-2 の留意事項に対する説明

● 汎化性能の調査 < PoC >

- 汎化性能とは何か、を顧客に説明しているか。
- 汎化性能を向上させる手法(以下、例を記載)と特性を顧客に説明しているか。
 1. 正則化
 2. 交差検証 など

● 汎化性能の目標 < 開発 >

- PoC で検証した汎化性能を顧客に説明できているか。
- その上で、汎化性能を要求仕様として明確に定めているか。

留意事項を怠ったことによる影響

- 充分な汎化性能を持たず、過学習された状態で AI モデルが運用される可能性がある。

●汎化性能を測定する方法 <開発>

- 交差検証では、テストデータを充分に確保して検証を行っているか。

留意事項を怠ったことによる影響

- 充分な汎化性能を持たず、過学習された状態で AI モデルが運用される可能性がある。

●運用時の交差検証方法を定義 <運用>

- 交差検証を定期的に行い、AI モデルを更新するためのフローを定義しているか。顧客に説明しているか。

留意事項を怠ったことによる影響

- AI モデルの性能が劣化していることに気付かないまま、運用を続けてしまう。

MR-3 の留意事項に対する説明

●学習過程の妥当性 < PoC/開発 >

●再学習事の学習過程の妥当性 <運用>

- PoC, 開発, 運用時に限らず学習が十分収束する学習用データセットを用意できているか。またその学習用データセットを用いた交差検証を行っているか。
- 反復計算により正答率や損失関数を改善していく学習において、正答率及び損失関数の残差を見える化し、異常な変化が無く収束していることを確認しているか。

留意事項を怠ったことによる影響

- 実際に AI モデルを開発→運用したときと性能と、PoC 時の性能が異なってしまう。
- 学習したときに使用した訓練データに依存した AI モデルが作られる(過学習する)。

MR-4 の留意事項に対する説明

●AI モデルの構造の妥当性 < PoC/開発 >

●ハイパーパラメータの記録 < PoC/開発 >

●再学習の際のハイパーパラメータの記録 <運用>

< PoC/開発 >

- AI アルゴリズム選択およびハイパーパラメータの設定根拠が明確かつ社内デザインレビュー等の手段で承認されているか。AI アルゴリズム、ハイパーパラメータの根拠が顧客に理解さ

れるものであるか。テストデータの解析結果を提示し、AI アルゴリズム、ハイパーパラメータの設定根拠を顧客に提示出来ることが望ましい。ハイパーパラメータ記録は構成管理に関するため PA-3 も参照すること。

<運用>

- AI アルゴリズム選択およびハイパーパラメータの設定根拠が明確かつ社内デザインレビュー等の手段で承認されているか。AI アルゴリズム、ハイパーパラメータの根拠が顧客に理解されるものであるか。テストデータの解析結果を提示し、AI アルゴリズム、ハイパーパラメータの設定根拠を顧客に提示出来ることが望ましい。ハイパーパラメータ記録は構成管理に関するため PA-3 も参照すること。

留意事項を怠ったことによる影響

<PoC>

- 実際に AI モデルを開発→運用したときと性能と、PoC 時の性能が異なってしまう。

<開発>

- AI コンポーネントの推論する結果の信頼性が低下する。

<運用>

- AI モデルの劣化が疑われたとき、ハイパーパラメータの変更によるものなのか、学習時と環境が異なることによるものなか判別出来ない。

MR-5 の留意事項に対する説明

● AI モデルに影響を及ぼすノイズの洗い出し< PoC >

- ノイズに対する頑健性について顧客に説明しているか。ここでノイズとは以下の 2 種類が考えられる。
 1. センサやアクチュエータの時系列データや画像データに発生するノイズ
 2. 想定している環境外のデータが混入すること
- 上記のノイズ候補を事前に抽出しているか。

留意事項を怠ったことによる影響

- 実際に AI モデルを開発→運用したときと性能と、PoC 時の性能が異なってしまう。

● AI モデルのノイズ耐性（ロバスト性）の妥当性 <開発>

- 生データのノイズフィルタ除去の仕様を確認できているか。（例えばセンサデータであればローパス/ハイパスフィルタや FFT などの信号解析を前処理として行っているか。）
- 想定している環境外のデータは学習するときの学習用データセットから除外しているか。

留意事項を怠ったことによる影響

- 混入したノイズにより、AI コンポーネントの正答率が低下する。

MR-6 の留意事項に対する説明

●再学習における性能劣化の許容度 <運用>

- 劣化の許容範囲とその KPI をビジネス課題を考慮して設定できているか。
- 訓練データの特性変化や出力追加の範囲を想定できているか。
- 想定した訓練データの特性変化や出力追加で再学習／追加学習した結果が、劣化の許容範囲に収まっているか。

留意事項を怠ったことによる影響

- 劣化の検討不足による運用後の劣化対処コストが上がる。

● AI モデルの自動更新・配備の検査内容の十分性 <運用>

- AI モデルの再学習／追加学習における十分な検査・試験の自動化を導入しているか。
- 特性変化や性能変化を測るメトリクスが明確で、かつ許容範囲を超えた場合の対処が決まっているか。

留意事項を怠ったことによる影響

- 思わぬ AI モデルの劣化が知らずに起きる可能性がある。

MR-7 の留意事項に対する説明

●外部ライブラリのテスト <開発>

- AI システムを評価するときに、外部ライブラリの API 仕様に基づく単体テストやシステムテスト仕様を作成し、その評価を行うこと。テストに使用するデータは本番運用と同等なデータであれば良いが、そのようなデータが取得出来なければテスト用のダミーデータで代替しても良い。

留意事項を怠ったことによる影響

- 外部ライブラリの評価がなされないままに AI システムがリリースされてしまうことによる、

システム品質低下リスク発生。

●外部ライブラリの責任範囲 < PoC >

- 外部ライブラリのベンダー（業務委託先も含む）との間で、不具合が発生したときの改修及びその費用や工数の責任範囲について合意されていること。
- フリーソフトウェアであった場合、GPL や MIT といったライセンス形態を確認して AI システムへの組み込みが可能かどうか判断すること。

留意事項を怠ったことによる影響

- 外部ライブラリの不具合が改修されるまでに時間を要し、AI システムの開発が遅延する。
- フリーライセンスの補償範囲を逸脱した不具合が発生した場合に回避手段を検討する必要が生じる。

7.6.5 System Quality

Model Robustness, Data Integrity の保証が難しい一方で、産業用システムは社会的役割からシステムの安定動作、安全性、セキュリティへの要求は高い。アプリケーションや事業体によっては外部機関での認証や、品質保証プロセスを通じてシステム全体の品質保証を説明する必然性がある。そこでは、AI コンポーネントを追加することによるシステム全体への品質影響や、経年劣化などの環境依存性に対する説明を産業用システムのステークホルダーに説明する。

表 7.15 System Quality の留意事項

ID	観点	留意事項	PoC	開発	運用
SQ-1	顧客側期待の高さ 狙っているのが「人間並み」なのかどうか	● AI システムに対する顧客価値の考慮 ・ 提供する AI システムは、顧客のビジネス課題解決に適したものとなっているか。また、その効果・価値を計測できているか。	○	○	○
SQ-2	発生しうる品質事故の致命度は許容できる程度に低く抑えられているか ・品質事故の致命度はドメインによって異なる（身体や生命への危害、経済的ダメージ／社会や環境への影響／不快感、魅力の低さ、意味のなさ、反倫理） AI システムが第三者の知的財産権を侵害しないか	● AI システムに対するリスクの抽出 ・ AI システムの出力が確率的動作であることを考慮した上で、リスク分析（例：FMEA、シチュエーション分析、HAZOP 分析等）を実施し、AI システム利用時に発生する可能性のあるリスク（経済リスク、安全リスク、環境リスク）を抽出したか。	○		
		● AI システムに対するリスク予測の考慮 ・ AI システムの出力が確率的動作であることを考慮した上で、AI システムの出力による品質事故発生のリスク分析（例：FMEA、シチュエーション分析、HAZOP 分析等）及び低減対策が検討されているか。	○		
		● AI システムに対するリスク予測の見直し・追加 ・ 開発時に実施したリスク分析（シチュエーション分析、HAZOP 分析等）を運用段階で見直しているか、新たなリスクが出たら追加しているか。			○

表 7.16 System Quality の留意事項（続き）

ID	観点	留意事項	PoC	開発	運用
SQ-3	システムの事故到達度・安全機能・耐攻撃性は充分か ・防護機構のよさ・多さ ・回避性や制御性の高さ ・自己修復性 システムに対するAIの寄与度を抑えられているか	<ul style="list-style-type: none"> ● AI システムに対する安全性の確保 <ul style="list-style-type: none"> ・ 安全・セキュリティ機構を含めたアーキテクチャ設計が考慮されているか。 ・ 提供する AI システムの出力に対する安全性設計は考慮されているか。 ● 異常出力を防止する制御機構の確保 <ul style="list-style-type: none"> ・ AI の異常を自己判断できる仕組みや、出力データをモニタし、適切な出力範囲にコントロールする仕組みの実装しているか。 ● AI システムに対する保全性（故障や異常を検知・診断し修復する能力）の確保 <ul style="list-style-type: none"> ・ AI の信頼性低下を検知し、AI を使わないシステムへのシームレスに移行する仕組みを実装を行っているか。（システムを止めずに AI を止められる仕組み）。 ● 再学習にフィードバックする入力データへの安全性の確保 <ul style="list-style-type: none"> ・ 学習に使用する学習用データに対して、性能劣化に繋がる悪意のあるデータの混入を防ぐことができるか。もしくは、学習前に悪意のあるデータを排除する機構があるか。 ● 運用中の入力データの安全性の確保 <ul style="list-style-type: none"> ・ 運用中の入力データについて、異常な動作に繋がるような、または悪意のあるデータを検知し、排除する機構を実装しているか。 ● 再学習にフィードバックする入力データの監視 <ul style="list-style-type: none"> ・ 学習にフィードバックするデータに対して、性能劣化に繋がる悪意のあるデータの混入を防ぐことができるか。もしくは、学習前に悪意のあるデータを排除する機構があるか。に繋がるような異常または悪意のあるデータを排除できる仕組みがあるか。 ● 運用中の入力データの監視 <ul style="list-style-type: none"> ・ 運用中の推論を利用する入力データについて、異常な動作に繋がるような異常または悪意のあるデータを排除できる仕組みがあるか。 		○	

表 7.17 System Quality の留意事項（続き）

ID	観点	留意事項	PoC	開発	運用
SQ-4	品質事故を引き起こしうる事象の発生頻度は低いと見積もることができるか ・事象の発生頻度 ・事象の網羅性 ・事象の発生に対する制御可能性	● AI システムの安全動作の妥当性の評価 ・AI システムの動作実績をもとに統計的手法等を用いて、安全性を示すことができるか。		○	○
SQ-5	ステークホルダーに対する保証性、説明可能性、納得性は十分か	● AI システムの説明可能性 ・AI の出力結果に対する根拠を説明することができるか。もしくは、統計的手法等を用いて、結果の妥当性示すことが可能か。	○	○	○
		● AI システムの納得性 ・AI システムの出力結果に対する根拠を示すことで、顧客の納得感が得られたか。	○		
		● AI システムの理解容易性 ・AI システムの出力結果に対する根拠を示すことで、顧客の納得感が得られたか。			○
SQ-6	AI の導入や変更がシステム全体のふるまいや性能などの品質に悪影響を与えていないか	●運用中のシステム動作の妥当性 ・継続的な運用に伴うシステムの性能などの品質が低下する可能性を検討したか。 ・AI システム入力データの特性の変化や、性能の劣化をチェックする仕組みはあるか。 ・AI システム入力データの特性変化に対して、品質事故の発生防止等を目的に、システムの正常動作を維持する方法が想定できているか。 ・システムを全体として、および意味のあるサブシステム単位で評価を行ったか。		○	○
SQ-7	将来のデータ増加・処理量増加に対して、システムを拡張することができるか。	●運用時に収集する入出力データ量の想定 ・運用時、どの程度の入出力データがあり、どの程度蓄積するかを見積もり、システム要件に反映しているか。現在取得していないが、将来取得見込みがある入出力データがあれば、拡張性として考慮しているか。	○	○	
		●運用中の入出力データ量の監視と制御 ・運用時、蓄積している入出力データ量を監視し、基準に基づき削除しているか。また、定めた見積もりを超過しないかを監視しているか。			○

表 7.18 System Quality の留意事項（続き）

ID	観点	留意事項	PoC	開発	運用
SQ-8	システムの構成品目（ハードウェアや OSS）について計画しているか	<ul style="list-style-type: none"> ●ソフトウェアやハードウェアの各種制約の検討 <ul style="list-style-type: none"> ※ AI をターゲットに実装した場合、顧客の要求する性能を出せるか検討する。 ・実装先に対するハードウェアの制限はあるか。 ・学習用データセットや学習済みモデルのサイズ削減の必要はあるか。その際に、性能の劣化はどこまで許されるのか検討したか。 ・再学習したモデルの配信方法を検討したか。 ・運用時にモデルを再学習する必要性を検討する。 ●運用時を想定したハードウェアの策定 <ul style="list-style-type: none"> ・運用時の負荷やデータ量に応じたハードウェアを選定しているか。メンテナンスや故障時の対応を計画しているか。 ●アップデートを考慮したソフトウェアの選定 <ul style="list-style-type: none"> ・OS や OSS 等の各種ソフトウェアの更新頻度やサポート期間を考慮してソフトウェアを利用しているか。ソフトウェアのアップデートに対する対応や、サポートが終わった際の対応について決めているか。 ●モデルの性能や構造に依存したハードウェア制約の検討 <ul style="list-style-type: none"> ・推論プログラムのネットワークの大きさと必要メモリのトレードオフを検討する。 ●計画に基づくハードウェアのメンテナンス <ul style="list-style-type: none"> ・運用中、負荷やデータ量は想定通りか ・開発時の計画に基づきハードウェアのメンテナンスを行っているか。 ●ソフトウェアのアップデート <ul style="list-style-type: none"> ・特にセキュリティアップデートがある場合など、OS や OSS 等のソフトウェアのアップデートに対してシステムの更新を行っているか。 ●ソフトウェアやハードウェアの各種制約の運用設計の合意 <ul style="list-style-type: none"> ・再学習の方法、再学習の配信方法についてシステムとしての運用設計を行い、客先と合意を得る。 	○		

SQ-1 の留意事項に対する説明

● AI システムに対する顧客価値の考慮 < PoC/開発/運用 >

< PoC >

- 顧客のビジネス課題を明文化し、顧客と認識をあわせたか。
- 顧客のビジネス課題解決における AI の必要性について顧客と認識をあわせたか。

< 開発 >

- 顧客のビジネス課題を解決できたことを判断するための品質目標を定めたか。
- 顧客のビジネス環境で AI を利用する場合の制約条件（入手可能な訓練データの数や質、学習用データセットの取り扱いに係る条件、AI システムの停止を防止するための冗長構成の要否など）を抽出したか。
- システムテストにおいて目標品質に対する評価を行い、顧客のビジネス課題解決に適したものとなっていることを確認したか。
- 値値の計測が難しい場合、計測できる代替メトリクスとの関連は妥当か。

< 運用 >

- 開発時に設定した品質目標を達成し、顧客のビジネス課題が解決ができていることを確認したか。
- 開発時に設定した品質目標や顧客のビジネス課題が AI システムで解決できていることを継続的に確認・フィードバックする仕組みがあるか。

留意事項を怠ったことによる影響

- 顧客価値を考慮せずに、設計時の品質目標を達成しても、顧客のビジネス課題解決できないことがある。
- 値値が時間の経過とともに低下していく可能性があるので、リリース時に達成していても時間とともにビジネス課題が解決できなくなることがある。

SQ-2 の留意事項に対する説明

● AI システムに対するリスクの抽出 < PoC >

- AI システムの出力が確率的動作であり、想定外の出力結果が得られる可能性があることを踏まえた上で、リスク分析（例：FMEA、シチュエーション分析、HAZOP 分析等）を実施し、AI システム利用時に発生する可能性のあるリスク（経済リスク、安全リスク、環境リスク）を抽出したか。

- AI システムを利用／運用中に発生する可能性のあるリスクを網羅したか。
- リスク分析の際に、AI システム特有の確率的動作、出力結果の説明可能性が低いことを考慮したか。
- 抽出したリスクの深刻度は適切か。

留意事項を怠ったことによる影響

- 抽出したリスクが適切でない（網羅性が低いあるいは深刻度が誤っている）と、適切な対策をアーキテクチャ、設計に反映できず、想定外のリスクに対して脆弱な AI システムになることがある。その結果として、AI システム停止、AI システムの動作異常を検知できることによる不良品の製造や見逃しの発生、あるいは機器の異常動作により人に危険が及ぶ可能性が生じる。

● AI システムに対するリスク予測の考慮＜開発＞

- AI システムの出力が確率的動作であり、想定外の出力結果が得られる可能性があることを踏まえた上で、リスク分析（例：FMEA、シチュエーション分析、HAZOP 分析等）を実施し、システム運用時に発生する可能性のあるリスク（経済リスク、安全リスク、環境リスク）を網羅し、それらの深刻度を分析したか。
- リスク分析に基づいてリスク低減手段（実行中監視、冗長性、出力保証のためのルール実装等）が設計されているか。
- リスク分析によって抽出したリスクに対して、AI コンポーネントの実動作に基づき、リスク低減手段によってリスクが低減したことを確認したか。

留意事項を怠ったことによる影響

- 抽出したリスクが適切でない（網羅性が低いあるいは深刻度が誤っている）と、適切な対策をアーキテクチャ、設計に反映できず、想定外のリスクに対して脆弱な AI システムになることがある。その結果として、AI システム停止、AI システムの動作異常を検知できることによる不良品の製造や見逃しの発生、あるいは機器の異常動作により人に危険が及ぶ可能性が生じる。

● AI システムに対するリスク予測の見直し・追加＜運用＞

- 開発時に実施したリスク分析（例：FMEA、シチュエーション分析、HAZOP 分析等）に加え、運用中に AI の確率的動作によって想定外の AI システムの出力データ、学習済みモデルの精度劣化があればそれらを新たなるリスクとして追加したか。
- 開発時に用いていた学習用データセットと運用時に得られた入力データとの間に差異がある場合には、学習済みモデルの精度低下のリスクが高くなる。差異があった場合には、AI シス

テムの運用担当者にフィードバックし、リスク低減の措置を行ったか。

留意事項を怠ったことによる影響

- リスク分析が不十分だとリスクが発生した場合に迅速な対応ができなかったり、複数のリスクが発生した場合には適切な優先順位付けができない。結果として事故や損害が生じることがある。
- 学習時に想定でなかった運用環境で発見したリスクを追加しないと、運用環境の特徴やシステムを取り巻く外部環境の状況の変化によって生じるリスクを把握することができず、故障や事故が発生することがある。その結果として、AI システム停止、AI システムの動作異常を検知できることによる不良品の製造や見逃しの発生、あるいは機器の異常動作により人に危険が及ぶ可能性が生じる。

SQ-3 の留意事項に対する説明

● AI システムに対する安全性の確保<開発>

- AI システムの出力が確率的動作であり、想定外の出力結果が得られる可能性があることを踏まえた上で、リスク分析において抽出した安全リスク事象が発生しても AI システムとしての品質を確保するために、システムのフェールセーフ機能設計や、ロールバック設計など、安全・セキュリティ機構を考慮した設計をおこなったか。
- 運用時の入力データに対して悪意のあるデータが混入する可能性を考慮し、学習時の学習用データと運用時の入力データの相関性をデータの平均値、分散、相関係数等の統計量を用いて比較分析し、同一の母集団からサンプリングしたものと言えるか検定を行ったか。
- 運用時の入力データが、学習時の学習用データとは別の母集団から得られたと判別出来たなら、別となった理由は何か(システムクラック、設置環境が変わった等)を調査したか。

留意事項を怠ったことによる影響

- 安全性確保の観点でフェールセーフやロールバック機能が適切に設計されていないと、リスク発生時における適切なシステムの出力を維持できなくなることがある。

● 異常出力を防止する制御機構の確保<開発>

- 出力データをモニタし、異常出力であっても適切な出力範囲に制御する機構を実装しているか。
- 異常出力の検知として、統計的手法(外れ値検知や変化点検知等)、機械学習を用いた手法(k 近傍法、単純ベイズ法等)を用いたか。

留意事項を怠ったことによる影響

- 異常出力を防止する制御機能がないと、リスク発生時に出力値が異常値を示したままになり、システムの安全性が低下することがある。
- 故障や異常を検知・診断し修復する能力が確保できていないと、リスク発生による AI の信頼性の低下を検知できないか、検知しても適切な対処が行われないことがある。その結果として、AI システム停止、AI システムの動作異常を検知できずに不良品の製造や見逃しの発生、あるいは機器の異常動作により人に危険が及ぶ可能性が生じる。

● AI システムに対する保全性（故障や異常を検知・診断し修復する能力）の確保＜開発＞

- AI コンポーネントの故障や異常を監視し、異常を検知した場合に AI を用いない代替系へシームレスに移行する仕組みを実装しているか。（システムを止めずに学習済み AI モデルを止められる仕組み）。

留意事項を怠ったことによる影響

- 故障や異常を検知・診断する能力が低いと、故障や事故を未然に防止することができないことがある。また、利用者に対しても、身体的、経済的なリスクを生じさせることがある。その結果として、AI システム停止、AI システムの動作異常を検知できないことによる不良品の製造や見逃しの発生、あるいは機器の異常動作により人に危険が及ぶ可能性が生じる。

● 再学習にフィードバックする入力データへの安全性の確保＜開発＞

- 運用中に再学習のためにフィードバックする入力データに対して、性能劣化に繋がる悪意のあるデータ（学習時の学習用データとは別の母集団から得られた運用時の入力データ、入力データの外れ値など）の混入を防ぐ機構、もしくは、再学習前の学習用データに対して悪意のあるデータを排除する機構を実装しているか。悪意のあるデータの入力を防ぐ機構は、例えば、データを秘匿化して入力データとして正しい形式を分からないようにし、秘匿化した後に混入された悪意のあるデータが容易に発見できるようにする手法がある。
- 学習前の学習用データに対して悪意あるデータを排除できなかった場合、AI システムを悪意あるデータが混入した学習用データで学習する前の状態に戻すことができるか。
- AI システム出力データの異常が発生した後で、混入してしまった悪意のある入力データを特定するために、例えば入力データが入力された時刻情報を保持する等の仕組みを設けたか。

留意事項を怠ったことによる影響

- 運用中に得られた入力データから、外れ値や欠損データを除外せずに再学習にフィードバックすると、学習済みモデルの精度低下を招く可能性がある。

● 運用中の入力データの安全性の確保＜開発＞

- 運用中の入力データについて、異常な動作に繋がるような、または悪意のあるデータを検知し、排除する機構を実装しているか。

留意事項を怠ったことによる影響

- 運用中に得られた入力データに異常な動作に繋がる外れ値データや欠損のあるデータが含まれていると、AI の確率的動作により、学習済みモデルから想定外の出力データが得られる可能性がある。学習済みモデルの精度低下だけでなく、AI システムに求められる機能性が担保できないので、AI システムの信頼性低下を招く可能性がある。

●再学習にフィードバックする入力データの安全性の監視<運用>

- AI システムの信頼性を向上させるため、運用中に再学習用にフィードバックする入力データに対して、性能劣化に繋がる悪意のあるデータの混入を監視できているか。もしくは再学習前に悪意のあるデータを排除できているか。
- 学習前に悪意あるデータを排除できなかった場合、悪意あるデータを学習する前の状態に戻すことができるか。

留意事項を怠ったことによる影響

- 再学習にフィードバックする入力データに悪意のあるデータの混入を検知できないと、再学習のモデルが実際の入力データを反映しないまま生成されてしまい、再学習モデルの評価の段階まで気づけないことがある。

●運用中の入力データの安全性の監視<運用>

- 運用中の推論を利用する入力データについて、異常な動作に繋がるような、または悪意のあるデータを検知し、排除できているか。
- 学習前に悪意あるデータを排除できなかった場合、悪意あるデータを学習する前に戻すことができるか

留意事項を怠ったことによる影響

- 運用中に推論を利用する入力データに悪意のあるデータの混入を検知できないと、誤った推論結果出力しても気づけないことがある。その結果として、再学習済みモデルの精度が低下する。さらに精度低下により、AI システム停止、AI システムの動作異常を検知できずに不良品の製造や見逃しの発生、あるいは機器の異常動作により人に危険が及ぶ可能性が生じる。

SQ-4 の留意事項に対する説明

● AI システムの安全動作の妥当性の評価<開発/運用>

- AI システムとしてのにおける異常動作（異常な出力、異常な動作につながるデータ入力、悪意のあるデータの入力）の発生に対して、安全動作機構で制御できたことを平均、分散、相関係数等の統計的手法で評価し、安全性を示すことができているか。

留意事項を怠ったことによる影響

- 運用者が異常動作に対する安全性を定量的に評価・提示できないので、発生頻度が高い、あるいは発生時の影響が大きいといった高リスクの異常動作に対して優先的に対策を取ることができないことがある。その結果として、AI システム停止、AI システムの動作異常を検知できないことによる不良品の製造や見逃しの発生、あるいは機器の異常動作により人に危険が及ぶ可能性がある。

SQ-5 の留意事項に対する説明

● AI システムの説明可能性< PoC/開発/運用 >

<PoC>

- AI コンポーネント開発時に用いるアルゴリズムとして、説明が容易な単純なモデルを使用したか。もしくは、複雑なモデルを単純なモデルで近似的に表現することを行ったか。
- AI システムとしての出力データに対する根拠として、外部環境や AI システムとしての入力データの特徴を説明する手段があるか。もしくは、統計的手法等を用いて、AI システムとしての出力データの妥当性示すことが可能か。

<開発>

- AI コンポーネント開発時に用いるアルゴリズムとして、説明が容易な単純なモデルを使用したか。もしくは、複雑なモデルを単純なモデルで近似的に表現することを行ったか。
- AI システムとしての出力データに対する根拠として、外部環境や AI システム入力データの特徴を説明する手段があるか。もしくは、統計的手法等を用いて、AI システムとしての出力データの妥当性示すことが可能か。

<運用>

- AI コンポーネント開発時に用いるアルゴリズムとして、説明が容易な単純なモデルを使用したか。もしくは、複雑なモデルを単純なモデルで近似的に表現することを行ったか。
- AI システムとしての出力データに対する根拠として、外部環境や AI システム入力データの特徴を説明する手段があるか。もしくは、統計的手法等を用いて、AI システム出力データの妥当性示すことが可能か。

留意事項を怠ったことによる影響

- AI システム出力データが不適切と判断された場合にその原因の解析が困難であるため、AI システムの開発者や運用者による迅速な修正が行えなくなることがある。

● AI システムの納得性< PoC >

- AI システムとしての出力データが顧客の想定する期待値と一致しているか。もしくは、期待値と一致しない場合であっても、AI システムとしての出力データが得られる根拠を示すことで、顧客の納得感が得られたか。

留意事項を怠ったことによる影響

- AI システム出力データに対して、顧客からの信頼が得られない可能性がある。

● AI システムの理解容易性< 運用 >

- AI システム出力データが顧客の想定する期待値と一致しているか。もしくは、期待値と一致しない場合であっても、AI システム出力データが得られる根拠を示すことで、顧客の納得感が得られたか。

留意事項を怠ったことによる影響

- AI システム出力データに対して、顧客からの信頼が得られない可能性がある。

SQ-6 の留意事項に対する説明

● 運用中のシステム動作の妥当性< 開発/運用 >

< 開発 >

- AI システムの出力データの精度など、AI システムが正常動作だと判断できる性能目標が定められているか。
- AI システムの出力データの精度をチェックする手段はあるか。
- AI システム入力データの特性の変化をチェックする手段はあるか。
- AI システムの出力データの精度劣化や AI システム入力データの特性変化をチェックする実施頻度を適切に決めたか。
- AI システム入力データの特性変化に対して、品質事故の発生防止等を目的に、システムの正常動作を維持する方法が想定できているか。

< 運用 >

- AI システム入力データの特性変化や AI システムの精度劣化のチェックを定められた頻度で実施しているか。

留意事項を怠ったことによる影響

- AI システムの性能目標が定められていないと、正常／異常の判断ができず、AI システムとしての出力データの精度低下を招く可能性がある。
- AI システム入力データの特性の変化や精度低下をチェックする手段がなかったり、チェックの適切な実施頻度が規定されていないと、AI システムの精度低下の発生に対して迅速に対応できないことがある。その結果として、AI システム停止、AI システムの動作異常を検知できずに不良品の製造や見逃しの発生の可能性が生じる。

SQ-7 の留意事項に対する説明

●運用時に収集する入出力データ量の想定< PoC/開発 >

- 運用中、保存する入出力データ量を適宜監視し、保存可能な入出力データ量を拡張する手段があるか。
- 再学習時の学習用データセットとして入力データや正解／不正解がラベル付けされた AI システム出力データを継続的に蓄積する場合において、入出力データ保存量を適正に保ち、かつ AI システムが適正に学習をしつづけられるように、入出力データの保存と廃棄の規約を事前に定めているか。
- 再学習用の入力データを蓄積するために、保存する入出力データ量の監視頻度は決められているか。

留意事項を怠ったことによる影響

- 運用中に収集した入出力データが保存できなくなったり、計画外のストレージの拡張作業が発生する可能性がある。

●運用中の入出力データ量の監視と制御< 運用 >

- システム要件で定めた最大保存入出力データ量の見積もり値を超過していないかの監視を定められた頻度で実施しているか。

留意事項を怠ったことによる影響

- 運用中に収集した入出力データが保存できなくなったり、計画外のストレージの拡張作業が発生する可能性がある。

SQ-8 の留意事項に対する説明

●ソフトウェアやハードウェアの各種制約の検討< PoC >

- ニューラル・ネットワークを使う場合、そのネットワークの大きさと必要メモリの関係を示し、実装先のハードウェアで動作可能かを確かめる。(※ネットワークを大きく(複雑に)すると、正解率はあがるが、必要メモリも増える。)
- 量子化の必要はあるのか、あるいは、重み情報のスパース化の要否を検討する。その際、計算精度の変更/メモリ/正解率劣化のトレードオフを示し、量子化／スパース化による性能の劣化がどこまで許されるのかを検討する。(※計算精度を下げると使用メモリは減るが、正解率が劣化する)
- 実装先のプログラミング言語は学習時の言語のままで良いのかを検討する。
- FPGA 実装の場合は、設計手法について開発期間と性能のトレードオフの観点から、手設計とするか、高位合成とするかを検討する。
- 運用中に学習済みモデルを更新する必要ある場合、客先と更新方法を合意する。更新方法の合意の際には、再学習は誰が行うか、再学習後のモデルの配信方法についても合意する。
例) 人力 (ROM 焼き、メモリカード手渡し)、自動 (ネットワーク配信) など

留意事項を怠ったことによる影響

- 正しい制約条件が特定できない場合には運用時に所定の性能が出せずに、ハードウェアの選定やシステム構成設計の手戻りが発生する可能性がある。

●運用時を想定したハードウェアの策定<開発>

- 運用中に AI モデルを実行させる際の負荷やデータ量等のシステム要件に応じたハードウェアを選定しているか。
- 将来、入力データが増加した際に拡張が可能なハードウェアを選定しているか。

留意事項を怠ったことによる影響

- 運用時の負荷やデータ量を計画していないと、適切なハードウェアを選定できないことがある。そのため、オーバースペックなハードやアーキテクチャを選定する可能性がある。

●アップデートを考慮したソフトウェアの選定<開発>

- OS や OSS 等の各種ソフトウェアの更新頻度やサポート期間を考慮して利用しているか。AI 関係の OSS は更新サイクルが短いので、特にシステムの開発サイクルとの違いを考慮することが重要。PA-8 も参照。
- ソフトウェアのアップデートに対する対応や、サポートが終わった際の対応を決めているか。

留意事項を怠ったことによる影響

- AI システムで使用している OS や OSS の更新サイクルが短い場合、あるいは利用している外

部ソフトウェアのサポートが行われなくなった後に問題が発見された場合に、開発中の AI システム側での対応工数が増加する。

●モデルの性能や構造に依存したハードウェア制約の検討<開発>

- ニューラル・ネットワークの大きさと必要メモリの関係を示す。
 - ※ネットワークを大きく（複雑に）すると、正解率はあがるが、必要メモリも増える。
- 量子化（計算精度の変更）/メモリ/正解率劣化のトレードオフを示す。
- ※計算精度を下げると使用メモリは減るが、正解率の性能が劣化する事を示す。

留意事項を怠ったことによる影響

- 正しい制約条件が特定できない場合には運用時に所定の性能が出せずに、ハードウェアの選定やシステム構成設計の手戻りが発生する可能性がある。

●計画に基づくハードウェアのメンテナンス<運用>

- システム要件で定めた最大保存データ量や負荷を遵守することの監視を定められた頻度で実施しているか。

留意事項を怠ったことによる影響

- 運用中にデータが保存できなくなる可能性がある。
- ハードウェアのメンテナンスが行われないと、AI システムのターンアラウンドタイムの劣化、あるいは計画外の AI システム停止やストレージの拡張作業が発生する可能性がある。

●ソフトウェアのアップデート<運用>

- 特にセキュリティアップデートがある場合など、OS や OSS 等の各種ソフトウェアのアップデートに対してシステムの更新を行っているか。AI 関係の OSS は更新サイクルが短いので、特にシステムの開発サイクルとの違いを考慮することが重要。PA-8 も参照。

留意事項を怠ったことによる影響

- ソフトウェアのアップデートを行っていない場合、あるいはサポートが終了した外部ソフトウェアを利用してアップデートができない場合には、問題が発見されても外部ソフトウェアの開発元による対処されない可能性が高い。そのため、運用中の AI システム側での対応が必要になる可能性がある。さらに、対応ができない場合は、AI システムの提供を終了せざるを得ない可能性がある。

●ソフトウェアやハードウェアの各種制約の運用設計の合意<運用>

- 再学習の方法、再学習の配信方法についてシステムとしての運用設計を行い、客先と合意を得る。

留意事項を怠ったことによる影響

- 再学習の方法、再学習の配信方法についての運用設計が行われていないと、運用中に精度低下が発生しても迅速な対応ができない。精度低下に伴って不正解が増加すると、人によるチェックや修正等が必要となり、AI システムを利用することによる効果が落ちる。

7.6.6 Process Agility

産業用システムでは明確な達成目標と実現可能性に基づいた開発が一般であった。一方で機械学習技術は達成目標と実現可能性の両方に不確実性があるため、AI コンポーネントや AI コンポーネント開発では反復的な開発と品質保証が必要となる。

表 7.19 Process Agility の留意事項

ID	観点	留意事項	PoC	開発	運用
PA-1	AI コンポーネントの開発においては、充分短い反復単位で反復型開発は行っているか、AI モデル・AI システムの品質向上の周期は充分短いか	●アジャイル型ソフトウェア開発の実行能力 ・アジャイルのような反復型ソフトウェア開発を行うにあたって必要となる顧客の協力、人材や設備等の社内環境や開発手順等に不備はないか。	○	○	

表 7.20 Process Agility の留意事項（続き）

ID	観点	留意事項	PoC	開発	運用
PA-2	運用状況の継続的なフィードバックは頻繁か	● AI システム PoC 時におけるシステムの利用者の協力度 ・ PoC での試行中、あるいは試行結果から、プロダクトオーナーが適切に判断できるよう、システムの利用者から充分な協力を得られ、継続的なフィードバックを得ることは可能か。	○		
		● AI システム開発時におけるシステムの利用者からのフィードバックの適切な反映 ・ システムの利用者から充分に協力を得られた PoC 試行結果が、開発時に適切に管理され、フィードバックできるようになっているか。		○	
		● AI システム運用時におけるシステムの利用者からのフィードバックの適切な反映 ・ システムの利用者からのフィードバックが行われているかを確認できるようになっているか。			○
PA-3	リリースとロールバックは簡便で迅速に行えるか 問題発生時に、原因解析のため状況を記録し取得できる仕組みがあるか また状況をもとに事象を再現できるか	● AI システムの構成管理の妥当性 ・ AI のプログラムや学習用データセット等のバージョンについて、適切に構成管理が行われているか。	○	○	○
		● 問題発生時の情報の記録・取得や再現調査など、対応手段の明確化と準備 ・ 問題発生時の調査のために記録する情報・取得する手順/ツールは明確か。事象を再現する環境等の準備は十分か。	○	○	○
		● リリース計画の妥当性 ・ AI のプログラムのリリース計画（カナリアリリースなど）は、AI プロダクトの特性や顧客の要求に応じて適切に決められているか。	○	○	
		● ロールバックの迅速性 ・ リリースした AI のプログラムに異常が発生した場合、迅速にロールバックを行う仕組みがあるか。	○	○	○
		● リリース計画通りのリリース ・ AI のプログラムは開発時に決められたリリース計画に従ってリリースされているか。また、必要に応じてリリース計画は見直されているか。			○

表 7.21 Process Agility の留意事項（続き）

ID	観点	留意事項	PoC	開発	運用
PA-4	よりよくなっている見込みはあるか ・新しい特徴量を迅速に追加できるか ・モデルを迅速に改善できるか ・学習や推論のデバッグを行う手段を有しているか	●段階的リリースによる性能向上の見込み ・学習に使用するデータの見直し、特微量の追加、モデルの改善等により、性能向上が期待できるか。	○	○	
PA-5	開発チームは適切な能力を持った人財を備えているか ・機械学習やデータサイエンスの「専門家」やドメインの専門家は含まれているか	● AI システム開発の実行能力 ・AI システムの開発に必要な人材が集められ、開発体制上の役割が明確になっているか。	○	○	
		●顧客と連携した開発チーム ・顧客側にも、業務プロセスやドメイン知識を理解している人を有しており、開発体制に組み込んでいるか。	○	○	
PA-6	経験を技術に反映させられているか	●前開発の経験の反映 ・既存の AI 適用先の経験を、次の開発に技術として反映できるプロセス、体制が構築されているか。	○	○	○
PA-7	開発チーム外のステークホルダーは充分納得しているか	●ステークホルダーの体制と役割を明確に認識しているか ・開発チーム外のステークホルダーとして、管理層や異なる部門、顧客などがある。体制と役割を明確に認識し、コミュニケーションはしているか。	○	○	
		●管理層や他部門と合意できているか・協力関係ができているか ・開発チームの管理層や関係する他部門と開発対象や協力内容について合意しているか、協力を仰げそうか。	○	○	
		●現場は十分納得しているか ・AI の運用に対して、サービス利用者は十分納得しているか、フィードバックを得られているか。			○

表 7.22 Process Agility の留意事項（続き）

ID	観点	留意事項	PoC	開発	運用
PA-8	システムライフサイクルを見据えた更新計画があるか	● AI システムの更新計画の妥当性 ・ AI システムのライフサイクル全体を見据えた更新計画が策定されているか、またその更新計画は妥当か。更新計画には、OSS の更新、入力データ特性の変化への対応、入力/出力データ量増加に対する対応などを含む。		○	
		● 計画に従って更新できているか ・ 運用中も継続して顧客と意見交換しながら進められているか。顧客からフィードバックを都度得られているか。			○

PA-1 の留意事項に対する説明

● アジャイル型ソフトウェア開発の実行能力 < PoC/開発 >

- 顧客の協力：プロダクトオーナーがプロジェクトに専任できるか、そのことがプロダクトオーナーの所属する組織から理解されているか、即断即決できるように権限を委譲されているか。
- 人材：アジャイル開発の経験はあるか。マインドや振る舞いに課題はないか。

事前に担保できる品質（特に精度等の性能指標）を予測することは難しく、実験を繰り返しながらどこまでの品質を実現できるのかを試験的・探索的なプロセスを経て探ることになる。場合によっては、要求やユースケースを調節して「適合率が低くても再現率が高ければ受け入れる」といった判断が必要となる。

従って、試験的・探索的なプロセスについて、顧客をはじめとしたステークホルダーに事前に理解してもらい、協力を取り付けておくことが重要である。その際、顧客の協力体制として、以下についても留意しておくと良い。

1. プロダクトオーナーがプロジェクトに専任できるか
2. 1について、プロダクトオーナーが所属する組織から理解されているか
3. プロダクトオーナーが即断即決できるように権限を委譲されているか

留意事項を怠ったことによる影響

最終的に出来上がるものが、顧客のやりたいことと違うものになってしまい、手戻りが大きくなる。

PA-2 の留意事項に対する説明

● AI システム PoC 時におけるシステムの利用者の協力度< PoC >

システムの出力結果、ログ情報などのデータ、システムの利用者からの評価結果などについて、事前に何を計測するかをプロダクトオーナーと合意しておき、適切な判断ができるように準備しておくことが必要である。その上で、プロダクトオーナーは PoC 中に定期的に、および PoC 終了後に、必要な判断ができるようになっていることが望ましい。

判断を容易にするため、事前に指標値を合意しておくと良い。定期的に、というのはたとえば、有意なデータが 1 週間で蓄積できるなら、週に 1 度判断できるような場を設定すると良い。

留意事項を怠ったことによる影響

最終的に出来上がるものが、システムの利用者のやりたいことと違うものになってしまい、手戻りが大きくなる。

● AI システム開発時におけるシステムの利用者からのフィードバックの適切な反映<開発>

開発開始前に PoC にて収集したデータに基づいて、AI システムの開発における優先度をシステム開発者が判断し、プロダクトオーナーと合意しておく。また、開発した AI システムの出力が期待された結果かどうかを判断するため、事前に指標値についても合意しておくと良い。

開発チームメンバーは事前に合意された優先度に従って開発を進める。

システム開発者は事前に合意された指標値の達成度合いを確認し、開発の妥当性を判断することが望ましい。

留意事項を怠ったことによる影響

最終的に出来上がるものが、システムの利用者のやりたいことと違うものになってしまい、手戻りが大きくなる。

● AI システム開発時におけるシステムの利用者からのフィードバックの適切な反映<運用>

システムの出力結果、ログ情報などのデータ、システムの利用者からの評価結果などについて、事前に何を計測するかをプロダクトオーナーと合意しておき、適切な判断ができるように準備しておくことが必要である。その上で、プロダクトオーナーは運用中に定期的に必要な判断ができるようになっていることが望ましい。

判断を容易にするため、事前に指標値を合意しておくと良い。定期的に、というのはたとえば、有意なデータが 1 週間で蓄積できるなら、週に 1 度判断できるような場を設定すると良い。

留意事項を怠ったことによる影響

運用中に発生するインシデントへの適切なフィードバックができる仕組みがないと、継続的な改善が図れない。

PA-3 の留意事項に対する説明

● AI システムの構成管理の妥当性 < PoC/開発/運用 >

トラブルが起きた際、誰でも問題を切り分けできるようにするため、同一の AI モデルが開発できるよう前処理・学習に使用するハイパーパラメータ値（乱数のシード値など）、学習用データセットが記録され、履歴管理されているか。

留意事項を怠ったことによる影響

同じ条件下で固定の結果を得られず（再現性がない）、AI モデルの精度比較の結果が変動する恐れがある。トライ＆エラーに時間がかかることにより、開発時間が延び、リリースの遅延につながる。

● 問題発生時の情報の記録・取得や再現調査など、対応手段の明確化と準備 < PoC/開発/運用 >

データの傾向の変化に対しモデルが更新されず精度が低下したり、データのノイズやモデルのチューニング誤りにより予測判断の異常が発生する。これらの問題が発生した場合に、原因解析のために、発生状況を記録し漏れなく取得できる仕組みを備えておく必要がある。その為に、記録する情報・取得する為の手順/ツールは明確にしておくとよい。また、解析した原因の妥当性の判断や事象の修正確認のために、事象を再現させるための環境などを、事前に準備しておくとよい。※記録する情報の例： モデルのバージョン・パラメタ・前処理/後処理の内容・データセットのバージョン

留意事項を怠ったことによる影響

問題の修正誤りや精度/性能の低下だけでなく、問題の調査時間の増大やリリースのスケジュールの遅れにつながる。

● リリース計画の妥当性 < PoC/開発 >

トラブルが起きた際、誰でも問題を切り分けできるようにするため、同一の学習モデルが開発できるよう前処理・学習に使用するハイパーパラメータ値（乱数のシード値など）、学習用データセットが記録され、履歴管理されているか。

留意事項を怠ったことによる影響

（特にカナリアリリースについて）新たな AI モデルを本番環境で動かしたときに、以前のものより推論精度が低くなった場合や誤った振る舞いを引き起こした場合、ロールバックが完了するまでユーザー全体がサービスを利用できなくなる恐れがある。

● ロールバックの迅速性 < PoC/開発/運用 >

追加した新しいカテゴリの学習用データセットにより、顧客要件の幅広い意味での性能が劣化した場合、容易に性能劣化を検知できるか。また、検知結果から自動的に前版にロールバックできるまたは環境変数などの何らかのスイッチにて簡便にロールバックできる仕組みを用意しているか。

※ AI システムや AI コンポーネント等の構成管理の範囲は、特性によって検討が必要である。

留意事項を怠ったことによる影響

ロールバックに時間がかかるまたはロールバックに失敗してダウンタイムが伸びる恐れがある。AI システムの特性により、原因調査から解決まで時間がかかることが想定される。

●リリース計画通りのリリース<運用>

AI のプログラムは開発時に決められたリリース計画に従ってリリースされているか。また、必要に応じてリリース計画は見直されているか。

留意事項を怠ったことによる影響

AI のプログラムが計画どおりにリリースされない場合は、AI システムの開発に手戻りや遅延が発生するなど全体に影響する可能性がある。

PA-4 の留意事項に対する説明

●リリース計画の妥当性< PoC/開発 >

事前に性能・性能向上を見込むのは難しいが、リリースを重ねながら実際に性能向上が期待できるようにするためにには、データの見直しや特長量の追加、モデルの改善がやりやすいようなプロセス、体制を整えておくことが必要である。また、際限なく性能改善を追求することを避けるためにも、コストに見合った性能目標を事前に合意しておくと良い。

(例) 期待どおり性能向上ができたことを示すためには、例えば、PoC フェーズでは、計画立案の中で、

- 検証完了条件として性能（・時期・コスト）を決めておく必要がある。
- 検証完了条件がないと、性能を追い求めるばかりで検証終了の目途がたたず、コストだけが積み上がる可能性がある。（PoC 貧乏に陥る）

留意事項を怠ったことによる影響

データの見直しや特長量の追加、モデルの改善がやりにくいで、性能向上のための作業に時間、コストがかかる。結果として、事前に合意したコストに見合った性能目標を達成できなくなる可能性がある。

PA-5 の留意事項に対する説明

● AI プロダクト開発の実行能力< PoC/開発 >

以下のようないくつかのスキルが必要である。

- 評価指標を疑う（評価結果が過大に出てしまう）※要件定義書で設定した目的を達成するために現行の評価指標が適切か確認する。
- 学習用データセットを疑う（現実の問題に対し性能が出ないなど）。※交差検定で確認する。

- 問題設定を疑う（問題設定に不備があり、目標が達成できない）。※ AI に解かせる問題が適切か、より問題設定の詳細化ができる。

留意事項を怠ったことによる影響

本来不要な特徴量がシステムに加わり、エンジニアリングコスト、計算資源コストが増大する。

システムに加えてはならない特徴量が混入したり、廃止された特徴量がシステムに残ったり、データソースの消失することにより、意図しない動作など運用時障害に繋がる恐れがある。

●顧客と連携した開発チーム< PoC/開発 >

機械学習モデルを開発するチーム（顧客の業務プロセス等も理解していること）と学習用データセットに関するデータサイエンティストチームに分かれることが多い。

チームに分かれた体制のなかでも、性能劣化などの問題に対して誰が責任を持つか明確にしておくと良い。また、開発チームに業務プロセスやドメインに対する知識が不足している場合は、顧客側の有識者をチームに入れるなどにより、開発チームと顧客の連携を高めることが望まれる。

留意事項を怠ったことによる影響

本来不要な特徴量がシステムに加わり、エンジニアリングコスト、計算資源コストが増大する。システムに加えてはならない特徴量が混入したり、廃止された特徴量がシステムに残ったり、データソースの消失することにより、意図しない動作など運用時障害に繋がる恐れがある。

PA-6 の留意事項に対する説明

●前開発の経験の反映/開発の知見の反映< PoC/開発/運用 >

- 実験目的・実験条件/環境・実験結果・考察が明文化されていること。
- 誰でも同じ学習環境及び精度が再現できる仕組みがあること（Dockerfile などで学習環境やハイパーパラメータ（学習回数、学習率、バッチサイズなど）が記載されている）。

留意事項を怠ったことによる影響

AI システムの開発では、ハイパーパラメータ設計などの試行錯誤に従来の開発に比べて時間を要する。その際に、以前の開発経験を活用することで、試行錯誤の時間短縮などを狙える可能性がある。（=これを実施しないと効率的に開発が行えない）

PA-7 の留意事項に対する説明

●ステークホルダーの体制と役割を明確に認識しているか< PoC/開発 > 体制・協力関係: 「いつもと様子が違う、感覚に合わない結果が多くなった」など顧客/システムの利用者からのご意見を素早く吸上げて改善できるようにご意見や問い合わせに対するエスカレーション運用のルールを定められているか。

留意事項を怠ったことによる影響

連携がとれず、意図しない成果物やインターフェースができ、結果的に後戻りが発生する。

●管理層や他部門と合意できているか・協力関係ができているか<PoC/開発>

体制・協力関係: 「いつもと様子が違う、感覚に合わない結果が多くなった」など顧客/システムの利用者からのご意見を素早く吸上げて改善できるようにご意見や問い合わせに対するエスカレーション運用のルールを定められているか。

留意事項を怠ったことによる影響

連携がとれず、意図しない成果物やインターフェースができ、結果的に後戻りが発生する。

●ステークホルダーの体制と役割を明確に認識しているか<運用>

体制・協力関係: 「いつもと様子が違う、感覚に合わない結果が多くなった」など顧客/システムの利用者からのご意見を素早く吸上げて改善できるようにご意見や問い合わせに対するエスカレーション運用のルールを定められているか。

留意事項を怠ったことによる影響

連携がとれず、意図しない成果物やインターフェースができ、結果的に後戻りが発生する。

PA-8 の留意事項に対する説明

● AI システムの更新計画の妥当性<開発>

AI モデル/AI コンポーネントをどのタイミングで更新するかが計画されているか。(障害発生時、性能劣化時などのトリガーで更新 or 定期的に更新するなど。AI モデルだけ or 他の部分も含めて更新)

開発スピードに合わせた更新計画が必要である。

留意事項を怠ったことによる影響

産業用プロセスデータで扱うシステムでは、Web サービスのように AI モデル/AI コンポーネントや他コンポーネントを簡単に更新できない可能性が高い。そのため、AI モデル/AI コンポーネントを更新する計画をシステムの他コンポーネントの更新と合わせて更新する計画を立てる必要がある。更新しないまま運用を続けると、性能劣化などが起きた状態のままで使うことになるので、顧客の期待に添わない可能性がある。

●計画に従って更新できているか<運用>

AI モデル/AI コンポーネントをどのタイミングで更新するかが計画されているか。(障害発生時、性能劣化時などのトリガーで更新 or 定期的に更新するなど。AI モデルだけ or 他の部分も含めて更新)

開発スピードに合わせた更新計画が必要である。

留意事項を怠ったことによる影響

産業用プロセスデータで扱うシステムでは、Web サービスのように AI モデル/AI コンポーネントや他コンポーネントを簡単に更新できない可能性が高い。そのため、AI モデル/AI コンポーネントを更新する計画をシステムの他コンポーネントの更新と合わせて更新する計画を立てる必要がある。更新しないまま運用を続けると、性能劣化などが起きた状態のままで使うことになるので、顧客の期待に添わない可能性がある。

●現場は十分納得しているか＜運用＞

システムをよくするためには、実際に AI システムを使うサービス利用者の意見を伺い、改良していく必要がある。利用者の意見をフィードバックする仕組みを構築することが必要である。

留意事項を怠ったことによる影響

本当に業務効率改善などに役立っているかが分からず、サービス利用者目線たった改良ができない可能性がある。

7.7 AI プロダクト開発プロセスでの品質保証観点

本章では、IXI モデル(図 7.8)で想定する開発プロセスと、品質保証観点との対応を示す。

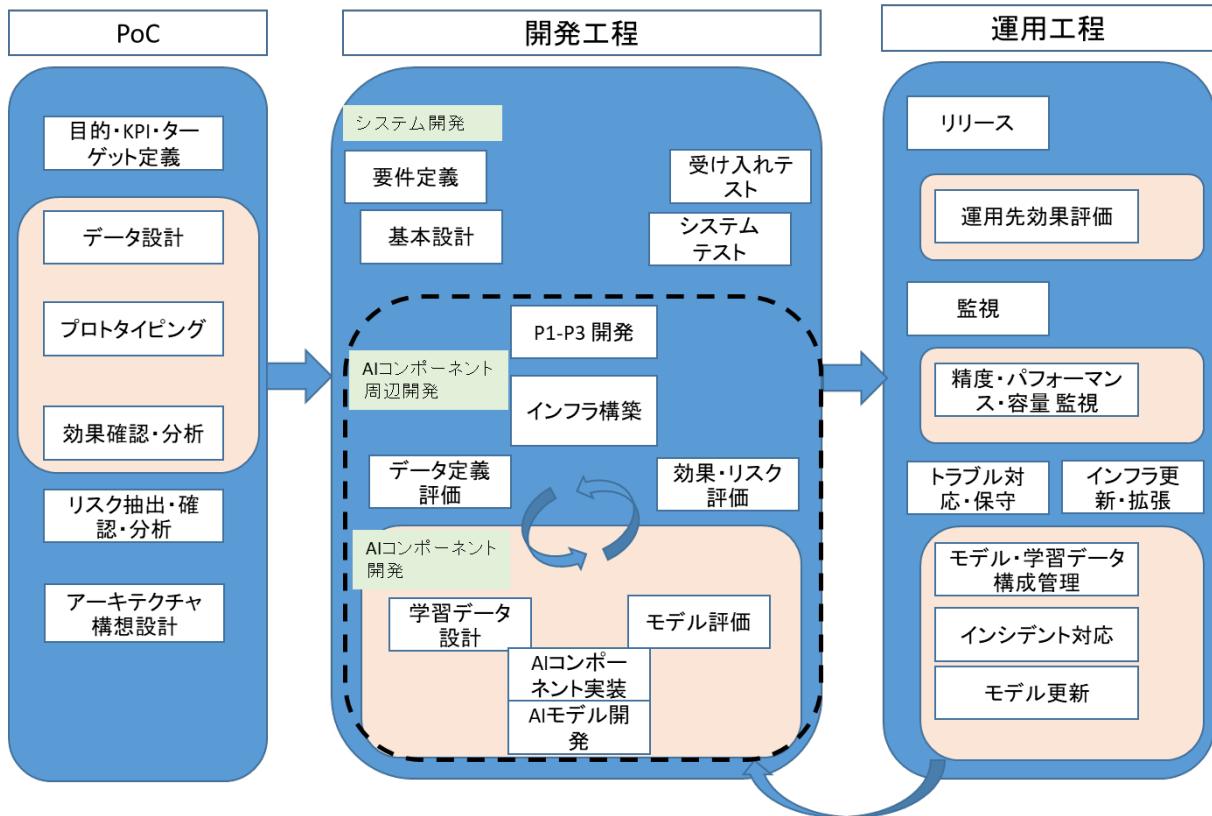


図 7.8 再掲：工程の全体像：IXI: Intelligent eXperimental Integration モデル

7.7.1 PoC

PoC では新たな概念やアイデアの実現可能性を示す活動である。目的・KPI ターゲットに対して、どのような AI コンポーネントを含めたアーキテクチャで対応可能であるか実証する。

IXI モデルでの PoC 工程の主要な活動を、表 7.23 に示す。

この工程での観点や 5 つの軸への対応を表 7.24 に示す。表の中の主な保証観点は、特に留意すべき観点であり、関連する観点は必要に応じて参考にすべき観点である。また SQ-* のようにアスタリスクで書いてあるものは、軸のすべての観点が対象になることを示している。

表 7.23 PoC 工程での主要な活動

活動	概要
1. 目的・KPI ターゲット定義	産業用システムに限らず、PoC での目標を設定し、関係者で合意する工程。この活動では、PoC での検証対象の開発・検証の完了基準を設定する。 産業用システムでは既存システムの稼働実績に基づく性能面の KPI ターゲットだけではなく、その実現性に伴うプロジェクトコスト、システムが許容可能なリスクといったビジネス要件に基づき設定する。
2. データ設計	1. により設定した課題に対して、AI コンポーネント開発で利用する入力データや学習用データセットの対象特定や収集のための活動である。産業用システムでは、長時間のデータが必要となる場合や（例：劣化推定）、精度保証が不十分な状態のデータを利用する場合がある。この場合、PoC 期間中で確かめる範囲とその確からしさを計画し、データ収集・評価の仕組みやレビューフィードバック体制を用意する。
3. プロトタイピング	2. で設計した内容に基づき、AI コンポーネント・モデル・P1~P3 の開発を実施する。産業用システムでは不確実性のあるモジュールを未検証で導入するケースは極めてまれであり、既存システム部（P1~P3）においては、データ収集にとどまりシステム品質低下リスクを抑制する場合がある。
4. 効果確認・分析	1~3. に基づいて AI コンポーネントの評価を行う。この評価では産業用システムでは 5M+E の変動を受けることを考慮した測定条件・評価条件をもとに実施する。分析結果次第で、2~3. の活動を繰り返す。
5. リスク抽出・確認・分析	対象システムの要件・構成や、認証・規格（例えば、機能安全規格・産業制御システムセキュリティ等）に基づきリスク対象の抽出・確認・分析を実施する。4. の結果をもとに、AI コンポーネントの不確実なふるまいをあてはめ、リスク事象に対して AI コンポーネントによる影響を受容可能であるか確認・分析し、アーキテクチャでの対策を実施する。
6. アーキテクチャ構想設計	開発時の全体アーキテクチャ構想を、データ設計・プロトタイピング・効果確認・分析といった活動によって明らかとなった各種要件に基づいて設計する。この活動によりデータ・モデル・構成を識別し、ステークホルダーの権利・契約の整合に利用する。

7.7.2 開発工程

開発工程では、PoC によって実証された範囲に基づき、実システムとしての設計・実装を行う。環境依存性や AI コンポーネントの運用形態によっては開発工程と運用工程の反復サイクルが非常に短い場合もあり得る。

開発工程は AI コンポーネントの不確実性に対応するため、「システム開発」「AI コンポーネント周辺開発」「AI コンポーネント開発」の 3 種類に分けて対応する（表 7.25）。これらの活動は反復性を持ち、3 つの特徴に対応した品質保証活動を実施する。

システム開発では、アーキテクチャ構想に基づいて、P1~P3・AI コンポーネントの設計・評価を実施する。従来型の開発では、詳細設計・実装・単体結合試験に該当する活動は「AI コンポーネント周辺開発」と「AI コンポーネント開発」に相当する。表 7.26 で主要活動を示し、表 7.27 で保証

表 7.24 PoC での主な保証観点

主要活動	保証での留意事項	主な保証観点	関連する保証観点
目的・KPI・ターゲット定義	目的（ビジネス課題）の具体性	CE-1	SQ-1, MR-1, CE-8
	期待効果、実証内容の明確さ	CE-6	-
データ設計	目的とデータとの対応付け	CE-4	DI-2
	データの質・量確保	DI-1,2	-
プロトタイピング	システムの実現可能性	MR-1	DI-1,2, MR-2,3,4,5,6,7, SQ-2,3, CE-3, PA-1,4
効果確認・分析	期待効果の確からしさ	CE-1	SQ-1
	環境依存性への評価	MR-1	DI-2, MR-2,5
リスク抽出・確認・分析	確率動作に対する受容性・リスク評価	CE-2	CE-3,4, SQ-2,5
	その他システム品質に対するリスク評価	SQ-2	SQ-3,4
アーキテクチャ構想設計	要件に基づく構成の識別	SQ-8	-
	権利・発明性などの整合	CE-5,7	-

表 7.25 開発工程内の 3 種の開発

項目	概要
システム開発	要件定義によりデータ収集を含む達成目標を明確にし、基本設計・システムテスト・受入テストを環境依存性に対応し反復型で開発・評価する。
AI コンポーネント周辺開発	データの定義・収集・管理をし、AI コンポーネントの精度やリスク評価を実施する。全条件網羅のデータ収集は不可能であるため、対象の事象に基づく条件の絞り込みや、データの統計的な評価による収集量の十分性といったデータ定義・評価活動を繰り返し、保証範囲の確認をする。
AI コンポーネント開発	得られたデータを利用し、適用アルゴリズム・学習モデル・ハイパーパラメータといった AI コンポーネントに必要な設定を開発・評価する。評価結果によっては収集すべきデータ定義（収集周期、特徴量等）を変更するといった、AI コンポーネント関連開発と関係する。

観点を示す。

AI コンポーネント周辺開発では、AI コンポーネントに必要な P1-P3 やデータ基盤の開発と評価を実施する表 7.28 で主要活動を示し、表 7.29 で保証観点を示す。

AI コンポーネント開発では、AI コンポーネントの動作を実現するための、アルゴリズム実装や、モデル開発といった活動を実施する。表 7.30 で主要活動を示し、表 7.31 で保証観点を示す。

表 7.26 システム開発での主要活動

項目	概要
要件定義	AI コンポーネントの目的・対象に基づいた、動作精度の目標や、P1～P3 で保証する範囲を決定する。また、System Quality に基づく要件も定義する。(例えば、セキュリティ要件を満たすためのデータ管理)
基本設計	要件定義に従い、各種コンポーネントを設計する。基本設計により対象とするデータが確定することもあり、システムで求められる環境依存性や説明容易性によっては多数のプロトタイピングをここで実施する。
システムテスト	実装された AI コンポーネント周辺システム、AI コンポーネントを動作ないし機上評価し、基本設計で意図した性能・精度の評価や、データ収集などの運用要件の評価を実施する。精度評価や収集データにより環境依存性への対応が不足すると判断した場合、要件定義や基本設計と反復する。
受け入れテスト	現場環境への導入判断をするための再評価、運用管理方法を確認する。例えば、対象外の環境条件でのシステムの振る舞いと、その対応・データ管理・確認手順の確認といった活動を実施する。運用環境中でモデル更新を前提とする場合、更新の前提条件、評価条件といった手順をこの活動で評価する。

表 7.27 システム開発での保証観点

主要活動	保証での留意事項	主な保証観点	関連する保証観点
要件定義	使用環境、ユースケースの定義	SQ-2	SQ-3,4,6,7,8
	システム構成条件の定義	SQ-8	SQ-6,7
	リスク管理	SQ-2	CE-2,5,7,8, SQ-4
	品質目標の設定	CE-1	SQ-1,2,3,4, MR-1
基本設計	システムのフェールセーフ設計	SQ-3	DI-4, SQ-4
	運用中評価の仕組み設計（カナリアリース）	PA-1	PA-2,3
	運用中インシデント対応でのロールバック設計	SQ-3	PA-3
	運用監視設計	SQ-6	DI-4, SQ-1,7, PA-2,3
	要件に基づくコンポーネントの設計	SQ-8	SQ-7
システムテスト	システム要求品質に対する評価	CE-1	SQ-*, PA-3
	説明性、環境依存性に対するロギング・データ十分性評価	SQ-5	DI-1,2,4,5, SQ-4
	挙動検証が必要なシステムのフェールセーフテスト	SQ-3	SQ-2,4
	運用監視の十分性評価	SQ-6	SQ-7
受入テスト	運用先環境での運用要件（管理手順等）適合性評価	PA-2	MR-2, SQ-*, PA-3,8
	運用先環境での環境依存性評価	DI-2	MR-5, SQ-6,7,8

表 7.28 AI コンポーネント周辺開発での主要活動

項目	概要
P1-P3 開発	基本設計に基づき、P1~P3 の開発を実施する。インフラ構築と連携し、AI コンポーネント開発や評価に必要なデータを収集する。
インフラ構築	AI コンポーネントの実装評価に利用するデータ収集・管理や、データを継続して管理するためのインフラ環境を構築する。
データ定義・評価	AI モデルに利用する訓練データや、効果・リスク評価に利用する評価データを定義する。そして、環境依存性にどの程度対応したデータ項目・データセット・特徴量であるか実データをもとに評価する。
効果・リスク評価	AI コンポーネントの実動作に基づき、要件化された効果・リスク（例えば予測精度、実行性能）を収集したデータをもとに評価する。

表 7.29 AI コンポーネント周辺開発での保証観点

主要活動	保証での留意事項	主な保証観点	関連する保証観点
P1-P3 開発	P1: 入力保証に対するルール実装	DI-2, SQ-3	DI-4,5
	P2: 実行中監視、冗長性の実装	SQ-3	SQ-7
	P3: 出力保証に対するルール実装	SQ-3	DI-5, SQ-6
インフラ構築	開発・運用時のデータ収集、モデル評価への仕組み	DI-1,2	CE-8, DI-3, MR-1,2,7
	運用時の監視、ロールバックへの対応	SQ-6	DI-4,5, SQ-1,7, PA-2,3
	データのプライバシー、安全性	CE-5, SQ-3	DI-5, SQ-2,4
	運用時の拡張性に対する仕組みづくり	SQ-7	PA-8
	アノテーション・ラベルなどの人の教示の仕組みがあるか	DI-2	-
	データ定義、モデル、SW の構成管理	PA-3	SQ-8
データ定義・評価	特徴量の妥当性確認	DI-2	-
	データのプライバシー、安全性確認	CE-5	DI-5, SQ-2,3,4
	評価データ（テストデータ）の妥当性確認	DI-2,3	MR-1,3
効果・リスク評価	AI コンポーネントの挙動検証に対する適切な手法の選択 (メタモルフィックテスティング、統計的評価)	SQ-6	DI-2, MR-4, SQ-1, PA-4
	誤判定、想定外に対する挙動確認と運用方法との対応づけ	SQ-3	DI-4, SQ-2,6
	システムの安全性に対する評価	SQ-3	SQ-2,4
	予測精度、実行性能に対する評価	SQ-6	MR-1,2,5

表 7.30 AI コンポーネント実装での主要活動

項目	概要
訓練データ設計	AI コンポーネントが利用するモデルを作成するために必要な訓練データを設計する。学習に適したデータとなるよう、クレンジング・水増し等を実施しデータを準備する必要がある。このデータ準備プロセスのしくみを設計する。
AI コンポーネント実装	AI コンポーネントが実システムで動作するためにアルゴリズム等を実装する。実装では AI コンポーネントに要求された指標を出力するよう、アルゴリズムへの入力データ加工処理や、ハイパーパラメータの選択といった実装を行う。
AI モデル開発	AI コンポーネントが利用するモデルを開発する。AI コンポーネント周辺開発で収集したデータを、訓練データ設計に基づいたデータセットにし、AI コンポーネントが動作できるようモデルを開発する。
モデル評価	AI コンポーネントと AI モデルが結合動作できる状態で、環境依存性や説明容易性に対応できるか評価する。

表 7.31 AI コンポーネント実装での保証観点

主要活動	保証での留意事項	主な保証観点	関連する保証観点
学習用データセット設計	学習用データセットの妥当性確認	DI-*	CE-4, MR-5
	アノテーション、ラベル付けしたものの正しさ確認	DI-2	-
	評価データ（テストデータ）の妥当性確認	DI-3	DI-5
	クレンジング、水増し、データ生成の方法は適切か？	DI-1,2	-
AI コンポーネント実装	ハイパーパラメータ選択の妥当性確認	MR-4	-
	入力データ加工処理の妥当性確認（アルゴリズム特有）	DI-2	-
AI モデル開発	学習方法の妥当性	MR-4	MR-3
	学習用データセットやハイパーパラメータなどの構成管理	MR-4, PA-3	-
モデル評価	学習から予測までの内部状態の変化を観察し確認できているか。（DNN カバレッジ等）	MR-4	-
	期待する予測精度に対する実績値の確認	MR-1,2,3	-
	環境依存性や説明容易性に対応できるかの評価	SQ-5	-

7.7.3 運用工程

運用工程では、従来のシステム運用に加えて、環境依存のデータ変動や、AI コンポーネントの帰納的動作に留意したシステムのリリース・監視・トラブル対応・保守・インフラ更新・拡張等の活動を実施する。表 7.32 で活動概要を示し、表 7.33 で保証観点を示す。

表 7.32 運用工程での主要活動

項目	概要
リリース	開発したシステムを運用先となる現場に据え付け・動作確認をしたのちに、運用先データに基づく評価を実施する。この評価条件は、環境依存性と運用要件に基づいた運用先評価条件として明確化する。
運用先効果評価	リリース時点で運用環境に対して開発時に想定した効果を得られるか評価し、運用開始判断の説明性を用意する。
監視	システムと AI コンポーネントの入出力の振る舞いを監視し、環境依存性等の要因による想定外の動作を検出する。インシデント発生時に、対応に必要な情報を収集する。
精度・パフォーマンス・容量 監視	AI コンポーネントは環境依存性により精度が低下し得る。精度低下を P2 との比較や経年変化での振る舞いの変動などの手段で監視する。この時、説明性確保には大量のデータが必要となる場合があるため、インフラのパフォーマンスやデータ容量も監視する。
トラブル対応/保守	確率的な振る舞いに対し、収集されたデータに基づき調査・対応判断する。従来型の対応では開発・検証環境で再現する方法が一般的であるが、環境依存性に対するデータ収集インフラやモデルによっては再現できない。どの程度の再現性がトラブル対応・保守で必要であるか明確にする必要がある。
インフラ更新/拡張	運用環境下で変化しそるデータ収集要件に対応する。例えば、開発時に想定されていないデータの長期保存に対応し、対象とするデータ定義の更新やデータ容量を拡張する。
モデル・学習用データ セット構成管理	運用中にモデルを更新する状況下では、AI コンポーネントが利用するモデルや学習用データセットを対応づけて保存し、説明容易性に備える。
インシデント対応	トラブル対応で緊急時のロールバック判断がなされ、モデルの更新を行うときもある。この時ロールバックにより本当に精度改善するのかを説明を用意する。
モデル更新	AI コンポーネントが利用するモデルの System Quality への影響評価に基づき、運用に利用するモデルを更新する。モデル評価の十分性にリスクがある場合は、例えば AI コンポーネントを冗長動作させ、新モデルの性能評価をする場合もある。

表 7.33 運用工程での保証観点

主要活動	保証での留意事項	主な保証観点	関連する保証観点
リリース	リリース後に問題発生した場合に被害を最小限に留める	PA-3	CE-7
運用先効果評価	環境依存性への調整・評価時間を考慮したリリース間隔	PA-3, 8	PA-7, SQ-8
	運用環境で開発時に想定した効果を得られるか	CE-1, SQ-1	CE-3
監視	入力、出力を監視し、異常が起きていないかをチェックする	DI-5, MR-1,3,9, SQ-6	CE-4
精度・パフォーマンス容量監視	データやログを、容量、プライバシー、安全性など考慮しながら適切に保存する	SQ-3	CE-4,5, DI-2, SQ-7
トラブル対応・保守	開発環境・検証環境でも再現可能とすること	MR-2	-
インフラ更新・拡張	リソース (GPU, HDD(データ領域)、ネットワーク (通信環境)) が必要に応じて拡張できるようになっているか	SQ-7	-
モデル・学習用データセット構成管理	モデル、実構成、データの組み合わせでの構成管理	SQ-7,8, PA-3	PA-7
インシデント対応	確率的な振る舞いを踏まえ、調査・修正などの必要性を判断	CE-2	CE-4, PA-7
	顧客の知識による推論と異なる結果・頻度に対する説明	CE-4, 6	CE-8, PA-7
モデル更新	データ追加の場合でも目標性能を満たすか	DI-1	CE-3, DI-3, SQ-8, PA-6,7
	データ忘却 (モデルのロールバック) リスクに対応できるか	PA-3	CE-3, SQ-8, PA-6,7
	特徴量は妥当であるか	MR-3	CE-3, SQ-8, PA-6,7
	訓練アルゴリズム・ハイパーコンフィグ等の変更有無は妥当か	MR-4	CE-3, SQ-8, PA-6,7

7.8 モチーフへの品質保証の適用

前述の観点や留意事項を仮想的なモチーフに対して適用することで、品質保証の流れやポイントを例示する。本節でのモチーフは包装機の AI を対象にしている。モチーフ検討において、包装機における AI 活用の事例としてオムロン社の論文 [5] を参考にしているものの、実際のオムロン社の製品や事例を対象としているわけではなく、記載内容はすべて仮想的に定めたものであることに留意いただきたい。

7.8.1 AI を活用する背景

ある食品メーカー A の加工工場では、製品を最後に包装する際に B 社製の包装機を利用している。包装機は製品をプラスチック製のフィルムで包み、密閉するものである。しかしながら、時々包装時にフィルムがずれることで正しく包めず、密閉が保てないことがある。この不具合が発生すると、以後の包装もフィルムずれが継続し、大量の商品不良が発生してしまうという課題があった。

そこで、包装機メーカー B に包装機の改良を打診した。B が調査したところ包装の精度をすぐに向上させることは困難なため、センサと AI を使った包装のずれを検知する仕組みを導入し、異常検知を早期にできることをめざした。

今回のモチーフでは包装のずれを検知する AI の導入プロジェクトを対象に、品質保証の流れをまとめる。

7.8.2 モチーフにおけるステークホルダー

食品メーカー A と 包装機メーカー B それぞれに、図 7.9 に示すステークホルダーが存在する。

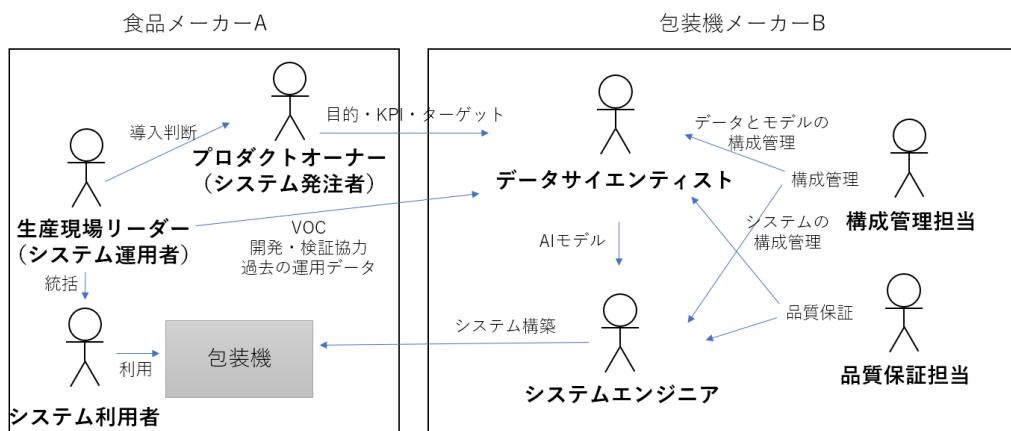


図 7.9 本モチーフでのステークホルダー

7.8.3 製品改良プロジェクトの全体の流れ

目的となっている包装のずれを検知する機能を導入するために、以下の流れでプロジェクトを進めた。

1. < PoC 工程> PoC によって、どの程度のフィルムずれが検知できるか検証し、実際の開発・導入の可否を決定した。 (7.8.4)
2. <開発工程>実際に包装ずれ検知 AI の開発を行い、既存の包装機に追加した (7.8.5)
3. <運用工程>包装機を食品メーカー A の工場で稼働させ、運用・保守を実施した (7.8.6)

以後、PoC 工程、開発工程、運用工程それぞれの開発の流れとそこでの品質保証観点を述べる。開発の流れは図 7.10 のようなアクティビティ図（スイムレーンチャート）を使って記載する。アクティビティ図のオレンジ色の四角は、前節で述べた主要活動を示し、主要活動を実現するさらに詳細なアクティビティをクリーム色の四角で示している。品質保証の観点や留意事項は黄色の吹き出しで記載し、そこには品質保証観点の ID との対応を記載した。

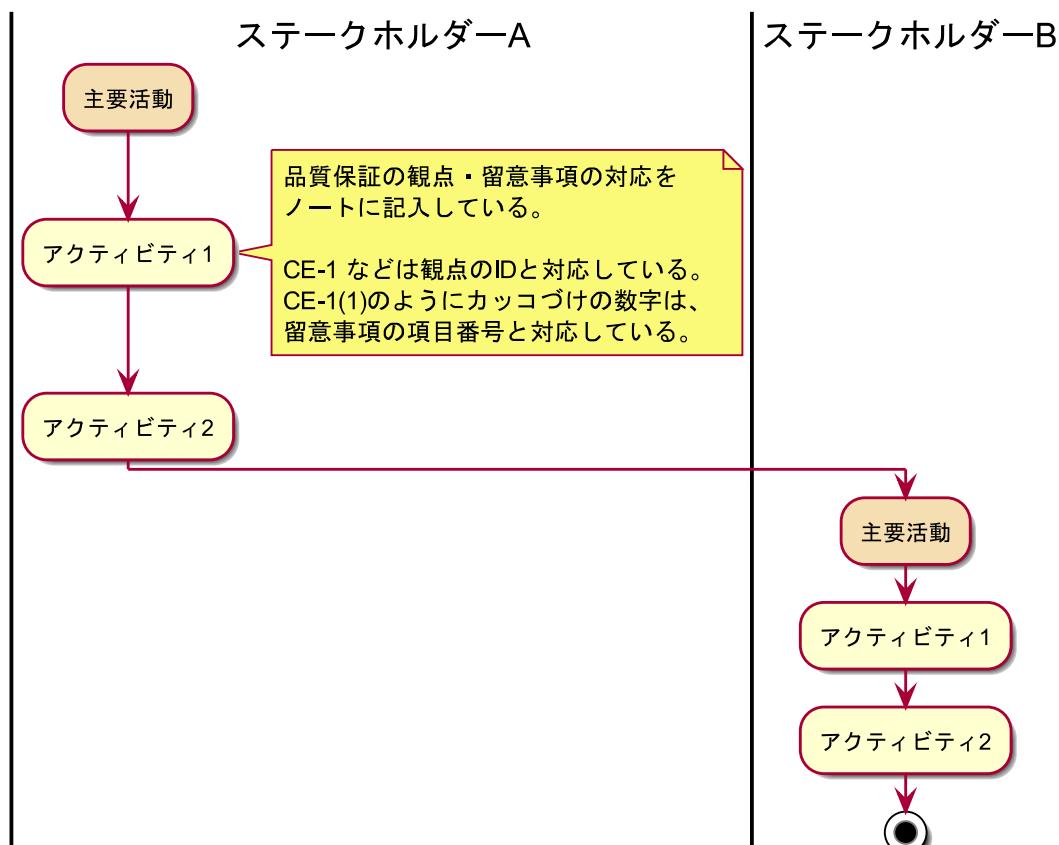


図 7.10 アクティビティ図の例

7.8.4 PoC 工程における開発の流れと品質保証

PoC 工程の開発は図 7.11 のように実施した。はじめ、目的・KPI・ターゲット定義を実施し、今回のプロジェクトの目的を再確認し、目標とする尺度を決めた。その後、データ設計やプロトタイピング、効果確認をすることでどの程度のフィルムずれが検出できるか評価を行った。これらの情報を元に、包装機に組み込んだときや運用のリスクについて抽出・確認・分析を行い、包装機に搭載する際のアーキテクチャを検討した。以上の結果を持って、顧客と実際の開発を進めていくかについて議論を行った。以下、アクティビティごとに品質保証としてどのような観点で留意したかをまとめるとめる。

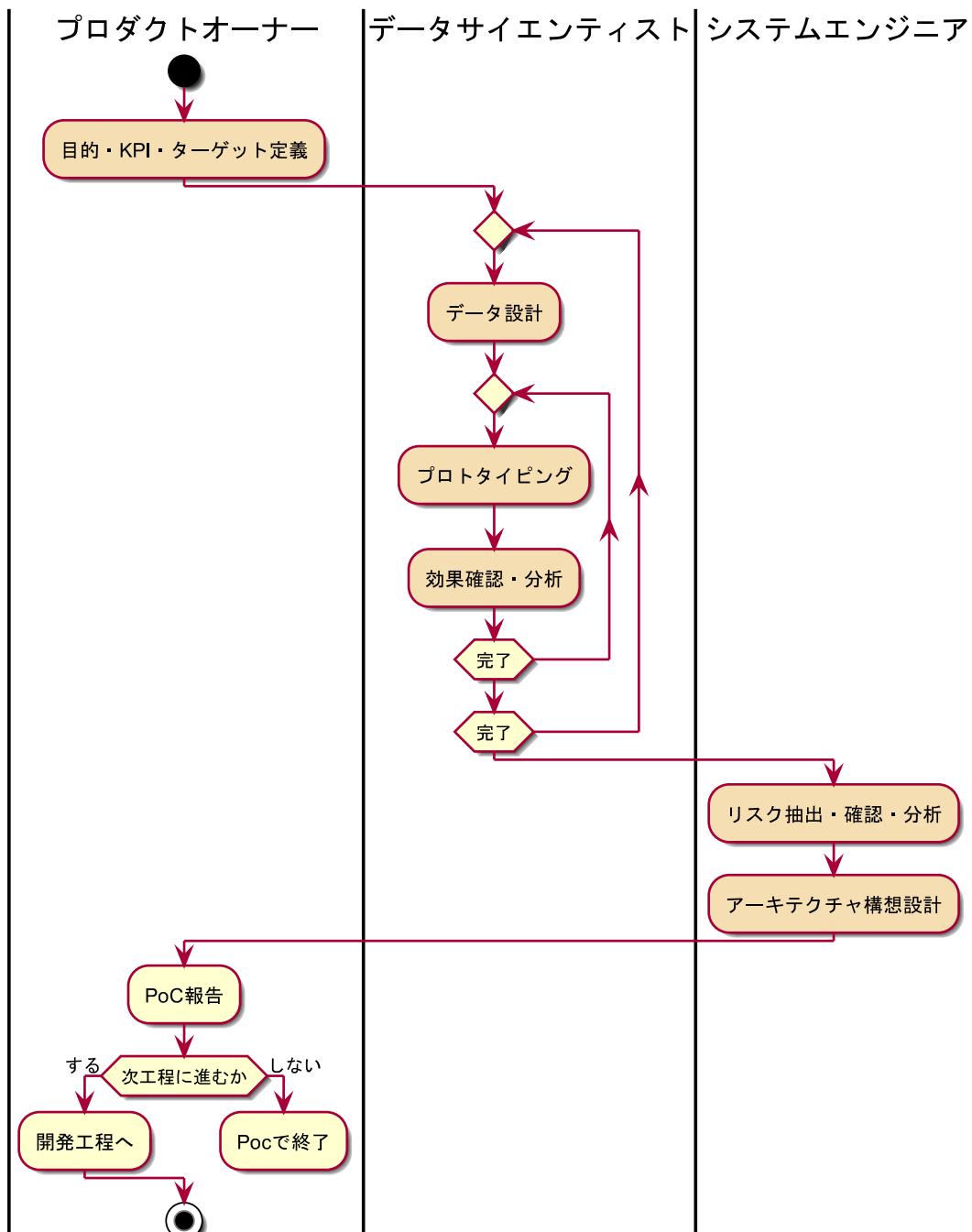


図 7.11 PoC 工程の流れ (全体図)

●目的・KPI・ターゲット定義

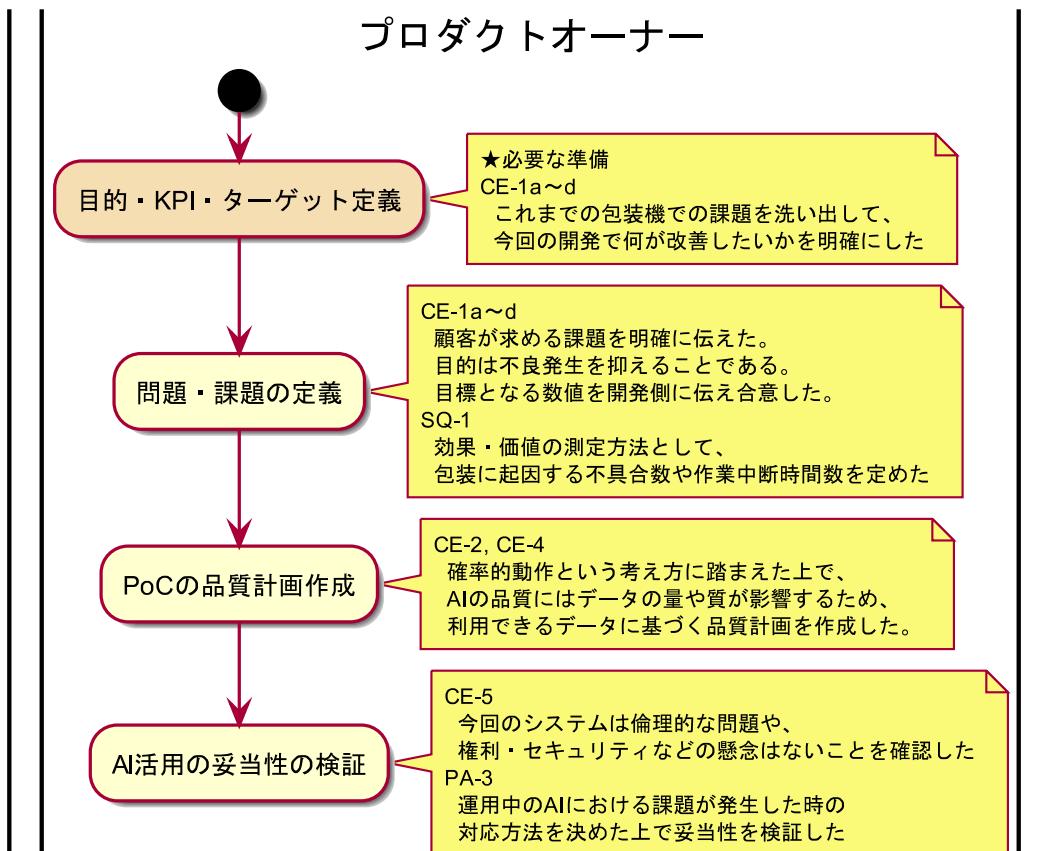


図 7.12 目的・KPI・ターゲット定義のアクティビティと考慮ポイント

目的・KPI・ターゲット定義では、食品メーカー A が達成したい目的を明確にし、今回のプロジェクトにおける目標の指標などを決める。今回のプロジェクトでは下記の通り定義した。

- 食品メーカー A では月間 200,000 個の製品を包装機でラッピングしている。既存の包装機にもルールベースによるフィルムずれ検知機能は有している。しかし見逃すことも多く、フィルムずれが直近半年間で 7 回発生し、3,000 個の不良を発生させてしまった。また作業の中止として合計 24 時間の生産できない時間が発生してしまった。
- 本プロジェクトの目的は、AI を使ってフィルムずれを早期に検出し、不良の発生を抑えることと、作業中断時間を減らすことである。KPI として包装に起因する不具合を月間 500 個から月間 100 個以下にし、作業中断時間を月間 4 時間から 1 時間に削減することとした。
- また、フィルムずれ検知機能単体の目標として、フィルムずれ検知の精度（適合率）を従来の 91 %から 95 %に向上することとした。

本アクティビティを進める上で、品質保証の観点として以下を考慮した。

- 「顧客期待の高さ」や「確率的動作という考え方の受容」という点において、AI の再現率・適合率が向上することは期待できるが、誤る可能性は依然として残ることを食品メーカー A と共有した。例えば、フィルムの蛇行があった場合も検知できない事がある。これをプロダクトオーナーや生産現場リーダーと確認した。
- PoC の品質計画作成時には、学習に使うデータについて「量や質の理解」の点で確認した。これまでのセンサデータや蛇行発生時の記録は食品メーカー A が蓄積しており、データサイエンティストに提供することが可能である。また、データは食品メーカー A で収集したものであり本プロジェクトにおいて権利の問題はない。個人情報や倫理的な課題には触れないことも確認した（「倫理的な検討」の実施）。また、システム全体の「効果・価値を計測できるか」という点では、プロジェクトの目標精度をフィルム蛇行検知の適合率・再現率ともに 95 %以上と定めた。加えて、リアルタイム処理を可能とすべく、1 回の判定には 1.0 秒未満とすることした。
- 最後に、AI 活用の妥当性を検証した。従来の仕組みでは、再現率 92 %、適合率 91 % のため、AI を活用して精度が向上できれば効果が大きい。また誤ったときには業務が停止するという問題は発生するものの、安全上の問題やセキュリティなどの問題につながることはないことを確認した（「対応方針の明確化」）。一方、AI を使うことで処理時間が長くなることが懸念される。これについては、時間制約を設けることとしている。これらをもとに、AI を活用することは妥当と判断し、AI の開発を進めることになった。

●データ設計

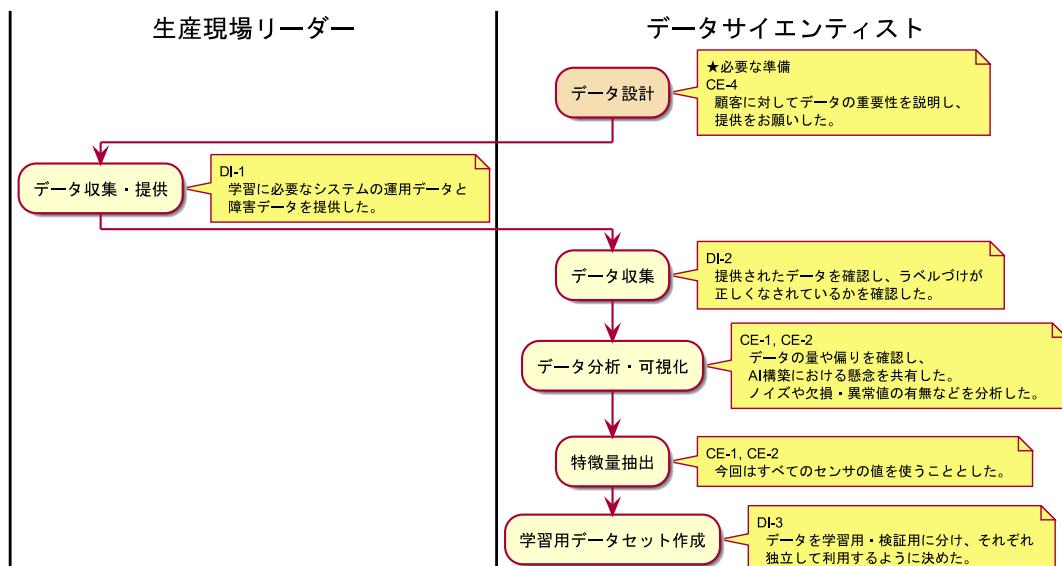


図 7.13 データ設計のアクティビティと考慮ポイント

今回のフィルムずれ検知では、包装機に組み込まれているモータのトルクセンサなどの値を活用する方針である。このため、利用できるセンサや食品メーカー A から提供してもらえる運用時のデータについて確認した上で、データ設計を実施した。

- 今回利用できるデータはモータのトルクセンサなど 8 種類の時系列データである。包装機にはあらかじめ毎秒のセンサ値を収集する仕組みを組み込んである。
- 食品メーカー A にこれまでの運用における収集データの提供をお願いした。その結果、3 月～8 月のデータと、障害が起きたときの日時が記録してある業務記録を提供してもらうことができた。包装機メーカー B ではデータと業務記録を対応づけることで AI に入力するためのデータセットを作成した。
- 包装機メーカー B はさらにノイズの調査や欠損の確認、異常値の分析などを実施した。その結果、1 週間に一度程度、特定のセンサが正しくデータがとれていないことが判明し、関連するデータは削除することにした。
- 時系列データなので、データを 1 週間ごとに分割し、交互に学習用データとテスト用データとすることにした。

本アクティビティを進める上で、品質保証の観点として以下を考慮した。

- はじめ「データの量・偏り」について分析した。今回は 1 日 8 時間の稼働で 6 ヶ月のデータが利用できるため量は非常に多いことが確認できた。一方、フィルムずれが発生しているデータは非常に少ないとや季節的に夏場に取得したデータしか存在しないことを確認し、モデル構築上の懸案事項として共有した。
- 「データの妥当性や検証データの妥当性」として外れ値やデータの欠損を分析した。今回は大きな外れ値がないことは確認できたが、日に 1 時間程度、データが欠落している場合があった。これらデータは使用しないようにデータセットから削除した。

●プロトタイピング

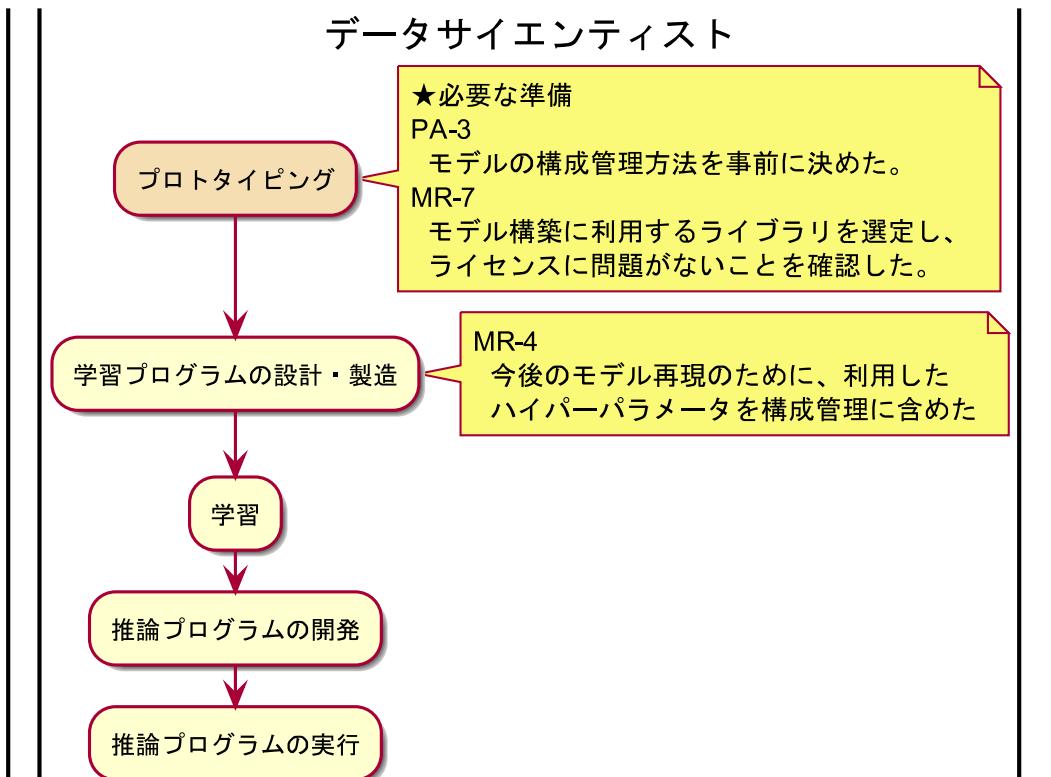


図 7.14 プロトタイピングのアクティビティと考慮ポイント

本開発は、時系列データを使った異常判別である。入力データは時系列データであるが、今回はランダムフォレストを使って実現することとした。10 秒間のトルクの値をテーブルデータとして入力し、異常の発生有無を判定する。プロトタイピングでは、学習用プログラムを作成し、学習を行った。

本アクティビティを進める上で、品質保証の観点として以下を考慮した。

- プロトタイピングを始める前に、構成管理方法を決定した。今回は、自社内に GitLab 環境を構築し、ソースコード・データ・ハイパーパラメータを git で管理するようにした。
- モデル構築のために必要なオープンソースソフトウェアであるライブラリを選定した。その際、ライセンスを確認し、商用利用可能かつソースコード開示不要であることを確認した。
- 評価のために複数のモデルを構築した。構築におけるランダムのシードや作成する決定木の数を制御するハイパーパラメータはソースコードとともに git で管理した。

●効果確認・分析

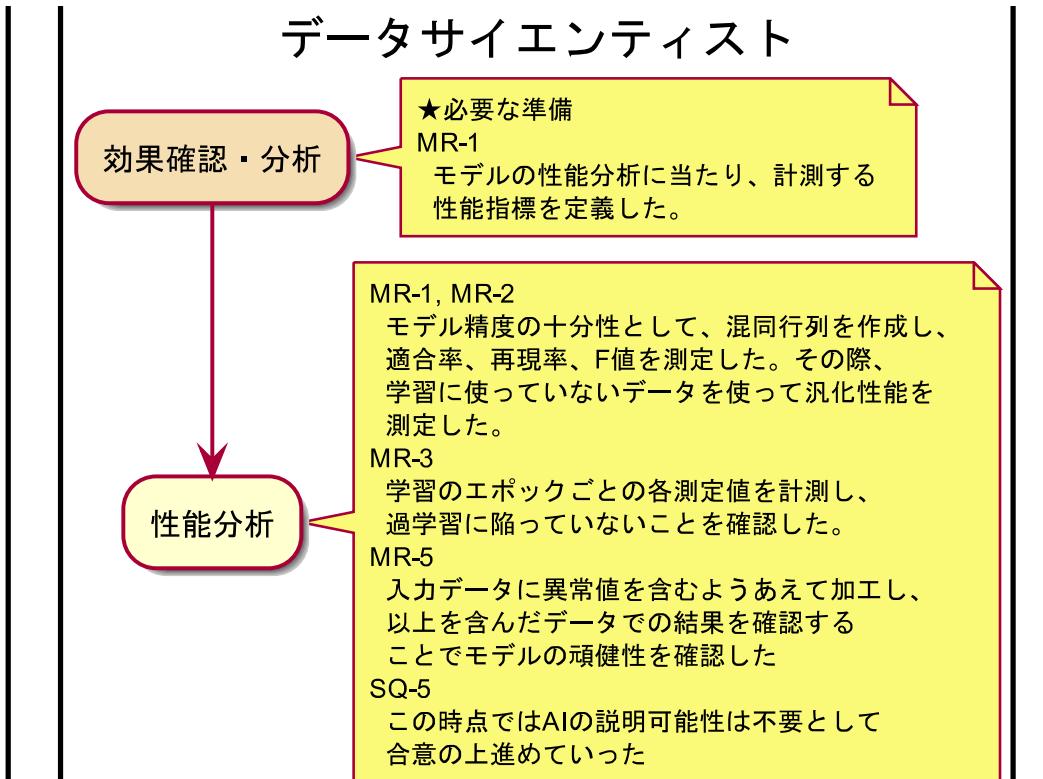


図 7.15 効果確認・分析のアクティビティと考慮ポイント

効果を確認するための指標として、適合率・再現率・F 値を測定することとし、テストデータによる評価を実施し、汎化性能を確認した。加えて、学習の妥当性として、学習のエポックごとの適合率・再現率を確認した。これは過学習に陥っていないかを確認するためである。

本アクティビティを進める上で、品質保証の観点として以下を考慮した。

- モデル精度の十分性では時季による性能の変化が重要であると考え、月ごとに評価を実施した。月ごとにテストデータを用意しフィルムずれ検知の成否に対する混同行列を作成することで、月ごとの適合率・再現率・F 値を求められる。これにより、6月の精度が悪いことがわかった。
- モデルの頑健性を確認するために、テストデータに欠落や、異常値を加えて評価を実施した。今回のモデルでは、欠落や異常値が数秒間の場合は出力に大きな影響がないことがわかった。10秒間以上の異常値が継続する場合、フィルムずれ検知が正しくないことが多くなるため、異常が継続していることを判別する仕組みが必要とわかった。
- 学習過程が妥当であることを示すために、エポックごとの MSE（平均二乗誤差）を求め、収束していることを確認した。併せて、学習データ・評価データの乖離がないことも確認した。

●リスク抽出・確認・分析

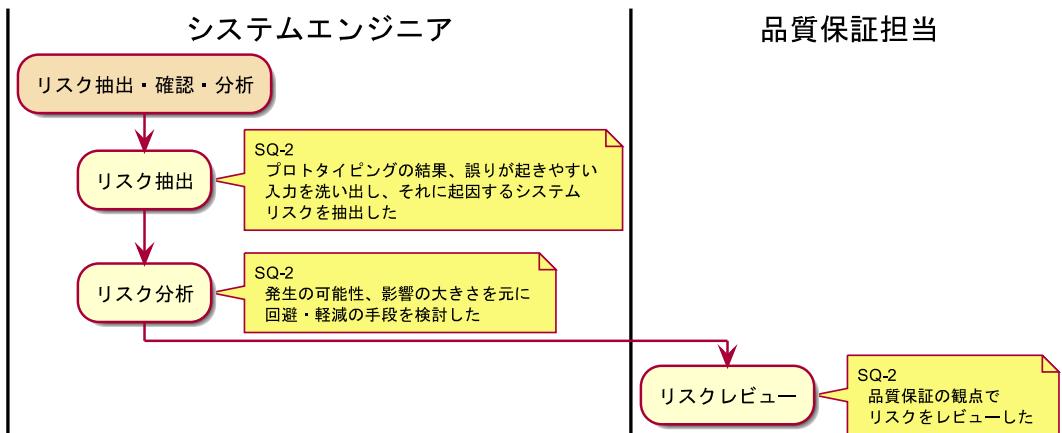


図 7.16 リスク抽出・確認・分析のアクティビティと考慮ポイント

プロトタイプによる AI 性能の結果を基に、システムエンジニアが開発におけるリスクについて議論を行った。従来のフィルムずれ検出を AI に置き換えることで、精度が向上することは試行でわかったが、時季による検知の精度の差がデータに存在する月しかわからなかった。また、今回の評価では試せていないセンサの個体差もリスクとなることを確認した。

本アクティビティを進める上で、品質保証の観点として以下を考慮した。

- 「品質事故の致命度を抑える」と言う観点からリスク分析を実施し、検出の精度が落ちる可能性のある状況を洗い出した。今回は時季とセンサの個体差が重大なリスクとして上がった。一方、安全性への影響や情報セキュリティなどもリスク抽出の中で検討したが、これらについてはリスクがないことを確認した。
- 品質事故の知名度を抑えるために、品質保証担当者のレビューを受けた。レビューでは、過去の事例等を参考にリスク抽出が網羅的であるかを確認し、問題ないと判断した。

●アーキテクチャ構想設計

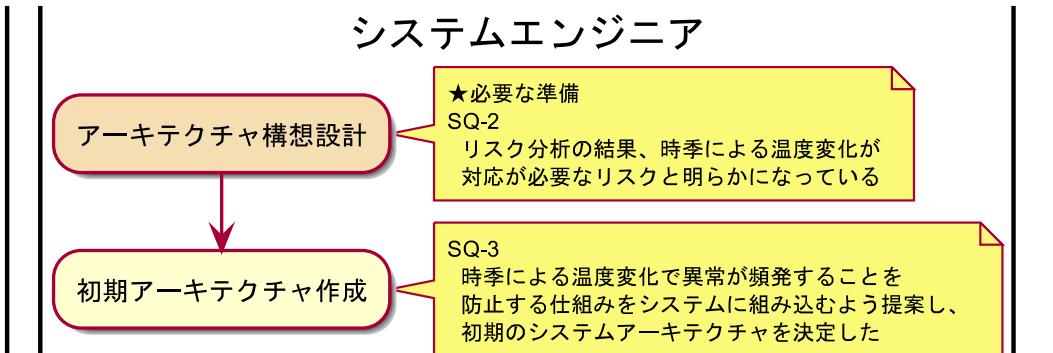


図 7.17 アーキテクチャ構想設計のアクティビティと考慮ポイント

AI のプロトタイピングと評価、またリスク分析の結果を基に、システムのアーキテクチャ構想を進めた。今回のポイントは、季節等の環境やセンサデータの欠損や異常値の継続によりフィルムずれ検知の精度が低下することである。まずはこれら気温・湿度による精度の差を抑えることや、センサの異常を早期に検出できる機構が必要と判断した。また、フィルムずれを検知した際のシステムの動作も重要である。強制停止だけでなく、人間にまず通知し、人間の判断もとれるようなシステムを設計した。

本アクティビティを進める上で、品質保証の観点として以下を考慮した。

- 「品質事故の致命度を抑える」点から、リスクとなる時季による性能の違いやセンサの個体差等について対策を施す設計とした。これにより、システムに対する事故到達度を下げることが可能になる。

以上の結果を基に PoC を終了し、顧客となる食品メーカー A に結果を報告した。食品メーカー A では導入の可否を検討し、精度については以前より向上することで満足でき、リスクは妥当で許容できるものでありシステムとして対策ができるとして、製品開発を進めることになった。

7.8.5 開発工程における開発の流れと品質保証

ここまで PoC 工程の品質保証の流れやポイントを説明した。PoC 工程の結果から開発工程に進むことが決定した場合について、引き続き品質保証の流れやポイントを説明する。開発工程は図のように実施した。今回の例はシステムインテグレーション (SI) の形態をとっているため、SI で一般的な受発注によるシステム開発の中で、データサイエンティストやシステムエンジニアが協力・分担して AI システムを開発する形となっている。また、概要フローの段階では表現されていないが、AI 開発においては現場の知見（ここでは生産現場リーダー）が非常に重要であり、それを生かすためにデータサイエンティストやシステムエンジニアに新たなアクションが必要となる場合が多い。AI

プロジェクトマネジメントの視点からも留意する必要があり、以降の説明で一部そのような想定も含めている。以下、アクティビティごとに品質保証としてどのような観点で留意したかをまとめる。

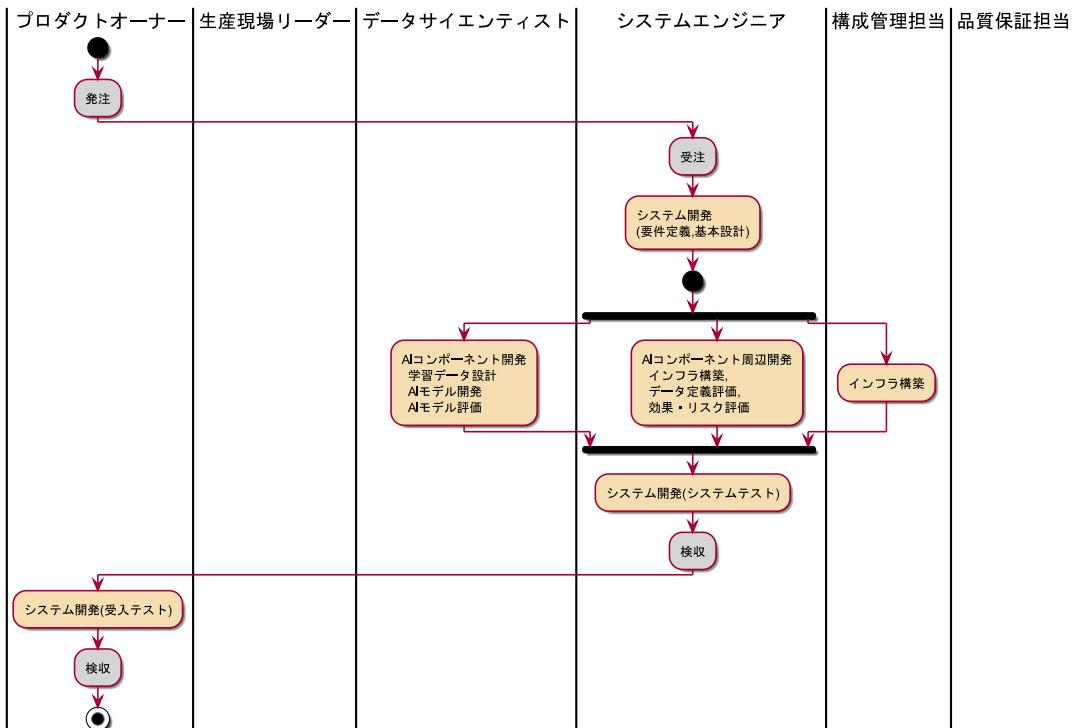


図 7.18 開発工程の流れ(全体図)

●要件定義, 基本設計

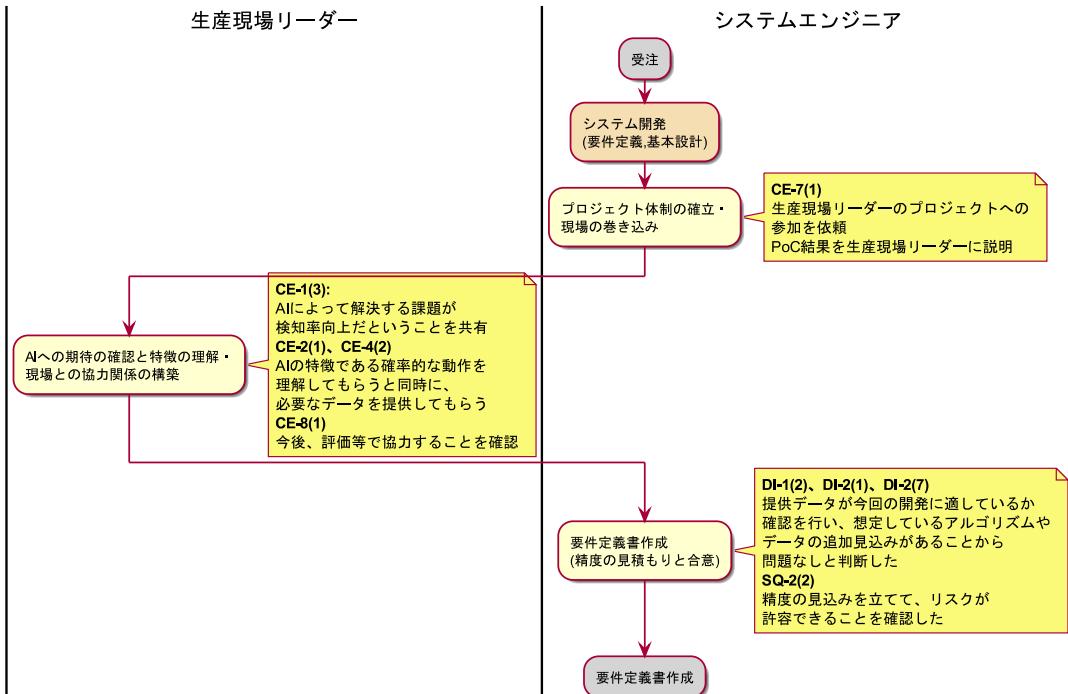


図 7.19 要件定義、基本設計のアクティビティと考慮ポイント

この段階では機能要件および性能などの非機能要件を定める。今回は PoC で一定の要件は得られているが、正式なシステム開発の契約として、改めてそれを見直し合意することが必要である。本アクティビティを進める上で、品質保証の観点として以下を考慮した。

- 特に AI 開発の場合、プロジェクトに現場の知見を反映させることが重要である。今回はその目的のため、開発チームからの要望として、生産現場リーダーもプロジェクトに加えることをプロダクトオーナーに提案し了承された。
- 生産現場リーダーに PoC の結果について説明を行ったところ、「AI ベースなので検出に確率的要素があることは理解するが、誤検出が増えるのであれば、適合率を 95 %から 97 %に向上させないと、誤検出に対応する現場作業員の追加が必要となり現場として導入が難しく、追加データは提供するので向上させることはできないか」との指摘があった。
- 開発チームで検討したところ、追加データが確実に得られる見込みであること、対策として、推論プログラムの改善、学習データの追加、推論プログラムの出力に対するルールベース判断の追加で、97 %の適合率は可能と判断し、目標を変更してプロジェクトを継続することとした。
- 本件は当初からリスクとして想定されていたため、リスク費により対応することとした。
- 本結果は発注元のお客様、生産現場リーダーにも報告し、合意された。

●設計・製造

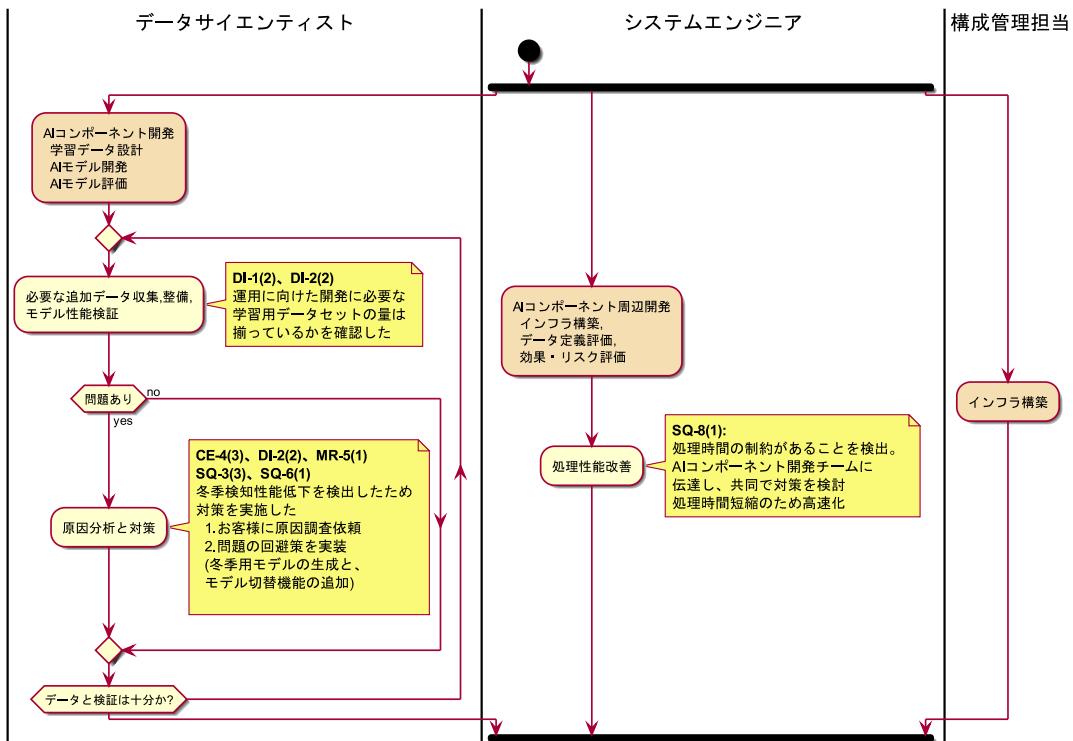


図 7.20 設計・製造のアクティビティと考慮ポイント

ここからデータサイエンティストやシステムエンジニアが協力・分担して AI システムを開発する体制となる。システムエンジニアは AI を組み込んだシステム全体の設計、データサイエンティストは、アルゴリズム選定・改善、追加データでの検証、改善などの役割分担が一般的である。このような形で本アクティビティを進める上で、品質保証の観点として、おもに以下の観点を考慮した。

- ソフトウェアやハードウェアの各種制約の検討
- 学習用データセットの質の確保
- AI モデルに影響を及ぼすノイズの洗い出し
- AI システムに対する保全性 (故障や異常を検知・診断し修復する能力) の確保

その結果、各開発チームの中で以下のような問題が新たに検出され、対策を行うことができた。

システムエンジニア (AI コンポーネント周辺開発)

<問題 1: AI コンポーネントの処理時間短縮必要>

- 包装機の制御システムとの I/F 設計の過程で、包装機はリアルタイムに包装を実行しているた

め、検出プログラムに許容される動作時間が 200[ms] であることが判明した。現在のプログラムの処理時間はそれよりも大きいため、削減が必要となる。このため、AI コンポーネント開発チームと共同で対策を検討することとした。(※検討結果は後述)

データサイエンティスト（AI コンポーネント開発）

＜問題 2：冬季検知性能低下＞

- PoC でのデータが 3 月～8 月のデータのみであったため、それ以降のデータも新たに追加した。その結果、冬季に入ってから検知性能が低下することが判明した。
- データを再度分析したところ、フィルムずれの発生率も上がっていることが確認された。モデルでこの差を季節変動ととらえて吸収することも可能ではあるが、温湿度など工場内の環境は常時一定に保たれているため、本来は冬季のみ検知性能が低下することは考えにくい。機械設備の不調等も考えられるため、お客様に結果をご報告し、原因を調査していただくこととした。
- 現時点では原因不明のため、受け入れ試験では冬季以外のデータに基づいて受け入れ可否を判断していただくこととした。ただし、現場運用を考え、冬季用とそれ以外の期間用の 2 種類のモデルを用意し、AI コンポーネント周辺開発において、それらを切り替えられる仕組みとした。大きく検知結果が増減する場合の通知手段を追加し、原因判明までは、お客様判断により適宜切り替えて運用する。
- 必要であれば、原因判明後に別途契約により新たなモデル開発を請け負うこととした。

＜問題 1：AI コンポーネントの処理時間短縮必要＞

- システムエンジニアによる AI コンポーネント周辺開発での調査から、処理時間の短縮が必要となった。当初 AI コンポーネントはすべて Python で開発の予定であったが、処理負荷の高い部分を切り出して、C/C++ などのコンパイラ言語で実装することとし、それにより処理時間の要求を満たせる見通しであることを確認した。
- この開発は AI コンポーネント開発チームで行い、組み込みと試験は AI コンポーネント周辺開発チームで行うこととした。

●システムテスト

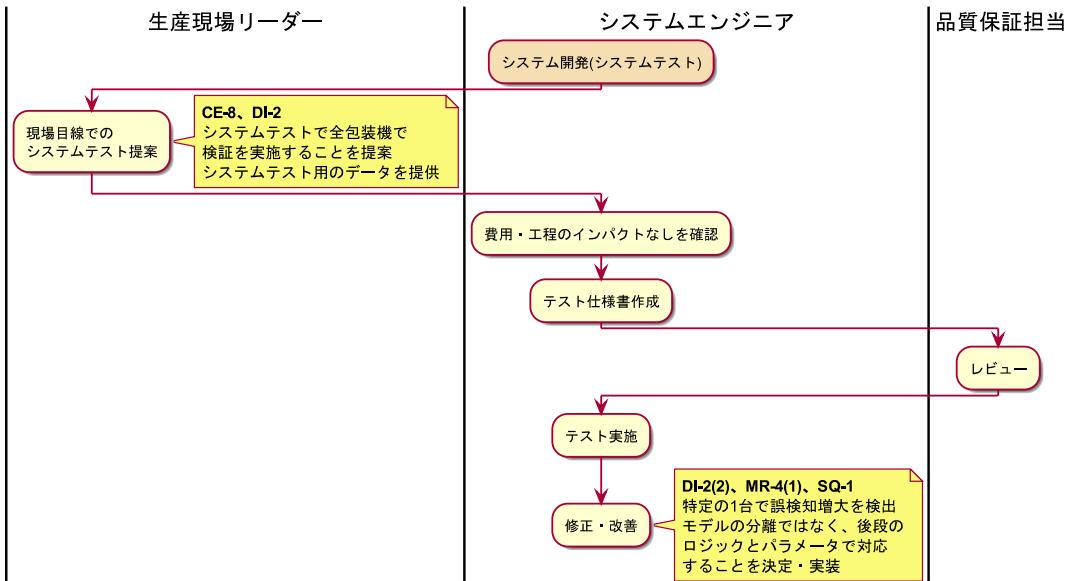


図 7.21 システムテストのアクティビティと考慮ポイント

ここからはデータサイエンティストやシステムエンジニアが分担して開発したコンポーネントを統合し、一つのシステムとして試験を行う。納入や運用前にできるだけ多くの問題を検出し、必要な対策やお客様との合意を形成することが必要であり、品質保証の観点として、おもに以下のような観点を考慮した。

- AI システムの開発に対する顧客の協力および関与
- 学習用データセットの質の確保
- 検証用データの妥当性

その結果、開発チームとお客様の生産現場リーダーのミーティングの中で以下のような提案が新たに行われ、問題の検出と対策を行うことができた。

- お客様の生産現場リーダーから、受入テスト前に極力問題を検出し排除するため、システムのテストの段階で、工場の複数の包装機すべてのデータを用いて検証を実施できないか提案があった。
- 開発チームで検討したところ、実データから学習データへの変換ツールは開発済みであり、費用・工程面のインパクトは小さいため、リスク回避の観点からも前倒しで実施することとした。

<発生問題 3: 特定包装機のみで誤検知増加>

- 複数の包装機のうち、今年度導入した 1 台で誤検知が大幅に増加した。この包装機だけモデ

ルを変更して対応するか、モデル出力の後段のロジックでフィルタする処理を加えるか、対策を検討した。

- データの分析を踏まえ、フィルタ処理で対応可能と判断した。これにより、モデルが包装機ごとに細分化することを防止し、かつ、運用前のパラメータ設定のみで対応可能とした。（※これは一例であり、原因によりモデル自体の変更が必要となるケースもありうることに注意。）

●受入テスト

開発チームおよびお客様の生産現場リーダーが密に協力して開発を行ったことで、受入テストで問題の発生はなく、無事検収された。

7.8.6 運用工程における開発の流れと品質保証

運用工程は図のように実施した。最初に食品メーカー A に包装機システムのリリースをする。ここでは、包装機を向上に納品、現場での効果確認、現場導入の判断を行った。

現場導入が可能と判断した後は運用を開始し、その中で包装機の精度、パフォーマンス、可用性を監視し、運用が正常に行われているかを確認する。

運用中に精度低下などのトラブルが発生した場合は、トラブル対応として、必要な情報を収集し、開発メンバーが解析を行う。

原因が包装機システムに使用している AI の精度による場合は、性能を改善させるために、AI の精度が低下するシーンの画像を取得し、AI モデルの学習を行う。学習したモデルをシステムに組み込み、開発でシステム評価を行い、リリース前に品質保証による品質評価を実施して、食品メーカー A にリリース可能な品質であるかを確認する。

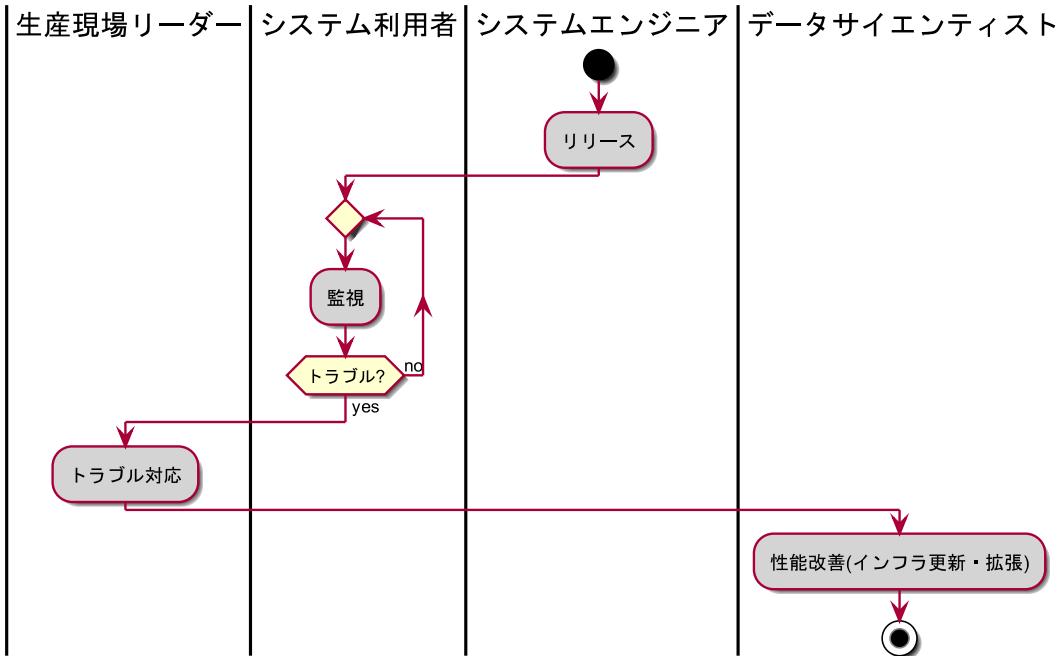


図 7.22 運用工程の流れ(全体図)

以下、アクティビティごとに品質保証としてどのような観点で留意したかをまとめる。

●リリース

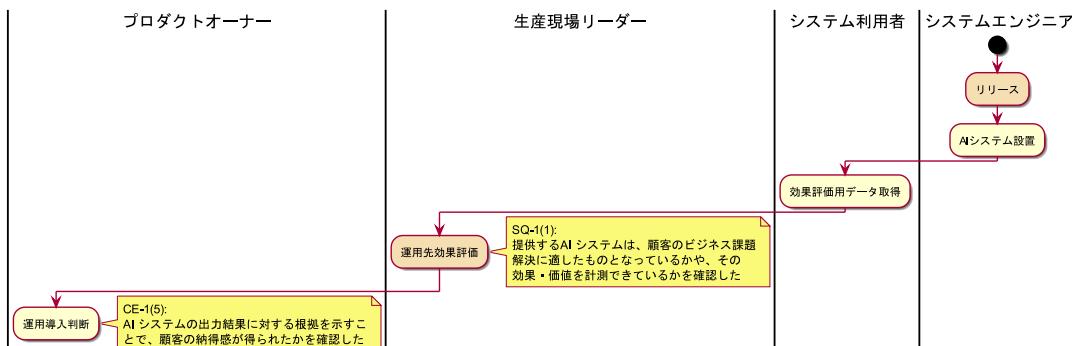


図 7.23 リリースのアクティビティと考慮ポイント

リリースでは、食品メーカー A に、包装機を納品し、工場への設置を行った。工場で包装機を使用する利用者が、運用時に使用されるデータを収集し、そのデータで生産現場のリーダーが効果確認を行い、包装機でフィルム不良検出の性能は、基準である適合率 97 %、再現率 95 %を達成していることを確認した。生産現場リーダーはプロダクトオーナーに、効果確認の結果である精度やフィルム不良を検出できなかったケースを報告し、プロダクトオーナーが現場適用が可能であると判断

し、運用の開始をすることが決まった。

本アクティビティを進める上で、品質保証の観点として以下を考慮した。

- 「提供する AI システムは、顧客のビジネス課題解決に適しているか」を確認するために、包装機でのフィルム不良の検出精度の基準は、プロジェクトの目標になっている適合率 97 %、再現率 95 % を運用中でも維持することとした。現場での運用に即したデータでの評価が必要であるため、効果確認を行うデータは包装機の利用者が収集することで食品メーカー A と合意を取った。
- 「AI システムの出力結果の根拠を示し、顧客の納得感を得る」ために、現場での効果確認評価で検出できなかったフィルム不良の種類がレアケースであり、モデルの学習に使用したデータに含まれていなかったことを説明し、性能改善を行うためには、このレアケースの画像が必要になることをプロダクトオーナーと生産現場リーダーに納得してもらった。

●監視

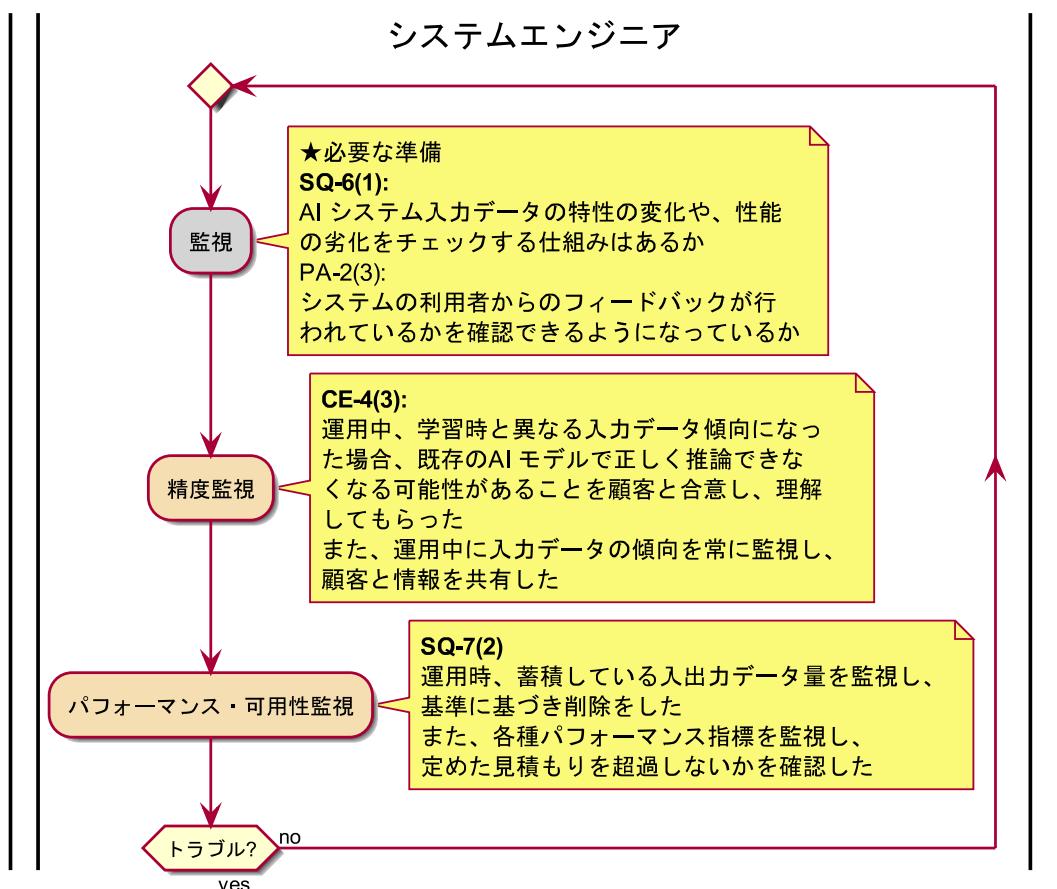


図 7.24 監視のアクティビティと考慮ポイント

運用する間は、システム利用者が使用する中で、フィルム不良の検出性能の低下や、パフォーマンス、可用性が低下した場合は、生産現場リーダーに報告する運用になっている。食品メーカー A では、今月から新製品を発売しており、その製品にも包装機が使用されている。しかし、この製品のパッケージは高級感を出すため、従来の製品では使っていないような高級感がある素材を使用していた。そのため、新製品でのフィルム不良検出の再現率は 80 %と見逃しが多く、現場で問題になっていた。

本アクティビティを進める上で、品質保証の観点として以下を考慮した。

- 「AI システム入力データの特性の変化や、性能の劣化をチェックし、フィードバックが行われる」よう、食品メーカー A とは、包装機を納入する前に、利用者が性能を確認し、フィルムズズレ検出の再現率が 85 %を下回った場合に、生産現場リーダーに報告し、生産現場リーダーが包装機メーカーの開発に連絡することの運用を決めていた。
- 「運用時、蓄積している入出力データ量を監視し、基準に基づき削除しているか。また、定めた見積もりを超過しないかを監視しているか。」は、システム利用者が包装機のストレージに保存しているデータを 3 ヶ月分だけ保存し、それ以前のデータは自動的に削除するようなシステム仕様としている。これによって、包装機のストレージ容量がいっぱいになることによって、パフォーマンスの低下、システムダウンなどが起こらないようにしている。

●トラブル対応

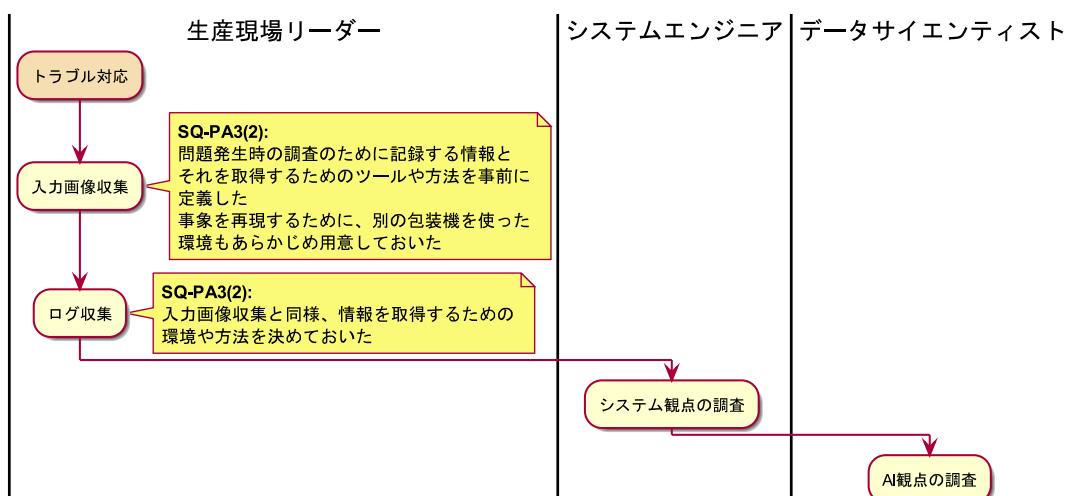


図 7.25 トラブル対応のアクティビティと考慮ポイント

新製品でのフィルム不良の検出性能の低下が起こった原因を解析するために、生産現場リーダーは包装機に入力されている新製品のデータと、システムのログを取得し、包装機メーカーのシステム開発者に提供した。システム開発者がシステムログを解析したが、システムの異常を示すような

ログは見当たらなかったので、データサイエンティストにエスカレーションをした。データサイエンティストが包装機への入力データを解析した結果、「AI モデルの学習では、ざらつきが大きい素材を使ったパッケージでのデータが無かった」ことが分かった。

本アクティビティを進める上で、品質保証の観点として以下を考慮した。

- 「問題発生時の調査のために記録する情報・取得する手順/ツールは明確か。事象を再現する環境等の準備は十分か。」の観点では、問題が発生した時の入力データ、AI が出力した結果の取得する方法を、生産現場リーダーに説明し、取得できるような準備を行っていた。

●性能改善 (インフラ更新・拡張)

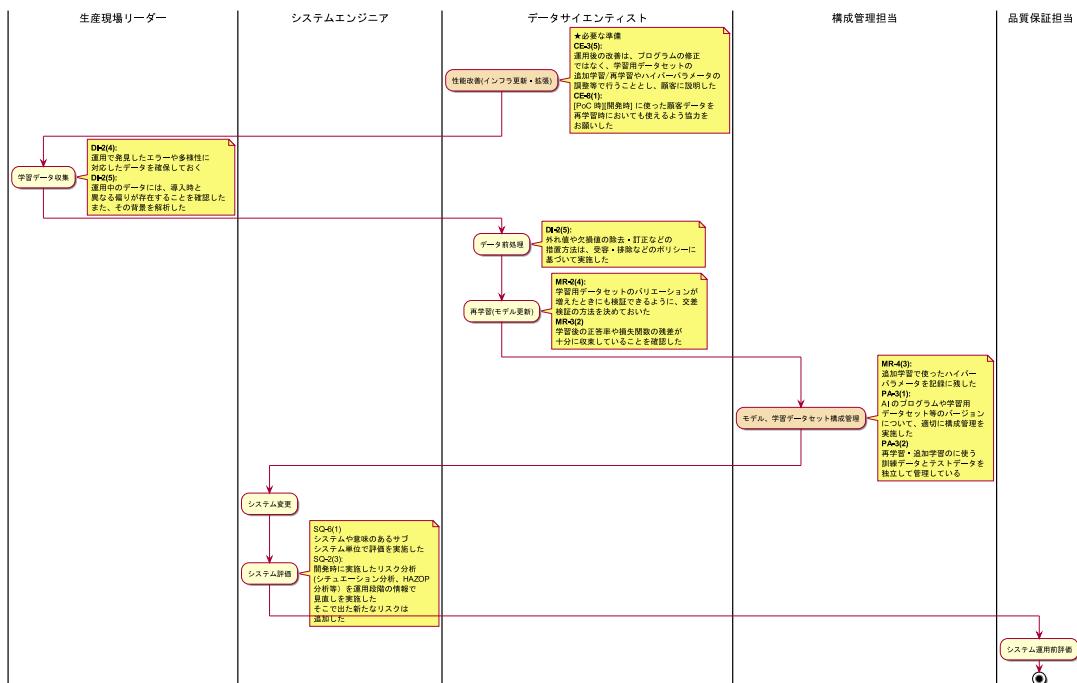


図 7.26 性能改善 (のアクティビティと考慮ポイント

包装機の性能を改善するため、AI モデルを再学習し、再学習したモデルを包装機システムに組み込むことになった。

再学習に使用する学習データを習得するため、収集するデータの要件を検討し、以下のデータを取得することにした。

- ・フィルム不良の種類（フィルムの捻じれ、フィルム破れ、フィルム位置のずれ）
- ・フィルムずれの大きさ
- ・フィルムのねじれの大きさ

生産現場の包装に関する条件は、このシステムでは一律であるため、フィルム不良の種類と程度を網羅すれば、性能改善できることをデータサイエンティスト、システムエンジニア、プロダクトオーナー、生産現場リーダーとレビューを行い、取得するデータの要件を決定した。

生産現場リーダーは要件に沿ったフィルム 510 枚分のセンサデータを現場から収集し、学習のために、データサイエンティストに提供した。

データサイエンティストは食品メーカー A から提供されたデータの分析を行い、AI モデルを再学習させるためも学習データセットを作成した。食品メーカー A から提供されたデータには、フィルム 10 枚分のセンサデータで欠損値があったため、学習データセットから除外した。また、フィルム包装が正常に行われている時のセンサデータに、フィルム不良とアノテーションがされていたため、アノテーションを付け直し学習データを作成した。

作成した学習データを用いて再学習を行った。学習には、学習データの中の 400 枚分のセンサデータを使用して学習を行った。残りの 100 枚分は評価で使用するため、学習には使用しなかった。

学習は学習率を変更しながら行い、最も正常に学習している条件のモデルを採用した。

AI を含んだモジュールの AI モデルを再学習したモデルに置き換えた。

再学習を行った AI モデル、学習ソフト、400 枚分のセンサデータの学習データセット、学習率などのパラメータは、後でトレースできるように GitLab にまとめて保存し、食品メーカー A での問題でリリースしたことが分かるようにして構成管理した。

システム開発者は包装機のソフトウェアに、データサイエンティストからリリースされた AI モジュールを組み込み、食品メーカー A で発生した問題への対策ソフトを作成した。

開発での性能評価は、学習で使用していない 100 枚分のセンサデータを用いて、学習で使用していないデータで基準である適合率 97 %、再現率 95 %を満足しているかを評価した。また、再学習によって、デグレが発生していないことを確認するため、従来の製品の画像データを食品メーカー A の生産現場リーダーから入手し、そのデータで評価を行い、問題ない事を確認した。

食品メーカー A にリリースする前に、品質保証部門で、必要な活動、評価が出来ているかの検証を行い、問題なかったため、食品メーカー A に包装機のソフトウェアをリリースした。

本アクティビティを進める上で、品質保証の観点として以下を考慮した。

- ・「運用で発見したエラーや多様性に対応したデータを確保する」ために、学習データを取得する際に、データ設計を行い、フィルム不良の種類、各不良の程度を明確にして、発生した問題に対処できる学習データを決めてデータ収集に着手している。
- ・「外れ値や欠損値の除去・訂正の根拠、措置方法について、受容・排除」は、取得した学習データに学習データの設計から外れるデータがないかの確認と各データにアノテーションされている正解ラベルの誤りがないかを確認することによって、学習データの質の向上を行っている。
- ・「学習後の正答率、損失関数の残差は、十分に収束しているか。」は、学習のパラメータを振

り、パラメータ毎の正答率と損失関数の残差の収束を確認し、最適なパラメータを採用している。

- ・「どのようなハイパーパラメータを設定して再学習を行ったか記録を残しているか。」、「AI のプログラムや学習用データセット等のバージョンについて、適切に構成管理が行われているか。」のために、AI モデルを学習して、リリースする際には、リリースバージョン、AI モデル、学習ソフトのバージョン、学習に使用したデータ、学習率などのパラメータを管理するようしている。
- ・「システムを全体として、および意味のあるサブシステム単位で評価」を行うため、包装機の包装不良の検出の性能基準は、適合率 97 % と再現率 95 % であるため、包装機のシステムレベルで評価を行った。
- ・「訓練データの特性変化や出力の追加等により再学習を行った結果、再学習前の性能に対する劣化は許容可能か」を確認するため、従来の製品の画像データを食品メーカー A の生産現場リーダーから入手し、そのデータで評価を行った。

参考文献

- [1] scikit-learn algorithm cheat sheet. https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html.
- [2] SQuBOK 策定部会. “ソフトウェア品質知識体系ガイド -SQuBOK Guide-(第 2 版)”. In: 2014. ISBN: 978-4274505225.
- [3] 稲垣敏之. “自動運転における人と機械の協調”. In: IATSS Review 40.2 (Oct. 2015), pp. 49–55. URL: <https://www.iatss.or.jp/common/pdf/publication/iatss-review/40-2-06.pdf>.
- [4] 経済産業省. “AI・データの利用に関する契約ガイドライン 1.1 版”. In: (Dec. 2019). URL: <https://www.meti.go.jp/press/2019/12/20191209001/20191209001.html>.
- [5] 広橋 佑紀 鶴田 浩輔 峯本 俊文. “マシンコントローラに搭載可能な AI 技術の開発”. In: オムロングループ技術情報誌 50.1 (May 2018), pp. 6–11. URL: https://www.omron.co.jp/technology/r_d/omrontechnics/2018/OMT_WEB_20180510.pdf.

8. 自動運転

8.1 はじめに

8.1.1 全体構成

- 8.1 節では、本章を記載した背景や目的、想定読者、前提知識やスキルなどについて示す。
- 8.2 節では、本章で使用する用語を定義する。
- 8.3 節では、自動運転の品質に限らない一般的な前提知識を示す。
- 8.4 節では、自動運転における AI 品質保証に関する特有の課題と対策について示す。
- 8.5 節では、自動運転における AI 品質保証の考え方を示す。
- 8.6 節では、ML 品質要求事項について記載する。主に品質保証部門の方を想定読者として記載している。
- 8.7 節では、ML モデル開発プロセスについて示す。主に開発部門及び品質保証部門の方を想定読者として記載している。
- 8.8 節では、ML 関連開発活動と品質保証について示す。主に開発部門の担当者を想定読者として記載している。

8.1.2 背景

- SOTIF や UL4600 など自動運転に関する安全に関する規格や、AIQM など AI 搭載システムの品質保証に関するガイドラインがそれぞれ策定されてきたが、自動運転に搭載される AI の品質保証に関しては、現場視点で纏められたものがないと思われる。
- 「高い安全性が要求される自動運転システムに搭載される AI をどのようにすれば品質保証できたとみなせるのか」について、自動運転開発に携わる多くの方の間での共通認識がない。
- 特に自動運転レベル 2~3 の AD/ADAS などに使用する画像認識用 AI の量産化が進められているが、開発や規制に關係する全ての関係者にとって、自動運転の法規・標準と AI 視点での品質保証のガイドラインが統合された自動運転システムに搭載される AI の品質保証のやり方が確立されていない。

8.1.3 目的

- 現在自動運転開発の AI 品質保証の難しさに直面している読者やこれから自動運転開発に携わる読者が、自動運転開発の AI 品質保証の全体像を掴み、結局どうやって AI を品質保証すれ

ばよいのかという問い合わせに答える一例を示す。

- このガイドラインを活用し、自社の製品開発や品質保証、顧客との合意形成に活用されることを期待している。
- 社会との合意形成を促進していくための土台として、本ガイドラインが活用されることを期待している。

8.1.4 想定読者

- AI を組み込んだ自動運転システム（主に自動運転レベル 2 3）開発の関係者（開発委託元、開発部門、品質保証部門、第三者認証機関など）

8.1.5 前提知識とスキル

- 車載開発や品質マネジメント (ISO 9001 など)、開発プロセス (Automotive SPICE など) の知識を有していること
- 機能安全に関する基礎的な知識を有していること
- AI や機械学習に関する基礎的な知識を有していること
- 「機械学習品質マネジメントガイドライン」[17] の概要を理解していること

8.1.6 想定対象製品

- 自動運転レベル 2 ~ 3 のオーナーカーに組み込まれる画像認識で周辺環境を認識する AI コンポーネント

8.2 用語集

用語集を表 8.1 に示す。

8.3 自動運転の前提知識

8.3.1 自動運転レベル

自動運転は、SAEJ3016 と呼ばれるアメリカ自動車技術会が発表している自動運転に関する 6 段階（レベル 0 からレベル 5）の自動運転レベルを定義している。（表 8.2 を参照）

表 8.1 用語集

用語	定義	備考
AI	人間が知能を使って行うことを機械にさせる ことであり、特に「機械学習」を用いて実現 する場合を指す。	AI・データの利用に関する契約ガイドライン [16] を参考にしている。
機械学習	あるデータの中から一定の規則を発見し、そ の規則に基づいて未知のデータに対する推論 や予測等を機械的に行う手法の総称。	AI・データの利用に関する契約ガイドライン [16] を参考にしている。
ML	機械学習 (Machine Learning) の略語。ここ では教師あり学習に限定する。	
ML 関連開発	自動運転システム開発の中で ML を利用する ために必要となる関連開発活動、および ML 開発自体。	
ML 品質要求	システム開発の中で ML の品質に対する要求	
ML モデル/ ML model	重みパラメータ。	
ML コンポーネント / ML component	入力データに対し訓練済みモデルを使って推 論処理を実行し、その結果を出力するコンポー ネント。例えば、画像データを入力し、物体 認識の推論結果を出力する。「8.8.2. ML コン ポーネントの品質保証」章参照	
DNN	Deep Neural Networks の略	
AIQM	「機械学習品質マネジメントガイドライン」 [17] の略称	
SEAMS プロジェク ト	「自動運転や人工知能搭載システムの安全性 を立証する技術」の研究開発を目的とした国 内プロジェクト。	
V&V	Verification(検証) and Validation(妥当性確認) の略	
コンセプトドリフ ト	推論時のデータの分布がモデルの訓練時と比 べて変化している事象。	
SOTIF	車載システムの仕様の不十分さやミスユース に対する安全性をスコープとした国際標準。	8.9.1 章を参照。

表 8.2 自動運転レベル

レベル	名称	定義	動的運転タスクの対象物・事象の検知及び応答
運転者が一部又は全ての動的運転タスクを実行			
0	運転自動化なし	運転者が全ての動的運転タスクを実行。	運転者
1	運転支援	運転自動化システムが動的運転タスクの縦方向又は横方向のいずれかの車両運動制御のサブタスクを特定の限定領域において持続的に実行。	運転者
2	部分運転自動化	運転自動化システムが動的運転タスクの縦方向及び横方向両方の車両運動制御のサブタスクを特定の限定領域において持続的に実行。	運転者
自動運転システムが（作動時は）全ての動的運転タスクを実行			
3	条件付運転自動化	運転自動化システムが全ての動的運転タスクを限定領域において持続的に実行。	システム
4	高度運転自動化	運転自動化システムが全ての動的運転タスク及び作動継続が困難な場合への応答を限定領域において持続的に実行。	システム
5	完全運転自動化	運転自動化システムが全ての動的運転タスク及び作動継続が困難な場合への応答を持続的かつ無制限に（すなわち、限定領域内ではない）実行。	システム

出典：自動車用運転自動化システムのレベル分類及び定義 JASO TP18004:2018(公益社団法人自動車技術会) より部分的に抜粋

8.3.2 一般的な自動運転システム構成

自動運転 WG が想定する自動運転システムの機能および検討の対象を図 8.1 に示す。

自動運転システムの機能は大きく「認知」「判断」「操作」の 3 つに分けられる。特に自車の周辺環境を理解する認知機能において、深層学習をベースとした画像認識 AI が応用され始めている。認知機能は、カメラなどのセンサから得られたデータを画像認識 AI で意味的情報（車、人など）に変換し、それを後段の判断機能が自車挙動を決定するために必要な情報として渡す役割を果たす。認知

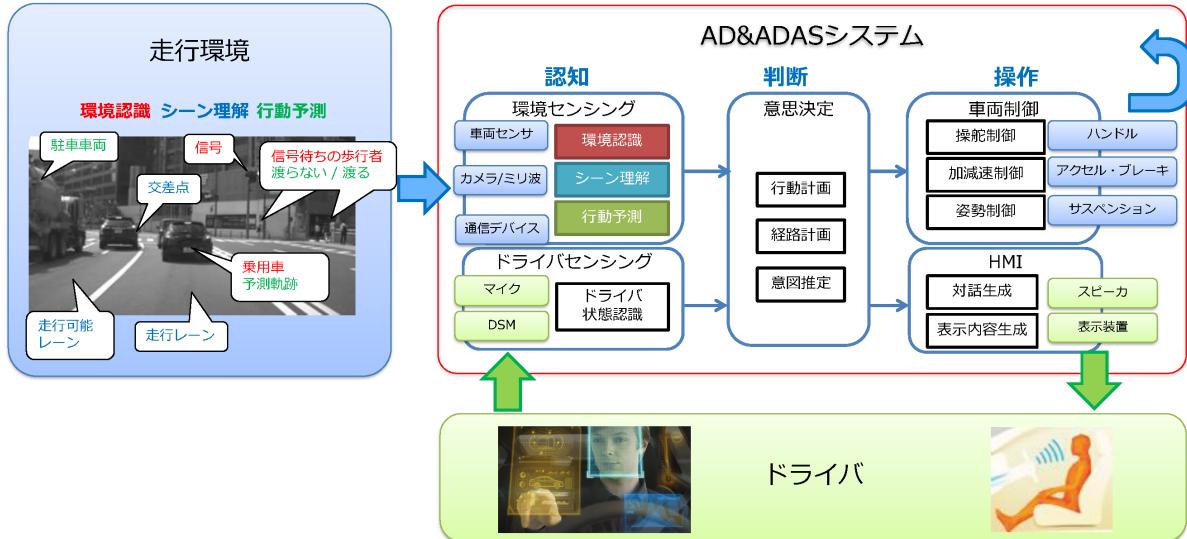


図 8.1 自動運転システムのアーキテクチャの例 [7]

の結果は判断、操作に影響を与えるとともに、品質保証の課題となるブラックボックス性が顕著な AI モデルである深層学習が認知機能の主要技術として使われることが多いため、認知機能を本 WG のスコープとした。認知機能に用いられる画像認識 AI には、走路認識（Lane Detection）や物体認識（Object Detection）など様々な応用技術が存在する。衝突を避けるべき対象が自車の進行方向に現れたときに作動することが期待される自動ブレーキ機能において、物体認識は眼前の対象物が衝突を避けるべき対象であると認識するための核心技術となる。

8.3.3 AI が使われている自動運転の機能

欧州自動車部品工業会（CLEPA）がまとめた一覧を図 8.2 に示す。データラベリングの効率化など開発段階で使われる機能やインフォテイメントなど人命リスクを伴わない機能を除くと、衝突被害軽減ブレーキ（AEBS）やアダプティブクルーズコントロール（ACC）で使われる認知機能、およびドライバステータスマニタ（DSM）のドライバ状態推定機能が、実際の製品に搭載されている AI の代表格である。認知機能やドライバ状態推定機能は、走る・曲がる・止まるといった自動車の走行機能の一部や、ドライバの居眠りや脇見を監視する機能の一部を担っていることから、セーフティクリティカルな車載 AI といえる [11]。

認知機能やドライバ状態推定機能には、画像認識タスク用の教師あり学習モデルが実装されている。認知機能では、歩行者、車両、標識などの認識対象を定義したクラスとその位置を特定する DL ベースの物体検出モデル、走行可能な領域を特定するセマンティックセグメンテーション、ドライバ状態推定機能では、ドライバの視線の推定などが一例として挙げられる。以降、本ガイドライン

で扱う AI 機能は、認知機能の画像認識タスクに用いられる教師あり学習モデルを想定する。

		定義					
AIのタイプ		人工知能(AI)	機械学習(ML)	教師あり学習(SL)	教師なし学習(UL)	半教師あり学習(SSL)	強化学習(RL)
		AIとは、環境を分析し、特定の目標を達成するために、ある程度の自律性を持つて行動することにより、知的な行動を示すシステムを指す。AIへのシステムには、仮想世界で動作する純粋なソフトウェアベースのものと、ハードウェアベースにAIを組み込むものがある。	MLとは、「学習データ」と呼ばれるサンプルデータに基づいて数学的モデルを構築し、明示的にプログラムされることなく予測や判断を行うAIのサブセットである。MLのアルゴリズムは、「教師あり学習」「教師なし学習」「強化学習」の3つに大別される。	SLは、人間によってラベル付けされたトレーニングデータを使用するMLのサブセットである。アルゴリズムはこのデータを取り込み、トレーニングセットに含まれていない新しいデータのラベルを正確に再現できるようなモデルを構築する。	ULは、ラベル付けされていない、あるいは関係性が不明な学習データを使用するMLのサブセット。このアルゴリズムは、分類されない新生のデータから、パターン、クラスタ、異常、関係、構造を自動的に識別する。ULは、データに関する新たな知識を発見し、それを持った他のMLプロセスのトレーニングに利用するために使用される。	SSLは、ラベル付けされたデータと、ラベル付けされていないデータや構造化されていないデータの組み合わせから「学習」する技術である。SSLは、少數の既知の模範例に基づいて構築され、この情報を用いて教師なしの学習を導く。	RLは、データを処理し、外部からの成功／失敗のフィードバックに基づいて行動する復習プロセスに基づくMLのサブセット。RLは、(外部から設定された)満足のいく性能閾値が達成されるまで、反復ごとに報酬関数を最大化しようと/orするモデルを開発する。
	非安全性機能 (例：インフォ テイメント、温度 制御)	対象外（ハイリスクではない） ・アプリケーションの例 ・自然言語処理		対象外（ハイリスクではない）	対象外（ハイリスクではない）	対象外（ハイリスクではない）	対象外（ハイリスクではない）
自動運転機能	認知	対象外（要求事項はAIではなくMLとのサブセットを対象とすべき） ・アプリケーションの例 ・AEBS, ACCの道路利用者検知 ・LDW, LKASの道路検知		アプリケーションの例 *縦断方向制御のための道路利用者検知（例：AEBS, ACC） *縦断方向制御のためのパシフ道路インフラ検知（例：LDW, LKAS）	アプリケーションの例 *データラベリングプロセスの効率化	アプリケーションの例 *安全性の低いシステム（例：ISA）のデータラベリングプロセスの合理化	いくつかのメーカーが認知にRLを使いつめているが、将来的には協調的認知にも使える可能性がある
	判断	対象外（要求事項はAIではなくMLとのサブセットを対象とすべき） ・アプリケーションの例 *自己車両位置と他の道路利用者に基づいたFCWとAEBSの作動				アプリケーションの例。 *ハイドモードは、制御アルゴリズムのトレーニング用として開発に使用されている。	アプリケーションの例 *laneセンタリングシステムやACCシステムでは、システムの学習に必要なコスト／データの削減のためにRLが使用される可能性がある。
	操作	対象外（AIを操作で使用しない）		対象外（AIを操作で使用しない）	対象外（AIを操作で使用しない）	対象外（AIを操作で使用しない）	対象外（AIを操作で使用しない）
	運転以外の機能	対象外（要求事項はAIではなく、MLとそのサブセットを対象とすべき） ・アプリケーションの例 ・運転者の顔を検出してIDを取得		アプリケーションの例 *ドライバーの視線／状態を検出するドライバーステータスモニタ（DSM）	アプリケーションの例 *CANデータを利用したドライバーの状態の推定		

図 8.2 AI が使われている自動運転の機能

8.4 特有の課題と対策

8.4.1 自動運転システムの特徴

自動車は多数の部品から構成される複雑な機械製品であるとともに、その制御においてソフトウェアが担う役割が大きくなっていることから、複雑な計算システムとしての側面も持つ。ここでは、ソフトウェアの観点からの自動車システムと、そのサブシステムとしての自動運転システムを考える。自動運転システムには、自動車システムの一部であることから生じる、以下の固有の特徴がある。

使用環境の多様さ

自動車は様々な地域で使用される可能性があり、自動運転システムは多様な気候、交通状況、交通規則などに対応しなければならない。不特定のユーザが使用するので使用環境を限定することは難しく、自動車が使われ得る多様な環境のもとで適切に動作することが求められる。

安全要求の高さ

自動車システムの誤動作は人命に関わるため、あらゆる状況下において重大な事故を回避できることが、自動運転システムにも要求される。さらに、自動車システムは ISO21448、ISO26262 などの安全規格や基準に適合する必要があり、自動車システムの一部である自動運転システムにも、これらの規格や基準を満たす安全性と信頼性が求められる。

システムライフサイクルの長さ

自動車のライフサイクルは構想企画、先行開発、量産開発、生産、運用、廃棄のフェーズから構成され、自動運転システムは先行開発から運用までのフェーズにおいて、適切な機能を保持し続けなければならない。特に、運用フェーズは一般ユーザが主体となるため期間を限定し難く、長期間にわたり継続的な更新を可能とするしくみが求められる。

複合システムの構成要素

自動車システムは、独立性の高いサブシステムから構成される複合システムである。このため自動運転システムには、制動システムや操舵システムなどの自動車システムを構成する他のサブシステムと相互に連携して、自動車としての統合的な機能を提供することが求められる。

サプライチェーンの複雑さ

自動車の開発・生産・保守は、完成車メーカーと部品メーカーの間の複雑なサプライチェーンを通じて行われる。自動車システムも、複雑なサプライチェーンのもとで複合システムを構築するための、固有のプロセスによって開発される。自動運転システムの開発では、そのようなプロセスとの親和性が求められる。

8.4.2 AI プロダクト品質保証上の課題

自動運転システムにおいて、機械学習により作成された ML モデルが担う役割は大きい。例えば、画像認識では認識対象を定める仕様を記述することが困難であり、認識対象のサンプルを学習させることで類似する対象も推論により認識する ML モデルの利用は欠かせない。一方、機械学習は複雑なモデルをデータに基づいて帰納的に作成する方法であるため、これまでの演繹的なソフトウェア作成方法とは異なり、作成された ML モデルのふるまいを完全に把握することは難しい。このことから、ML モデルの品質保証には、演繹的に作成されるソフトウェアとは本質的に異なる問題が存在する。

AI プロダクトに共通する品質保証の枠組みは、本ガイドラインの 2 章に記載されている。これに加えて、自動運転システムで使用される ML モデルの品質保証では、8.4.1 節の自動運転システムの特徴から以下の観点に留意する必要がある。

- 多様な使用環境に適切に対応できること
- 安全上の制約条件が常に満たされていること
- 運用時の使用環境の変化に適応できること

多様な使用環境に適切に対応できること

自動運転システムは使用される環境のバリエーションが多く、ML モデルの役割は既知の使用環境からの推論により様々な使用環境に柔軟に対処することにある。このためには、ML モデルの訓練が多様な使用環境を想定して行われていることが保証できなければならない。さらに、様々な使用環境のもとで ML モデルが安定して推論精度を確保していることも、保証できなければならない。

安全上の制約条件が常に満たされていること

セイフティクリティカルな AI システムでは、人間が AI を監視し AI がアクションを起こす前に AI のふるまいを確認できることは重要な要件であるが、自動運転システムでは人間による瞬時の確認が困難な場合もある。このため、ML モデルの推論結果が誤っていた場合にも危険な状態に陥らないようシステムが設計され、安全を確保するうえで必要な制約が常に満たされていることが保証できなければならない。

運用時の使用環境の変化に適応できること

自動車のライフサイクルは長く、その間に自動運転システムが使用される環境が劇的に変化する可能性がある。例えば、運用期間中に歩道上の歩行者の乗り物や交通規則などが大きく変わり、開発時に設定した歩行者の概念からずれが生じる可能性がある。このような使用環境の多様性の範囲を超える変化（コンセプトドリフト）にも、ML モデルを適応させる方法が提供されていなければならない。

8.4.3 AI 開発プロセスにかかる課題

機械学習を適用した自動運転システムには、入力データに対して ML モデルを使って推論を実行し結果を出力する、ML コンポーネントが組み込まれる。ここで、ML モデルの開発プロセスは学習データの収集から訓練、テストを繰り返す帰納的な過程に基づいており、他のコンポーネントの、要件定義から仕様記述、プログラム作成、テストへと進む演繹的な開発過程とは基本的に異なる。一方、自動運転システムは自動車システムを構成するサブシステムであり、その開発プロセスは自動車システムの開発プロセスの中に位置づけられなければならない。特に、自動車システムの開発には多くのサプライヤーが関わることから、各サプライヤーの責任範囲を規定する品質保証の枠組みを提供できることが求められる。さらに、自動車の長いライフサイクルを通じて自動車システムの品質を適切に保持できる、継続的なプロセスであることも求められる。

自動運転システムに特徴的な開発プロセス上の課題として、次を挙げることができる。

- 開発範囲毎の品質保証が可能な AI システム開発方法の確立
- ライフサイクルの全期間をサポートする開発プロセスの確立

開発範囲毎の品質保証が可能な AI システム開発方法の確立

自動車システムの開発において想定する使用環境の分析やリスクアセスメントは重要な項目であるとともに、それによって導出された安全制約をシステムが満たすことが厳密に検証される必要がある。一方、ML モデルの開発過程は統計的な手法による訓練と妥当性確認の繰り返しであり、ML コンポーネントのふるまいも統計的にしか評価できない。上流工程でのリスクアセスメントから下流工程での制約充足性の確認までの一連の過程を安全規格・基準に沿って行えるとともに、自動運転に関連するサブシステムも含めた総合的な品質保証が ML コンポーネントの確率性のもとで行える開発方法の確立が求められる。

ライフサイクルの全期間をサポートする開発プロセスの確立

自動車システムのライフサイクルにおいて運用フェーズが占める期間は長く、その間に開発時に想定した使用環境が大きく変化してしまう可能性がある。自動運転システムは使用環境の変化の影響を受けやすいうことから、運用フェーズを通じて継続的な使用環境の変化の分析とリスクアセスメントならびに対策が必要になる。この点から、自動運転システムの開発プロセスは出荷後も開発が継続する連続的なプロセスとみなすことができ、開発フェーズと運用フェーズが一体化した継続的な開発プロセスと品質保証の確立が求められる。

8.4.4 課題に対する考え方とアプローチ

AI プロダクト品質保証上の課題については、学習データセットの品質、ML モデルの品質、システムの品質の観点からアプローチする。一方、AI 開発プロセスの課題に関するアプローチでは、AI プロダクト品質保証上の課題に対応する開発プロセスと、AI を含まない自動車システムの開発プロセスとの親和性の両立を図る。

多様な使用環境に適切に対応できること

<考え方>

未知の入力データに対しても妥当な推論結果が得られるよう、想定される使用環境に対して十分なカバレージを持つ学習データセットを使って、ML モデルを訓練する。また、一般に ML モデルのふるまいは不安定で、入力データのわずかな差に対して推論結果が大きく変わる可能性がある。このような特性を考慮し、学習データセットの一部であるテストデータセットだけでなく、テストデータ

タセットにないデータについてもテスト・検証を行い推論精度を評価する。

<アプローチ>

学習データセットの十分性については、AI プロダクト品質保証の分類軸の Data Integrity からの評価を行う。このとき、学習させる使用環境が想定される使用環境を網羅していること、各使用環境に対して十分な学習データが用意されていることに留意して評価する。ML モデルの精度については、AI プロダクト品質保証の分類軸の Model Robustness からの評価を行う。メタモルフィックテスティング、疑似オラクルを使ったテスト、形式検証などの手法を用いて、推論精度と汎化性能ならびに入力データの微小な変化に対する推論結果の安定性の評価を行う。

安全上の制約条件が常に満たされていること

<考え方>

システム・アーキテクチャにおいて、ML モデルを使った推論を実行する ML コンポーネントとは独立した、フェールセーフのための安全機構を設ける。これにより、ML モデルの推論精度が十分に確保できない場合でも、安全機構を含めたシステムのふるまいが安全制約を満たすことを、システム設計の上で保証する。

<アプローチ>

AI プロダクト品質保証の分類軸の System Quality からの評価を行う。例えば、ML モデルを含むアクティブ・コントローラ、安全制約を満たすことが検証された安全機構であるベースライン・コントローラ、切り替えを担う判定モジュールから構成される simplex control architecture などに基づいてシステムが構成され、実行時に ML コンポーネントの推論出力を監視して安全制約に違反する可能性があると判断される場合には安全機構に切り替えるよう設計されていることを検証する。

運用時の使用環境の変化に適応できること

<考え方>

運用時の ML モデルへの入力データと推論結果を監視して入力データの統計的な分布の変化を検知するとともに、分布の変化が顕著な場合には ML モデルを再訓練して変化に適応するしくみを整備する。

<アプローチ>

AI プロダクト品質保証の分類軸の Model Robustness からの評価を行う。推論結果の不確実性などの、入力データと推論結果の分布の変化を計測するためのメトリクスにより、使用環境の変化を検知できることを評価する。また、再訓練が適切な時期に行われることと、再訓練した ML モデルが変化後の使用環境に適合することを検証するしくみがあることを確認する。

開発範囲毎の品質保証が可能な AI システム開発方法の確立

<考え方>

ML コンポーネントの入出力を明示的に規定し、他のコンポーネントとのインターフェースを仕様化することで、自動車システムにおける品質保証の基本的な考え方の一つである V & V に基づいて開発を分割する。このとき、ML コンポーネントは ML モデルを含むため機能の詳細を含まない部分的な仕様しか規定できないが、安全機構を合わせることで機能的な品質を安全機構に基づいて保証する。

<アプローチ>

ISO21448、ISO26262 などの安全規格、基準に基づく自動車システム開発プロセスをベースに、ML コンポーネントとそれにかかる安全機構を包含するよう、開発プロセスを拡張する。ML モデルの開発過程を自動車システムの V & V 型の開発プロセスに付加し、ML コンポーネントと安全機構の組み合わせについて安全制約の充足性を評価することで、V & V に基づき品質を担保する。

ライフサイクルの全期間をサポートする開発プロセスの確立

<考え方>

自動車システムの開発プロセスの中で、ML コンポーネントの開発過程の内部に ML モデルの訓練過程を位置付ける。これにより、ML モデルの訓練過程を他のコンポーネントの開発過程から切り離し、開発フェーズと運用フェーズを通じた継続的な訓練を可能にする。

<アプローチ>

ML モデルの継続的な訓練において、訓練による ML モデルの変化が ML コンポーネントに及ぼす影響の記録ならびにディグレードの評価と、ML コンポーネントと他コンポーネントとの間のインターフェースの整合性の検証を行う。これにより、ML モデルが変化した際の ML コンポーネントの品質を継続的に担保し、使用環境の変化に適合するための ML モデルの変化を可能にしつつ他コンポーネントへの影響を抑制する開発プロセスとする。

8.5 自動運転における AI 品質保証の考え方

本節では、どのように自動運転における AI を品質保証するかについて、基本的な考え方を示す。

8.5.1 自動運転における AI 品質保証の考え方とアプローチ

<考え方>

- 従来の車載開発と同等の高い品質要求に対応するため、従来と同等の品質保証を行う。

- 従来の方法で対応できない部分は、従来をベースに ML に拡張し、新たなやり方として定義する。

<アプローチ>

従来を踏襲できるところ

- ML モデルの品質は、それが組み込まれたシステムの V&V によって保証される。 ··· 8.5.5 章

従来をベースに ML に拡張するところ

- ML モデル自体の品質保証は、ML 拡張として新たに定義する。 ··· 8.5.2 章
- ML モデル開発プロセスと既存開発プロセスの I/F を新たに定義する。 ··· 8.5.3 章
- ML モデルは既存の開発プロセスにおいて規定済みのアプリケーションパラメータとして扱う。 ··· 8.5.4 章

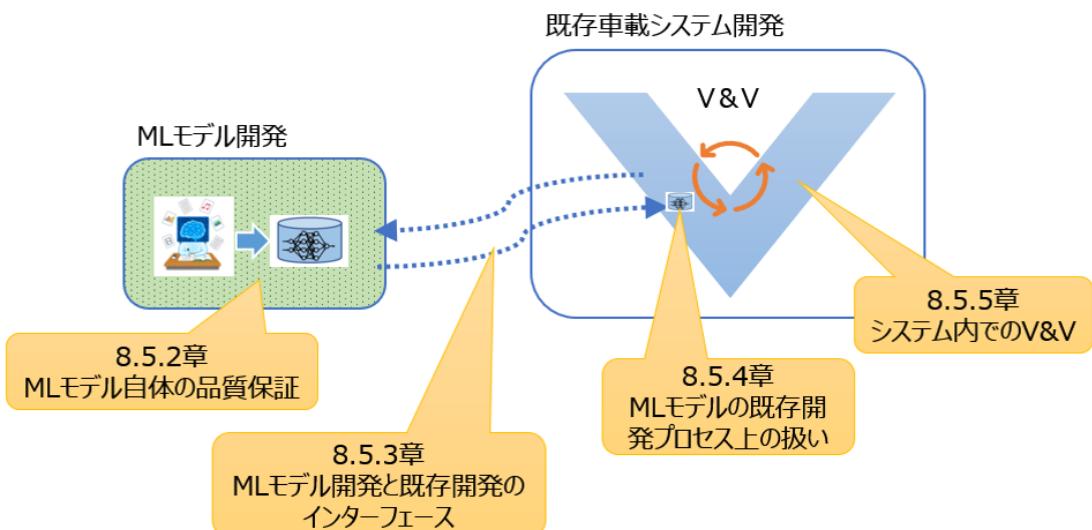


図 8.3 既存車載システム開発と ML モデル開発との関係

8.5.2 ML モデルの品質保証

ML 拡張部分として、ML モデル自体の品質保証方法を新たに定義する。

ML モデルの品質保証の考え方

- ML モデル自体も従来の車載システム開発と同じく下記により品質保証を行う。

- ISO9001：作業成果物が要求を満たしていることを客観的証拠で示すことにより品質を保証する。
- 「プロダクト品質」と「プロセス品質」の両方から品質を確認する。
 - * 「プロダクト品質」は成果物公式レビュー(作業成果物の確認)により確認する。
 - * 「プロセス品質」はプロセス監査によりプロセスの実施を確認する。
- 品質ゲートにて、フェーズ毎に「プロダクト品質」と「プロセス品質」を確認し不適合の是正を行う。
- また、ML モデル自体についても車載分野における品質保証の基本的な考え方の 1 つである V&V を適用し、正しく製品を作ったか、正しい製品を作ったかの両面で品質を確認する。

ML モデル開発のプロセス

- 基本的な考え方
 - ML モデル開発はソフトウェア開発とは開発方法が異なるため、新たに ML モデル開発プロセスを定義する。
 - プロセス監査により、プロセス通りに開発が実施されていることを確認する。
- 具体的な扱い
 - 新たに ML モデル開発プロセスを定義し、自社の既存開発プロセスに追加する。
 - ML モデル開発プロセスの例を「ML モデル開発プロセス」章に示す。
 - 備考) 「ML モデル開発プロセス」は複数のプロセスから構成されてもよい。

ML モデルのプロダクト品質

- 基本的な考え方
 - ISO/IEC CD 25059 Software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) - Quality model for AI systems を参照する。
 - ML モデルに関する品質を確認するための軸、つまり品質特性をふまえた上で、当該プロダクトに必要とされる品質特性とそのレベルを規定し、ML モデルに対する品質確認を行う。
- 具体的な扱い
 - ISO/IEC CD 25059 はまだ発行されていないため、ML モデル用の品質特性と品質要求事項を定義する。
 - ML モデルに求められる品質要求事項は、QA4AI のみでは足りないため、他のガイドラインや標準も利用する。
 - AIQM に規定されている品質特性は標準化を見据えて体系的に整理されているため、ML モデル用の品質特性として使用する。
 - AIQM は分野横断的なガイドラインであるため、自動運転分野において解釈して適用

する。

- ML モデルのプロダクト品質に関する品質要求事項の例を「ML 品質要求事項」節の「ML モデル品質要求事項」に示す。
- これらを参考に各製品毎に必要となる品質要求事項を定め、作業成果物が品質要求事項を満たしていることを確認する。

ML モデルの検証（Verification）

- 基本的な考え方
 - ML モデルの検証にソフトウェア開発の検証方法を適用することはできない^{*1}。そのため、ML モデル用の検証方法を新たに定義する。
 - ML モデルの検証すなわち正しく作られているかの確認は、
 - * ① ML モデルにあった正しい作り方を定義し、その通りに作られていること確認すること
 - * ②作成した ML モデルの振る舞いが、ML モデルに対する要件に対して意図通りの振る舞いであるかを評価により確認すること、である。
 - ML モデル開発は、正しい作り方をしたからといって意図通りの推論結果を出力する ML モデルを開発できるとは限らない。そのため、正しい作り方で作った上で、ML モデルの振る舞いが ML モデルに対する要件に対して意図通りであるかを評価することによって、正しく作られたことを確認する。
 - ML としての性質（非線形や確率的な振る舞いなど）を評価し、その程度を定量的に十分把握できている状態にし、それが意図した範囲内であることを確認できていることが ML モデルの検証のゴールと考える。
- 具体的な扱い
 - ML モデルについての「正しい作り方」を定義し、その通りに作ったかを確認する。
 - 「正しい作り方通りに作ったか」の確認は、レビュー及びプロセス監査で確認する。
 - レビューでは、作業記録や作業成果物の内容を確認することにより、「正しい作り方」が出来ているかを確認する。技術的な内容の確認は、有識者による技術レビューが必要となる場合がある。
 - プロセス監査は、作業記録や作業成果物によりプロセスの実施状況を確認し、正しい作り方でく作ったかを確認する。ML モデル作成のプロセスは「ML モデル開発プロセス」章にて例示する。

^{*1} ソフトウェア検証方法を適用できない理由：ML モデルの開発は帰納的開発と呼ばれ、計算や判断を行うアルゴリズムを訓練データから獲得する。これは人がアルゴリズムを設計しプログラムに実装する従来の演繹的なソフトウェア開発とは異なる。帰納的に獲得した ML モデルの重みパラメータの本質的な意味を理解することは難しく、個々のパラメータが設計通り正しく実装されているかをテストで検証することはできないし意味をなさない。

- ① ML モデルの「正しい作り方」は、ML 開発の一般的な ML モデル開発のフローに従うことになるが、ML 技術の進化も速いことから、ここでは、大枠のみをプロセスとして例示することとし、作り方に関しては現行必要と考えられる項目に関してのみ品質要求事項として定義する。「ML 品質要求事項」の節にて「ML モデル品質要求事項」の中の「内部品質要求」として整理する。詳細な作り方と品質基準については、各組織において適宜定義、追加して欲しい。
- ②作成した出来上がった ML モデルの振る舞いが、ML モデルに対する要件に対して意図通りであるかの評価による確認は、ML モデルのに対する意図した振る舞いに対する要件仕様と検証基準を定め、その検証基準を満たすことをテストなどで確認することである。
- 例えば、推論精度を評価し、合格基準以上であればテスト合格とすることである。具体例としては、入力画像に雨や霧に相当するノイズを追加した際の推論結果の差分を評価し、検証基準値以内であればテスト合格とすることである。
- ML モデルの振る舞いに対する要求は、「ML モデル品質要求事項」の中の「外部品質要求」として整理する。

ML モデルの妥当性確認（Validation）

- 基本的な考え方
 - ML モデル単体としては製品として動作しないため、ML モデルの妥当性は、ML モデルを組み込んだ製品（車両レベル）あるいはコンポーネントにおけるでの妥当性確認により確認される。が完了して、初めて完了する。
- 具体的な扱い
 - 最終的な妥当性確認は、量産車両や量産試作車両での評価やテストまで行うことはできないが、その時点での手戻りは影響が大きい。
 - そのため、先行開発初期の段階から車両レベルでの妥当性確認を進めることを強く推奨する。
 - 実開発においては、実車で取得した外界データを入力とした ML コンポーネントレベルの仮想テスト環境で、妥当性確認を実施する。

8.5.3 ML モデル開発と既存開発プロセスの I/F

帰納的開発である ML 開発を、演繹的開発である従来の車載開発プロセスにどのように組み込めばよいかを明らかにする。

帰納的開発と演繹的開発の I/F

- 開発方法をブラックボックスとして扱い「in:前提条件」と「out:作業成果物」を I/F とする。これにより開発方法に依存せず双方向に前提条件と作業成果物の受け渡しができる。
- 開発プロセスでは、要件仕様化とそれに基づいた作業成果物を作成することから、IN は前提条件、OUT は要件仕様とそれを満足した作業成果物とする。

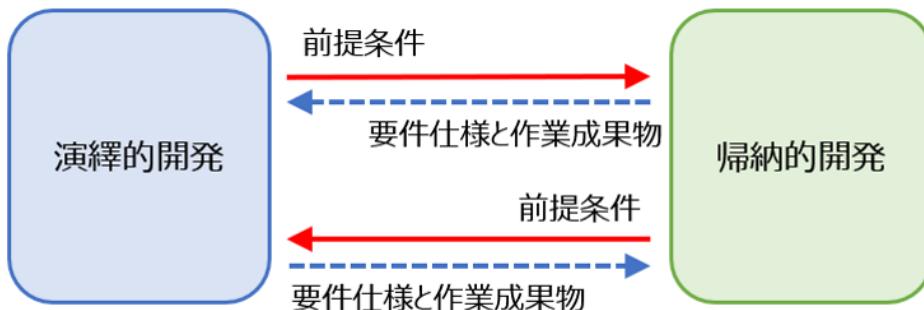


図 8.4 演繹的開発と帰納的開発の I/F 概念図

車載開発プロセスと ML モデル開発の I/F

- ML 開発において帰納的に開発される最終成果物は ML モデルである。そのため、帰納的開発部分は「ML モデル開発」とする。
- Automotive SPICE における「SYS.3 システムアーキテクチャ設計」の中で、ML モデルに割り当てられたシステム要件が、ML モデル開発の入力となる。
- 既存開発プロセス(演繹的)と ML モデル開発(帰納的)の I/F を以下に示す。
 - IN
 - * システム要件仕様
 - * システムアーキテクチャ設計
 - OUT
 - * ML モデル要件仕様
 - ・機能仕様
 - ・性能仕様：処理速度
 - ・構造仕様：モデルフォーマット、モデル構造
 - ・データセット仕様：アノテーション仕様、データセット
 - ・前提条件：使用 HW

- ・制約事項：リソース、運用環境
- * ML モデル
- * ML モデル評価報告
- * データセット
- * データセット評価報告
- * トレーサビリティ記録
- * 各種レビュー記録
- * その他作業成果物

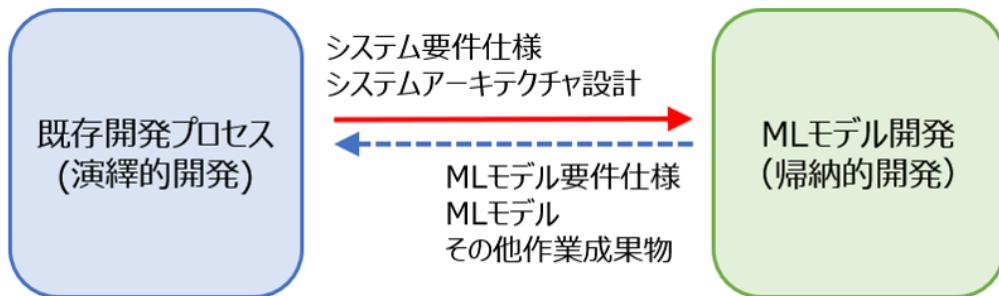


図 8.5 既存開発プロセスと ML モデル開発の I/F

8.5.4 ML モデルの既存開発プロセス上での扱い

ML 開発の主な成果物である ML モデルを既存開発プロセス上でどのように扱うか記載する。

ML モデルの既存開発プロセス上での扱い

- 基本的な考え方
 - 既存の開発プロセス上では ML モデルは「アプリケーションパラメータ」として扱う。
 - ML モデル (DNN モデル等) は試行錯誤や実験等により決められたパラメータセットである。
 - 車載開発において、同じく実験等により決められたパラメータセットとしてエンジンパラメータを代表とする「アプリケーションパラメータ」がある。
 - 開発プロセスにおいて、同じ性質のものは同じ扱いにする方が、既存の開発プロセス上で扱いやすくリーズナブル（合理的）である。
 - そのため、車載開発プロセスにおいては、ML モデルは「アプリケーションパラメータ」

として取り扱う。

- 具体的な扱い
 - 「アプリケーションパラメータ」は、開発プロセス上ではシステム要件の 1 つとして扱われている。
 - ML で達成する性能や訓練データなどのシステム要件を、「SYS.3 システムアーキテクチャ設計」にて ML モデルというシステムエレメントに配置する。そのシステムエレメントは、ML モデル開発プロセスで実装され、「SYS.4 システム統合および統合テスト」で統合される。
 - ML で達成する性能や訓練データなどのシステム要件および ML モデルは、PoC や先行開発を通して段階的に成長、成熟する。

8.5.5 ML モデルを統合したシステムの V&V

ML モデルを統合したシステムの、車両・システムレベルの V&V について記載する。

ML モデル統合システムの車両・システムレベルでの V&V

- 基本的な考え方
 - ML モデルをシステムに統合し、各統合レベルで検証を行う。
 - ML コンポーネントレベルと車両レベルでは検証と妥当性確認を行う。
 - ML モデルに関する統合レベルとして少なくとも以下の 3 つがある。
 1. ML コンポーネントレベル（検証と妥当性確認）
 - * ML モデルと推論エンジンを統合したソフトウェアレベルのコンポーネント
 - * ML モデルと推論エンジン、ソフトウェアの統合テストを実施する。
 - * 市場環境のデータを使った妥当性確認を行う。
 2. ML セーフティーアーキテクチャレベル（検証）
 - * 外部の安全機構が施されたアーキテクチャレベル
 - * 外部安全機構に対する安全設計の検証を行う。
 3. 車両レベル（検証と妥当性確認）
 - * ML モデルを搭載した車両レベル
 - * ML を搭載したシステムとしての統合テストを行う。
 - * ML を搭載した車両レベルの妥当性確認を行う。
 - 「ML モデル」の品質は、それが組み込まれたシステムの V&V によって保証される。
- 具体的な扱い
 - ML コンポーネントレベルでは、評価ボードや試作 HW などの実機テスト環境にて、処理速度と性能のトレードオフの試行錯誤を行うことが多い。このレベルでは、処理速度、

推論性能などの検証を行う。市場環境のデータを用いた妥当性確認もこのレベルで行う。

- ML セーフティアーキテクチャレベルでは、フォールトイインジェクションテストによる AI の誤り（誤判定など）についての安全性を検証する。
- 車両レベルでは、ML モデルを含む車両レベルの検証と妥当性確認を行う。
- 最終的な ML モデルの妥当性確認は、量産試作車両や量産車両での評価やテストまで行うことができないが、その時点での手戻りは影響が大きい。そのため、先行開発初期段階から、車両レベルの妥当性確認と市場環境のデータを使った ML コンポーネントレベルでの妥当性確認を行い、ML モデルの品質を成熟させておくことが重要である。

8.6 ML 品質要求事項

本節では、自動運転における ML 品質要求事項について記載する。ML 品質要求は、システム開発における ML 利用に関連する品質要求である。主に品質保証部門の方を想定読者として記載している。

以下に記載する ML 品質要求事項は、各社の開発プロセスの中で確認するように定義されることを想定している。また、開発委託元が委託先への要求事項として使用しても良い。

8.6.1 節では、ML 品質要求の考え方とアプローチについて示す。

8.6.2 節では、ML モデル要求事項の構造と分類について示す。

8.6.3 節では、ML 品質要求事項の一覧を示す。

8.6.1 ML 品質要求の考え方とアプローチ

<考え方>

- 従来の車載開発と同等の高い品質要求に対応するため、従来と同等の品質保証を行う。
- 従来の方法で対応できない部分は、従来をベースに ML に拡張し、新たなやり方として定義する。

<アプローチ>

- 従来を踏襲できるところ
 - 国際規格 ISO/IEC 25000 SQuaRE シリーズ [8] の品質要求事項に関する基本的な考え方
 - 従来をベースに ML 拡張するところ
 - ML モデルに対する要求の扱い
1. 要求事項の分類は JIS X 25030[9] 図 9 「システム要求事項の分類」を利用する。
 2. システム要求事項の分類内に「ML モデル要求事項」を ML 拡張として新たに定義する。
 3. ML モデル要求事項内の各品質要求を下記のように整理する。

- 利用時品質要求：ML モデル搭載製品利用時の品質への要求（妥当性確認に利用される）
 - 外部品質要求：ML モデルの振る舞いに関する品質への要求
 - 内部品質要求：ML モデルの作り方に関する品質への要求
4. ML 品質要求事項には、開発プロセス、開発組織に対する要求も含まれる。
5. 「内部品質要求」には機械学習品質マネジメントガイドライン [10]（以下 AIQM）の内部品質特性を利用する。

8.6.2 ML モデル要求事項の構造と分類

図 8.10 に ML モデルに関連するシステム要求事項の構造を示す。

- SO/IEC 25030 (JIS X 25030) SQuaRE シリーズのシステム要求事項の分類に従い、ML モデル要求事項の構造を整理する。
- 帰納的に開発される「ML モデル」については、ソフトウェアとは開発方法が根本的に異なるため「ML モデル要求事項」として新たに分離して整理する。
- 「ML モデル要求事項」内の構造は JIS X 25030 図 9 システム要求事項の分類を参考に ML に拡張した。

システム要求事項	ソフトウェア要求事項	ソフトウェア製品要求事項	固有の特性への要求	機能的 requirement 事項				
				利用時の品質要求事項				
				ソフトウェア品質要求事項		外部品質要求事項		
				内部品質要求事項				
システム要求事項	ソフトウェア開発要求事項	割り当てられた特徴の要求事項			例えば、価格、配布日、製品の先行き及び製品の供給者に対する要求事項を含め、管理面での要求事項			
		開発プロセス要求事項			開発組織要求事項			
その他のシステムの要求事項	例えば、コンピュータハードウェア、データ、機械部品及び人手による業務プロセスへの要求事項を含む							
MLモデル要求事項	MLモデル製品要求事項	固有の特性への要求	機能的 requirement 事項					
			MLモデル品質要求事項		利用時の品質要求事項			
		割り当てられた特徴の要求事項	例えば、価格、配布日、製品の先行き及び製品の供給者に対する要求事項を含め、管理面での要求事項		外部品質要求事項	内部品質要求事項		
MLモデル開発要求事項		開発プロセス要求事項			開発組織要求事項			
		開発組織要求事項						

図 8.6 ML モデルに関連するシステム要求事項の分類

図 8.7 は、ML モデルに関連する要求事項の構造の説明である。

- 青線で囲んだ部分は「JIS X 25030 図9 システム要求事項の分類」である。
- 赤線で囲んだ部分が ML 拡張として新たに定義した「ML モデル要求事項」である。
- 帰納的か演繹的かで開発方法が本質的に異なるため、要求構造としても分離した。

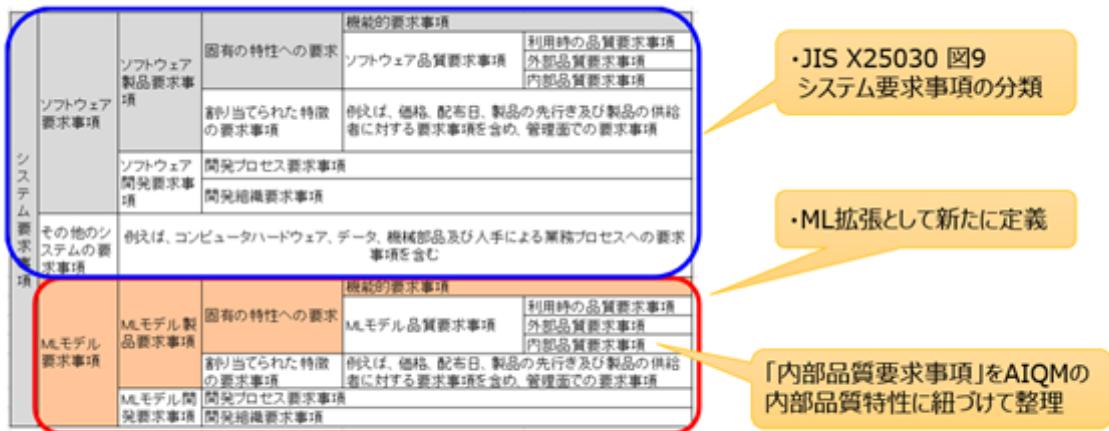


図 8.7 ML モデルに関連する要求事項の分類の説明

ML モデル要求事項の分類の詳細を、以下に示す。

- 製品要求事項
 - 利用時品質要求事項
 - * 「ML モデルを組み込んで製品として利用する際の品質要求」として整理する。
 - * 妥当性確認に利用される。
 - * 例えば、市場環境のデータを使った車両レベルの妥当性確認の要求事項などである。
 - 外部品質要求事項
 - * 「ML モデルを推論エンジン上で動作させた時の振る舞いに対する品質要求」として整理する。
 - * 例えば下記についての要求である。
 - ML モデルの正確性
ML モデルの性能に対する要求 (精度、適合率、再現率、F 値など)
 - ML モデルの安定性
レンズの汚れや雨や霧などのノイズに対する頑健性の要求
 - * 注意点

- ・ ここでの「外部品質」は AIQM で用いる「外部品質」とは一致しない。AIQM の「外部品質」は、要求される品質の「レベル」を設定するが、品質指標として必ずしも直接的に測定できるものではないと考える点に注意が必要である。
- ・ ここでは ML モデルの振る舞いに対する品質要求とし、例えば ML モデルの精度などの具体的な要求を扱うものとする。
- ・ 外部品質特性の項目(軸)については、今後の検討課題とする。
- 内部品質要求事項
 - * 「ML モデルの作り方に対する品質要求」として整理する。
 - * AIQM の内部品質特性と対応付ける。
 - ・ A-1 問題領域分析の十分性
 - ・ A-2 データ設計の十分性
 - ・ B-1 データセットの被覆性
 - ・ B-2 データセットの均一性
 - ・ B-3 データセットの妥当性
 - ・ C-1 機械学習モデルの正確性
 - ・ C-2 機械学習モデルの安定性
 - ・ D-1 プログラムの信頼性
 - * AIQM 内部品質特性の詳細は AIQM 本文 [10] を参照されたい。
- 開発要求事項
 - 開発プロセス要求
 - * ML モデル開発プロセスへの要求事項として整理する。
 - * 例えば下記のような要求である。
 - ・ ML モデル開発について ISO9001 などの QMS による品質管理
 - ・ トレーサビリティと一貫性の要求
 - ・ 市場環境のデータを使った車両レベルの妥当性確認の要求
 - 開発組織要求
 - * ML モデル開発に必要な開発組織の要求を記載する。
 - * 記載内容例)
 - ・ ML モデル開発に必要なスキル管理要求

8.6.3 ML 品質要求事項

本節では、品質保証のための要求事項を例示する。自動運転における ML 品質要求事項としては QA4AI のみでは足りないため、AIQM の品質要求事項をベースとして利用している。AIQM は分野横断的なガイドラインであるため、自動運転分野において解釈し適用する。

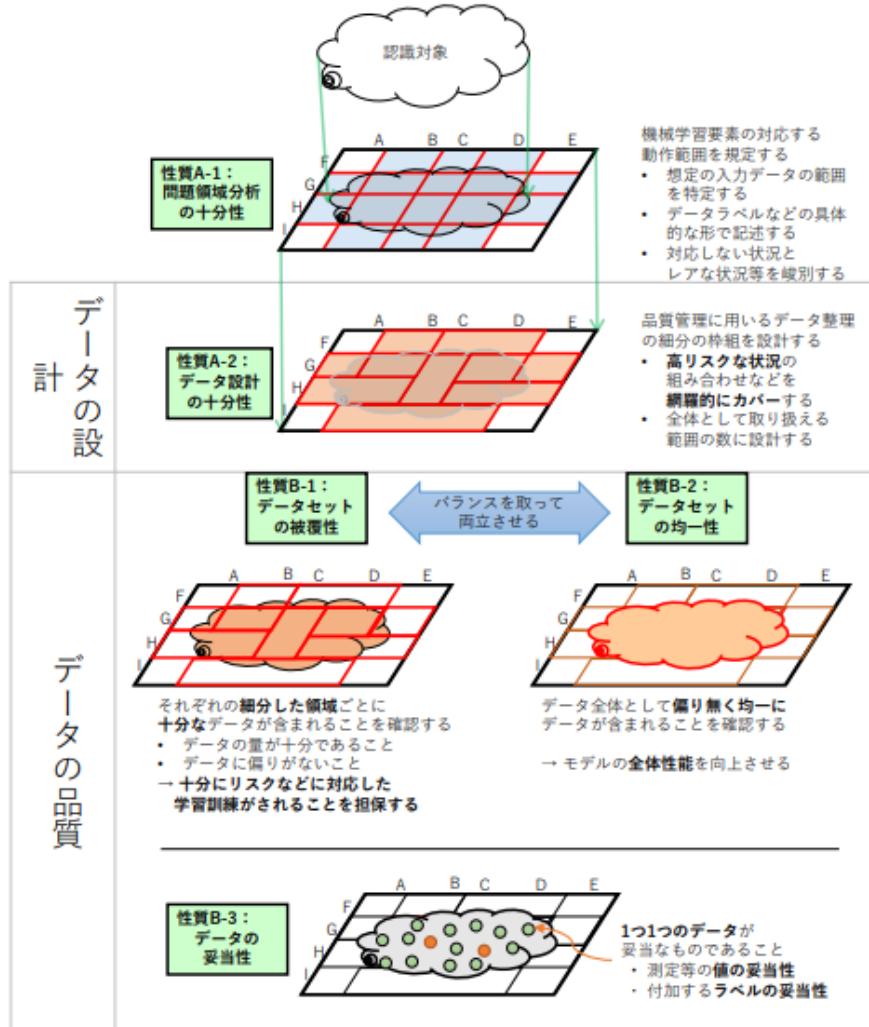


図 8.8 AIQM の内部品質特性（1）（引用元：機械学習品質マネジメントガイドライン [10]）

- AIQM では明示されていないため、下記について要求事項として明記した。
 - ML モデル開発プロセス要求事項
 - ML モデル開発組織要求事項
- ML モデルの不確実性(非線形や確率的な振る舞い)が許容範囲であるかの確認のため、「市場環境のデータを使った車両レベルの妥当性確認」を追加した。
- 内部品質特性」の要求レベル
 - AIQM は「外部品質特性」のレベルとして SIL に相当する AISL、AIPL、AIFL を決め、各外部品質特性レベルにあわせて内部品質特性の要求レベルを規定している。
 - しかし要求レベルが ML を利用した車載システムの開発において妥当であるかどうかは

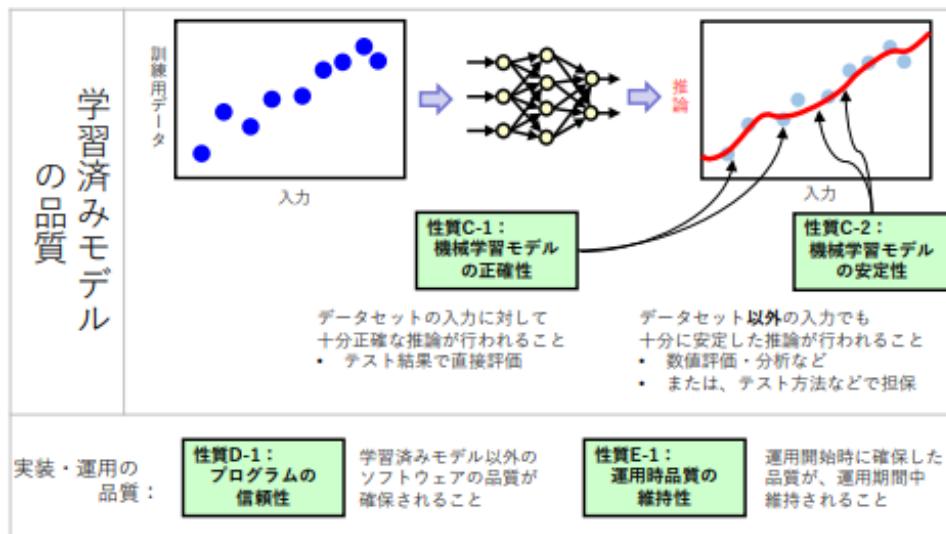


図 8.9 AIQM の内部品質特性（2）（引用元：機械学習品質マネジメントガイドライン [10]）

議論が十分になされていないため、今後の課題とし、業界で決めていく課題と認識している。

- 現状は AIQM のレベルを参考とし、各製品特性に応じて品質要求を決めていくこととする。
- その際、各品質要求の目的もあわせて記載するので、品質要求検討の際に参照されたい。

システム要求事項

カテゴリ	目的	品質要求事項	AIQM 内部品質特性	QA4AI
運用時モデル監視機能	運用時において、ML モデルの性能が対応できなくなる状況が発生することが予測され、その状況を監視するため。	運用時に ML モデルの性能を監視できる仕組みを持つこと	E-1 運用時性能の維持性	System Quality

カテゴリ	目的	品質要求事項	AIQM 内部品質特性	QA4AI
運用時モデル Update 機能	運用時において、ML モデルの性能が対応できなくなる状況が発生することが予測され、対応できるモデルでの運用を実現するため。	運用時に ML モデルを Update できる仕組みを持つこと	E-1 運用時性能の維持性	System Quality

ML モデル利用時品質要求事項

カテゴリ	目的	品質要求事項	AIQM 内部品質特性	QA4AI
車両レベルの妥当性確認	帰納的開発部品である ML モデルの不確実性(確率的振る舞い)が許容範囲内であることを車両レベルの妥当性確認にて確認すること	ML モデルの不確実性(確率的振る舞い)が許容範囲内であることを車両レベルの妥当性確認にて確認すること	A-1 問題領域分析の十分性	System Quality

ML モデル外部品質要求事項

カテゴリ	目的	品質要求事項	AIQM 内部品質特性	QA4AI
ML モデル性能評価	適切な評価指標により性能確認を実施するため。	(1)ML モデルの性能要件としての性能評価指標(精度、適合率、再現率、F 値、計算量など)およびその値を決め、その経緯と理由を記録すること	C-1 機械学習モデルの正確性	Process Agility / Model Robustness

カテゴリ	目的	品質要求事項	AIQM 内部品質特性	QA4AI
		(2)(1) の選定理由と選定内容、および指標値が有識者により妥当と判断されていること		

ML モデル内部品質要求事項

カテゴリ	目的	品質要求事項	AIQM 内部品質特性	QA4AI
要求分析	実世界の利用状況において、システム全体として価値を提供するため。	システム要件が具体化され、ML モデルへの要求が明確にされていること	A-1 問題領域分析の十分性	System Quality
データ設計	要求を満たすデータの範囲を規定するため。	必要なデータ属性の組み合わせが合理的な基準のもとで網羅的に設計できていること	A-2 データ設計の十分性	Data Integrity
データの十分性	規定されたデータ範囲において十分な量のデータを確保するため。	レアケースを含むさまざまなケースについて、十分な量のデータがデータセットに含まれていること	B-1 データセットの被覆性	Data Integrity
データのバランス	訓練した ML モデルが、求められる推論性能を満たすため。	各ケースに対応するデータ量とデータセット全体のデータ量のバランスが、学習の進行において適切であること	B-2 データセットの均一性	Data Integrity
データの妥当性	ML モデルの訓練やテストの目的に対して適切なデータを確保するため。	正確性や完全性などのデータ品質が担保されているとともに、付与されているラベルが適切であること	B-3 データセットの妥当性	Data Integrity
ML モデルの精度	学習データを入力した際の ML モデルのふるまいが、期待通りであるため。	学習が適切に行われていることが、交叉検証などの適切な指標を用いて示されていること	C-1 機械学習モデルの正確性	Model Robustness

カテゴリ	目的	品質要求事項	AIQM 内部品質特性	QA4AI
ML モデルの安定性	学習データ以外のデータを入力した際の ML モデルのふるまいが、許容範囲内であるため。	ML モデルが十分な汎化性能を持つことが、適切な指標を用いて示されていること	C-2 機械学習モデルの安定性	Model Robustness
過学習防止(適用技術)	<ul style="list-style-type: none"> ・過学習防止が適切に行われていることを確実にするため。 ・保守、運用時に再学習のインプットとするため。 	<p>(1) 過学習防止に用いた技術とその選択/非選択理由を記録すること</p> <p>(2)(1) 技術の適用方法、選択/非選択理由が有識者により妥当と判断されていること</p>	C-2 機械学習モデルの安定性	Model Robustness
過学習防止(判定基準)	汎化性能低下を防止するため。	過学習無きこと	C-2 機械学習モデルの安定性	Model Robustness
対ノイズ頑健性(適用技術)	<ul style="list-style-type: none"> ・対ノイズ頑健性確保が適切に行われていることを確実にするため。 ・保守、運用時に再学習へのインプットとするため。 	<p>(1) 対ノイズ頑健性確保に用いた技術とその選択/非選択理由を記録すること</p> <p>(2)(1) 技術の適用方法、選択/非選択理由が有識者により妥当と判断されていること</p>	C-2 機械学習モデルの安定性	Model Robustness

ML モデル開発プロセス要求事項

カテゴリ	目的	品質要求事項	AIQM 内部品質特性	QA4AI
品質マネジメント	従来と同等の品質保証を行うため。	ML モデル開発についても ISO9001 などの QMS による品質マネジメントを行うこと	D-1 プログラムの信頼性	System Quality
要件仕様化	帰納的開発に由来する非線形や確率的な振る舞いは、実使用環境において許容範囲内(妥当)であることの確認をもって ML モデルの品質が保証される。そのため、ML モデルの要件仕様化には、車両レベルの妥当性確認まで含めた繰り返しが必要である。	ML モデル開発において帰納的に決定する必要のある要件仕様については、その要件仕様化のために、分析、評価、検証、妥当性確認を繰り返して要件仕様を最適にすること ML モデルの妥当性確認については、車両レベルの妥当性確認まで実施すること	A-1 問題領域分析の十分性 A-2 データ設計の十分性	System Quality
トレーサビリティ	・ML モデル作成に関わる品質を説明するため。 ・カバレッジ、影響分析、要件実装状況の追跡等に役立てられる。	トレーサビリティを確立すること	B-3 データの妥当性	Process Agility
一貫性	参照先または作業成果物間の内容および意味に矛盾や重複がない事を確実にするため。	一貫性の確保を行うこと	B-3 データの妥当性	Process Agility

ML モデル開発組織要求事項

カテゴリ	目的	品質要求事項	AIQM 内部品質特性	QA4AI
スキル管理	ML モデル開発には、機械学習など従来のソフトウェア開発などとは異なるスキルや知識が必要となる。	ML モデル開発に必要となるスキル、知識を識別し、プロジェクトに対して必要なスキルおよび知識を持った人材を割り当てられるよう、スキル管理を行うこと	N/A	Process Agility

8.7 ML モデル開発プロセス

本節では、自動運転分野における ML モデル開発プロセスについて記載する。主に開発部門及び品質保証部門の方を想定読者として記載している。

機械学習 (ML) は、ソフトウェアとは開発方法やその特性が異なるため、ML 開発に特化した活動についての新たなプロセスの追加が必要となる。

ここでは ML モデル開発プロセスの概観を例示する。既存プロセスで対応できる活動については、ML 開発に特有の考慮事項を記載する。

以下に記載する ML モデル開発プロセス群は、各社の既存の開発プロセスに追加される形で定義、実装されることを想定している。各プロセスのアクティビティやタスクまでは記載できていないが、各社で適宜定義されたい。また、必要に応じて、プロセスを複数プロセスに分割してもよい。

8.7.1 ML モデル開発に必要な開発プロセス

- 新規追加プロセス
 - ML モデル開発エンジニアリングプロセス群
 - * MLE.1 ML モデル要件分析プロセス
 - * MLE.2 ML データセット構成プロセス
 - * MLE.3 ML モデル構築プロセス
 - * MLE.4 ML モデルテストプロセス
- 既存の Automotive SPICE ベースのプロセスを利用し、ML 開発に対して考慮が必要なプロセス
 - SUP.1 品質保証プロセス

- SUP.8 構成管理プロセス
- SUP.9 問題解決管理プロセス
- SUP.10 変更管理プロセス
- MAN.3 プロジェクト管理プロセス
- MAN.5 リスク管理プロセス
- SYS.2 システム要件分析プロセス
- SYS.3 システムアーキテクチャプロセス
- SYS.4 システム統合テストおよび統合テスト
- SYS.5 システム適格性確認テスト
- その他プロセス (ISO15288 など)
 - 妥当性確認プロセス

8.7.2 ML モデル開発プロセス群（例）

ML model Engineering process 群

- MLE.1 ML model requirements analysis process
 - 目的
 - * システム要件仕様及びシステムアーキテクチャ設計から ML モデルに対する要件を分析し、ML モデル要件を仕様化することである。
 - 成果
 - * システムの ML モデルエレメントに割り当てるべき ML モデル要件が定義されている。
 - ML モデル要件が正確性および技術的実現可能性、検証可能性について分析されている。
 - ML モデルの運用環境への影響が分析されている。
 - ML モデル要件を実装する優先順位が定義されている。
 - ML モデル要件が必要に応じて更新されている。
 - 一貫性および双方向トレーサビリティが、システム要件と ML モデル要件との間で確立されている。一貫性および双方向トレーサビリティが、システムアーキテクチャ設計と ML モデル要件との間で確立されている。
 - ML モデル要件が、コスト、スケジュール、および技術的な影響に対して評価されている。
 - ML モデル要件が合意され、影響を受けるすべての利害関係者に伝達されている。

* 備考1 検証可能性では、ML モデル要件が正しく実装されていることをテスト又

はレビューなどで確認可能かどうかを判断する。これは「MLE1.5 ML モデル要件に対する検証基準の導出」の入力となる。

- * 備考 2 ML モデル運用環境は、ML モデルを動作させる推論エンジンやプロセッサなどのハードウェアやソフトウェアなどである。

- 入力作業成果物

- * システム要件仕様書
- * システムアーキテクチャ設計書
- * HSI 仕様書
- * 利害関係者要求一覧表

- アクティビティ

- * MLE1.1 ML モデル要件の導出
- * MLE1.2 ML モデル要件の構造化
- * MLE1.3 ML モデル運用環境の影響分析と制約の識別
- * MLE1.4 ML モデル要件の分析と仕様化
- * MLE1.5 ML モデル要件に対する検証基準の導出
- * MLE1.6 トレーサビリティの対応付け
- * MLE1.7 作業成果物のレビュー
- * MLE1.8 ML 要件仕様書の承認と提出

- 出力作業成果物

- * ML モデル要件仕様書
- * ML モデル適格性テスト計画書
- * レビュー記録表
- * トレーサビリティ記録

● MLE.2 Dataset composition process

- 目的

- * ML モデル要件に適合した ML モデルを作成するためのデータセットを作成することである。

- 成果

- * ML モデル要件に適合した ML モデルを作成するためのデータセットが作成できている
- * 前記データセットの仕様が策定できている
- * 前記データセットの評価できている

- 入力作業成果物

- * ML モデル要件仕様書
- * ML モデル適格性テスト計画書

- アクティビティ
 - * MLE2.1 ML モデルを作成するためのデータセットの仕様策定
 - * MLE2.2 ML モデルを作成するためのデータセットの作成
 - * MLE2.3 ML モデルを作成するためのデータセットの評価
- 出力作業成果物
 - * データセット仕様書
 - * データセット
 - * データセット評価報告書
- MLE.3 ML model iterative process
 - 目的
 - * ML モデル要件に適合する ML モデルを作成することである。
 - 成果
 - * ML モデル要件に適合する ML モデル
 - 入力作業成果物
 - * データセット仕様書
 - * データセット
 - * データセット評価報告書
 - アクティビティ
 - * ML model design
 - base model selection
 - base model evaluation
 - model modification plan
 - * hyperparameter design
 - * ML model train
 - * ML model validation
 - * ML model test
 - * ML model evaluation
 - 出力作業成果物
 - * ML モデル
 - * ML モデル仕様書
 - * ML トレーニング記録
 - * レビュー記録表
 - * トレーサビリティ記録
- MLE.4 ML model testing process
 - 目的

* ML モデルが ML モデル要件を遵守していることをテストで確認することである。

- 成果

* ML モデルテストプロセスの実施に成功すると次の状態になる。

- ML モデルに対するテストを実施するために、開発計画およびリリース計画と整合性のある ML モデルテスト戦略が策定されている。
- ML モデルが ML モデル要件を遵守していることを確認するために、ML モデルテスト仕様が ML モデルテスト戦略に従って作成されている。
- ML モデルテスト仕様に含まれているテストケースが、ML モデルテスト戦略およびリリース計画に従って選択されている。
- 選択したテストケースを使用して、ML モデルに対するテストが実施され、ML モデルテスト結果が記録されている。
- 一貫性および双方向トレーサビリティが、ML モデル要件とテストケースを含む ML モデルテスト仕様との間、およびテストケースとテスト結果の間で確立されている。
- ML モデルテストの結果が要約され、影響を受ける全ての利害関係者へ伝達されている。

* 備考1 ML モデルテスト戦略は、ML モデルテスト計画書に記載する。

- 入力作業成果物

- * ML モデル要件仕様書
- * ML モデル適格性テスト計画書
- * ML モデル

- アクティビティ

- * MLE4.1 ML モデルテストの準備
- * MLE4.2 ML モデルテストのテストケース作成
- * MLE4.3 ML モデルテストの実施
- * MLE4.4 トレーサビリティの対応付け
- * MLE4.5 作業成果物のレビュー
- * MLE4.6 ML モデルテスト結果の承認と提出

- 出力作業成果物

- * ML モデル適格性テスト仕様書兼報告書

8.7.3 ML モデルの Process Flow Diagram

- ML モデルを作成するための入出力及びプロセスの流れを下図に示す。

Process Flow diagram

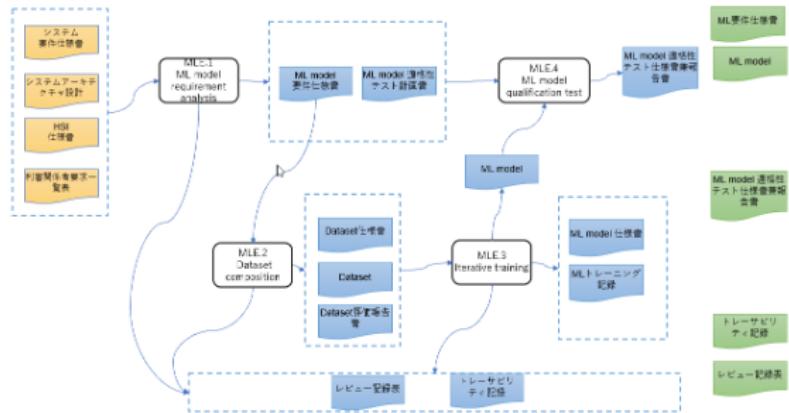


図 8.10 ML モデルの Process Flow Diagram

8.7.4 ML モデル開発のトレーサビリティと一貫性

- トレーサビリティと一貫性の要求
 - ML モデル開発に必要なトレーサビリティと一貫性について説明する。
 - トレーサビリティ管理項目
 - トレーサビリティ管理項目の例を記載する。
 - トレーサビリティ関係図
 - トレーサビリティ管理項目間の関係を表すマトリックスを作成する。

8.7.5 関連プロセスへの考慮事項

- Automotive SPICE や ISO15288（プロセスとライフサイクルステージ）をベースとした開発プロセスなど、既存の関連プロセスへの考慮事項項目の例を記載する。

SUP.1 品質保証プロセス SUP.8 構成管理プロセス

- 構成品目例
 - 構成管理の注意点
 - ベースライン

SUP.9 問題解決管理 SUP.10 変更依頼管理 MAN.3 プロジェクト管理

- スキル管理
 - 備考) ML モデル開発についても必要なスキル、知識、および経験の識別が必要である。
 - 運用保守

- 備考) 性能低下の監視と保守について計画段階で考慮すること。

MAN.5 リスク管理 ACQ.4 サプライヤ監視

- DIA
- サプライヤ選定

SYS.1 要件抽出

< Automotive SPICE 以外の関連プロセス > Validation : 妥当性確認プロセス (ISO15288 など)

8.8 ML 関連開発と品質保証

8.8.1 自動運転開発における ML 関連開発活動と品質確認観点

- ここでは自動運転開発における ML 関連開発活動と対応する品質確認観点の概要を説明する。
- 概観をつかむために、車載 ML 搭載プロダクト開発における ML 関連の開発活動とそれに対する品質確認観点を図 8.11 に示す。
- 品質確認観点は AIQM の内部品質特性を用いて記載する。

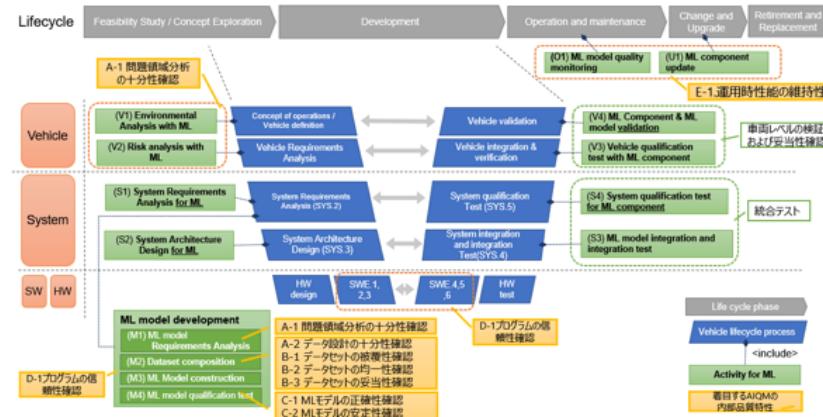


図 8.11 車載 AI 搭載プロダクト開発における ML 開発活動と AIQM 内部品質特性

<凡例>

- 「Life cycle phase」(グレー) は、車載システムのライフサイクルフェーズ
- 「Vehicle lifecycle process」(青色) は、車載システム開発ライフサイクルプロセス
- 「Activity for ML」(黄緑色) は、ML 開発に関する活動
- 「着目する AIQM の内部品質特性」(オレンジ) は、AIQM の内部品質特性

<図の説明>

- ML モデル開発は、(M1)～(M4) の「ML model development」である。
- 「ML モデル開発」はプロセスとして定義する必要がある。
- 新規に定義する場合は 8.7 章の「ML モデル開発プロセス」も参考にされたい。
- (V1)～(V4)、(S1)～(S4) の ML 関連開発は、既存の開発活動の一部として実施される。
- (V1) と (V4)、(V2) と (V3)、(S1) と (S4)、(S2) と (S3) は、ML 関連開発についても、それぞれ V 字の左右で対向する活動である。
- (M1)～(M4) の ML モデル開発は、システム要件である ML モデルを作成する活動としてシステム要件分析の一部として紐づけられる。詳細は「ML モデルの既存開発プロセス上での扱い」節を参照。
- 開発フェーズだけではなく、運用時の「(O1)ML モデル性能監視」や「(U1)ML component Update」の活動もある。
- (M1)～(M4) の「ML model development」に AIQM の内部特性の「A-1」～「D-1」の全てが集まっている。
- ML モデルの完成は (Vx)、(Sx)、(Mx) の全てが完成されている必要がある。そのためウォーターフォール的な開発は馴染まず、先行開発の初期段階から「十分な環境分析とともにした要件分析と実装、車両レベルの妥当性確認」までを何度も繰り返しながら品質を高めていくという反復開発 (Iterative and Incremental 開発) が必要となる。
- 最終的な妥当性確認は、量産車両や量産試作車両での評価やテストまで行うことはできないが、その時点での手戻りは影響が大きい。リスク低減のために PoC や先行開発初期の時点から、ユースケースの網羅的な洗い出しを行い、実車やそれに近い環境での妥当性確認 (評価やテスト) を行う必要がある。その際撮影された画像を用いると、ML コンポーネントレベルにおける仮想テスト環境での妥当性確認が可能となる。
- 従来の開発にもまして、ML 開発の場合はより一層、初期開発段階における妥当性確認を含めた実現性確認、ML 要件の洗い出しが重要となる。

ML 関連活動と品質確認観点を図 8.12 に示す。開発関連アクティビティ列の①～⑦は AIQM 記載の対応する活動である。品質確認観点列の A-1～D-1 は、AIQM の対応する内部品質特性である。記号表記の無いものは、独自に追記した。

8.8.2 ML コンポーネントの品質保証

本節では、自動運転における ML コンポーネントの品質保証について説明する。主に開発部門の方を想定読者として記載している。

ML コンポーネントは入力データに対して、訓練済みモデルを使って推論を実行し、その結果を出

ML関連開発活動と品質確認観点

ID	対応プロセス	主なML関連開発アクティビティ	品質確認観点
V1	Concept of Operation / Vehicle definition	①-1 安全性機能の事前検討：適用規格の確認	A-1. 問題領域分析の十分性
V2	Vehicle Requirement Analysis	①-2 システムの機能要件（目的・目標）の特定 ①-3 システム利用のリスクナリオ検討（リスクアセスメント） ①-4 システム全体の利用動品質特性の要求の検討	A-1. 問題領域分析の十分性
S2	System requirement analysis	システムに割り当てられた要求分析と要件仕様化	A-1. 問題領域分析の十分性
S3	System architecture design	②-5 システム構成要素の設計と機能要件の割り当て	安全設計の実施
M1	ML model requirement analysis	②-6 AI外部品質の達成要求レベル特定 ②-7 AI内部品質の達成要求レベル特定 ② AI要件分析	A-1. 問題領域分析の十分性 A-2. 問題に対する被覆性
M2	ML dataset composition	③-1 データセット設計、調整 ③-2 データ準備	B-1. データセットの被覆性 B-2. データセットの均一性 B-3. データセットの妥当性
M3	ML model design and construction	④ モデル設計、訓練、実装最適化	D-1. モデルの健全性
M4	ML model qualification test	⑤ AIモデルの品質確認・検証	C-1. AIモデルの正確性 C-2. AIモデルの安定性
S4	System integration & integration test	⑥ MLモデルを含めたシステム統合テスト	安全設計の検証
S5	System qualification test	MLに関するシステム要件適合性確認テスト	MLコンポーネント要件遵守
V3	Vehicle integration & verification	⑦ 車両レベルの検証	MLモデルを搭載したシステム評価
V4	Vehicle Validation	⑧ 車両レベルの妥当性確認	MLモデルの妥当性確認

図 8.12 ML 関連開発活動と品質確認観点

力するコンポーネントである。例えば、画像データを入力し、物体認識の推論結果を出力する。8.7.3 章の ML モデルの Process Flow Diagram で作成される ML model structure と weight parameter で、ML コンポーネントを Inference Engine として搭載したもの。

ML コンポーネントの構成

ML component

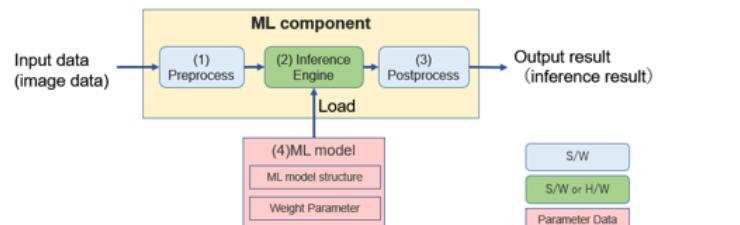


図 8.13 ML コンポーネントの構成

ML コンポーネントの構成は下記である。

- (1) 前処理部（SW）
 - 入力データを推論エンジンに入力する。（推論エンジンへのデータフォーマット変換を含む）
 - ソフトウェア I/F として実装される。

- (2) 推論エンジン (SW or HW)
 - ML モデル (ML モデル構造、重みパラメータ) に従い、入力データに対して推論演算を行う。
 - 例えば画像データを入力し、物体認識の推論結果を出力する。
- (3) 後処理部 (SW)
 - 推論結果を ML コンポーネントの外部 IF として出力する。
 - ソフトウェア I/F として実装される。
- (4) ML モデル (パラメータ)
 - 「ML モデル構造」と「重みパラメータ」から構成される。
 - Deep Learning では「ML モデル構造」は「DNN モデル構造」である。

ML コンポーネントの品質保証の考え方

- ML コンポーネントは、従来のハードウェア/ソフトウェアで実装される (1)～(3) までと、機械学習固有の特性を持つ (4)ML モデルが統合されたコンポーネントである。ここでは統合後のシステムとしての品質保証について述べる。まず、ML コンポーネントを「(4)ML モデル」と「それ以外 ((1) 前処理部、(2) 推論エンジン部、(3) 後処理部)」に分けて考える。
- ML モデル自体の妥当性確認は、実使用環境での製品レベル（市場環境のデータを使った車両レベルの妥当性確認）で行われる。最終的な妥当性確認が量産車両や量産試作車両での評価やテストまで行うことはできないが、その時点での手戻りは影響が大きい。そのため、先行開発初期の段階から車両レベルでの妥当性確認を進めることを強く推奨する。
- 実開発においては、ML コンポーネントレベルの仮想テスト環境で妥当性確認を実施し、リスク軽減を図る。

備考

- 「(2) 推論エンジン」は、ML モデルの指定に従い演算処理を実行するだけの SW または HW モジュールである。そのため、従来通りの演繹的な開発とテスト (ユニットテスト、統合テスト) を行う。推論エンジンを外部調達する場合は、品質保証レベルの確認や受け入れテストを実施する。
- 「ML コンポーネント、(1) 前処理、(2) 推論エンジン、(3) 後処理」は、QA4AI ガイドラインや AIQM ガイドラインで用語やそのスコープが統一されていないため、本稿では上記の定義で統一する。ただし、将来の標準化動向に合わせて改変も検討する。

8.8.3 安全性要求への対応アプローチ

システムとしての安全設計

- Deep Learning をはじめとして ML 技術を用いることにより高い認識性能を達成することができる。しかし、安全性を高めるためには、認識結果が正しくない場合がありうることを想定した設計にすることが重要である。
- 一例として、ISO/TR 4504:2020 Road Vehicle - Safety and cybersecurity for automated driving systems - Design, verification and validation や SEAMS プロジェクト、SaFAD などで ML コンポーネントの外に安全機能（フェールセーフ機能）を追加して、システムとして安全性を確保するアプローチが提案されている。
- 本 WG でも、ML コンポーネントの外に安全装置を配置し、システムとして安全性を確保するというアプローチを推奨する。
- 外部に安全機構を持たせた場合に想定される活動を以下に示す。
 - ISO26262 Part 4 システムレベルの製品開発
 - * 安全設計
 - Safety envelope などの AI 安全設計
 - パーテショニングによる安全設計：AI(非安全系) から安全系への不具合伝播を防ぐ
 - 安全設計の検証
 - * フォルトインジェクションテストによる安全設計の検証
 - AI の誤り（誤判定など）についての安全性を検証する。
 - 安全メカニズムが正しく動作するかを、構成要素のさまざまな故障を発生させることにより安全性を確認する。

SOTIF(ISO 21448) との対応関係

ここでは自動運転開発での AI 開発フローと SOTIF の対応関係を図 8.15 に示す。AI 開発活動のうち、以下の対応関係がある。

- AI モデル品質確認、システム統合テスト、車両レベル検証が「SOTIF 検証」に該当する。
- 車両レベル妥当性確認が「SOTIF 妥当性検証」に該当する。
- リスクアセスメントが「SOTIF 関連ハザード特定 & リスク評価」に対応する。

SOTIF の反復的な活動は、AI の反復的な開発の中で実施することができる。

Safety Architecture Concept

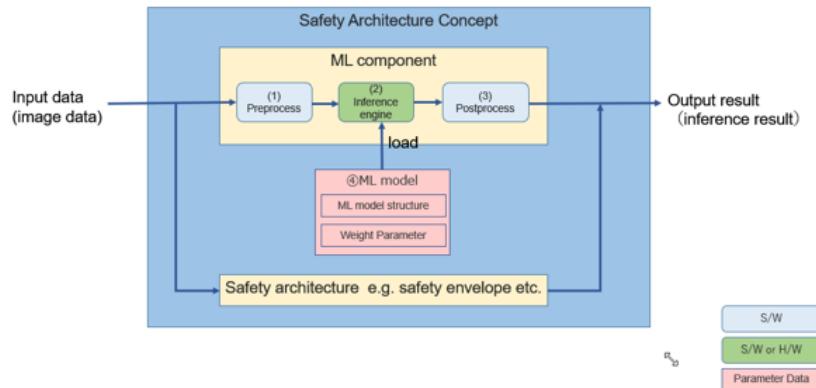


図 8.14 ML コンポーネントにおける安全装置の配置

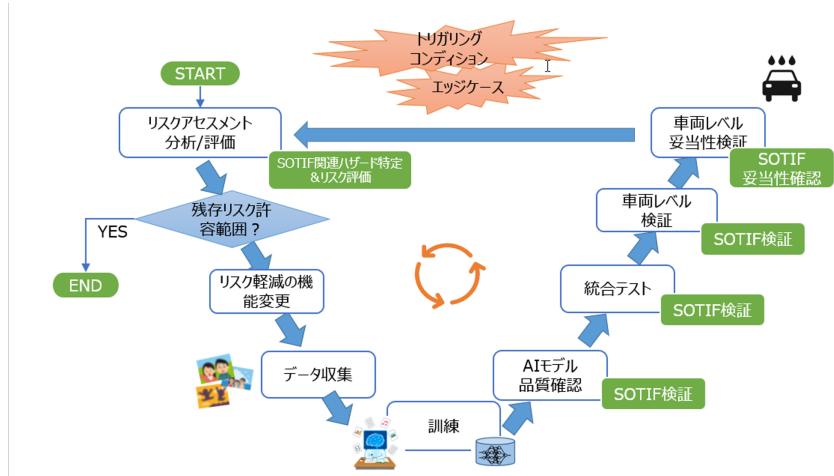


図 8.15 自動運転開発での AI 開発フローと SOTIF の対応関係

8.9 付録

8.9.1 自動運転関連標準

ISO 21448 (SOTIF: Safety of the Intended Functionality)

ISO26262 で規定されている車載 E/E システムの安全性はランダムハードウェア故障やシステムチック故障すなわち仕様からの逸脱に対する安全性（機能安全）がスコープであった。しかし、先進安全システム (ADAS) や自動運転システム (ADS) においては、機能安全を担保するだけでは必ずしも安全とは言えず、外界環境を認識し、判断、制御を行うシステムの機能の不足（仕様の不十分さ

や性能限界) や人間による合理的に予測可能なミスユースについて考慮する必要がある。センサ等の性能限界、機能不足、ミスユース、仕様の不十分さの安全性をスコープとしているのが SOTIF である。

ハザードを引き起こす条件がないか（許容できるレベルであるか）をシナリオ（すなわちシステムと外界のインタラクション）ベースで安全論証するのが SOTIF の基本的な考え方である。シナリオは未知と既知、ハザードがあるかないか、の 2 軸 4 象限で区分される。未知または既知のハザードが及ぶシナリオを反復的プロセスによって特定し、妥当性確認によって既知かつハザードがないシナリオに変えることで安全を担保する。シナリオに内在し、ハザードを引き起こす可能性がある条件をトリガリング条件 (triggering conditions) と呼ぶ。機械学習モデルのレイヤーで見たとき、機械学習モデルが認識対象でないものを認識対象と間違える（偽陽性）または認識対象を見逃してしまう（偽陰性）ような判定結果を誘発するシーンがトリガリング条件と考えられる。トリガリング条件ごとにリスクを細分化し、対策要否を検討し、対策を打つまでを反復的に繰り返すことでの安全性を高める。

ISO TR 4804 (Road vehicles – Safety and cybersecurity for automated driving systems – Design, verification and validation)

Aptiv、アウディ、バイドゥ、BMW、コンチネンタル、ダイムラー、フィアット・クライスラー・オートモビルズ、HERE、Infineon、インテル、フォルクスワーゲンの 11 社によるコンソーシアムが 2019 年 6 月に公開した、安全を考慮した自動運転システム (ADS) を開発するための技術や検討事項の指針をまとめた白書を基に作成されたのが ISO TR 4804 である。ANNEX B には機械学習ベースの画像認識技術を用いた自動運転の認識システムを開発するときのプロセス、成果物、技術課題などが記載されており、本 AI 品質保証ガイドラインの自動運転パートの参考先の一つとなっている。本白書では「ADS の最終的なステートメント、指針、あるいは標準を意図したものではない」と述べられているものの、ADS の基本コンセプトをシステムレベルから機械学習レベルまで包括的かつ具体的に記載した文書は他になく、ADS 開発の羅針盤的文書とみなしてもよいだろう。

ISO PAS 8800 (Road Vehicles – Safety and artificial intelligence)

ISO PAS 8800 は、道路運送車両における AI の不十分な性能や誤作動の挙動に影響を与える安全関連の特性やリスク要因を定義する。また、開発・展開のライフサイクルの全段階に対応するフレームワークを記述している。これには、機能に関する適切な安全要件の導出、データの品質と完全性に関する考察、故障の制御と緩和のためのアーキテクチャ対策、AI をサポートするために使用するツール、検証および妥当性確認技術、ならびにシステム全体の安全性を保証するための論拠として必要な証拠が含まれている。2023~24 年頃の PAS (公開仕様書) 発行を目指し、2023 年 3 月現在作成中である。

ANSI/UL 4600 Standard for Safety for the Evaluation of Autonomous Products

ANSI/UL 4600 は 2020 年に発行された自律型製品の安全性に関する規格である。自動運転システムに対する規格ではないが、アネックスにおいて、自動車の機能安全規格である ISO 26262 や ISO 21448 (SOTIF) へのマッピングが示されており、自動運転車の安全性保証に利用されることが期待されている。本規格の大きな特徴の一つは、安全ケースを安全保証のフレームワークとして大々的に採用している点である。安全ケースは、安全が高度に要求されるシステムに関する規格で採用されている、安全に関する論証とその根拠資料を構造的に記述するものである。本規格では、安全ケースをより厳密に適用するためのガイドラインが規定されている。特徴の一つには、安全をアセスするための基準として安全遂行インディケーターがあり、安全に関連する様々な項目について規定することが要求されている。

8.9.2 画像認識 AI プロダクトのテスト技術

一般に、機械学習によって作成したモデルの評価は、学習データセットを拠り所としたテストにより、正解率、再現率、適合率などの指標を使って行う。この時、学習データセットを訓練データセットと試験データセットに分け、訓練データセットを使って学習させたモデルを、試験データセットを使ってテストすることで、学習時にはない未知の入力データに対する出力データが適切かどうかが評価できる。しかし、自動運転の画像認識機能には多種多様な未知の入力データへの対応が求められるのに対し、学習データセットがカバーする範囲は限られており、学習データセットに基づく評価だけでは十分とは言えない。自動運転に求められる水準のテストを行うためには、これらとは異なるアプローチが必要である。この観点から、以下では説明性・解釈性からのアプローチ、メタモルフィック・テスティング (Metamorphic Testing) の応用、敵対的サンプル (Adversarial Examples) に対する頑健性評価について述べる。

帰納的な開発プロセス、BlackBox 性、CACE (Changing Anything Changes Everything) 性といった特徴から AI プロダクトの品質保証のためのテスト技術は確立しておらず、ここでは多様なテスト技術から一部を紹介する [12]。「技術カタログ」とあわせて参照していただきたい。

説明性・解釈性からのアプローチ

自動運転のための画像認識の領域は、近年の AI プロダクトの発展をけん引してきた領域であり、それらに対する説明性・解釈性についての研究も多く発表されている。例えば、「機械学習における説明可能性・解釈性」に記載の、CAM、LIME、TCAV といった技術は、論文内でも画像認識を題材に、有効性を検証している。これらを用いることで、モデルおよび学習データに意図しないバイアスがないかどうかを検討するための情報を抽出することができる。例えば、LIME の論文内では、狼とハスキー犬を識別するタスクのデータセットに、意図的なバイアス (狼のデータは、背景を必ず雪にする)を入れ、これを抽出可能であることを示している。

これら技術により検証可能になる側面がある一方、いくつかの課題がある。検証可能になる側面として、作成したモデルの判断根拠が"人間の判断根拠と沿うかどうか"という観点での妥当性確認がある。抽出した説明/解釈情報を人間が確認することで、明らかに人間の判断根拠と異なる情報から判断をしているなどの点をチェックすることができる。一方、課題として、その良し悪しの判定基準をどう決めるか、という重要なポイントがある。これに言及している論文は少ない。上述の LIME の論文内評価では、複数人の被験者を用いてバイアスを含んだモデルであるかどうかに気がついたかどうか、により手法を評価しているが、システムの検証項目としての成立性については言及していない。そもそも、検証は要件に対してなされることを考慮すると、説明したような定性的な観点を、どのように要件定義するのか、ということの裏返しの課題になっていると考えられる。

説明性・解釈性技術の今後の発展により、モデルに対するホワイトボックス的な評価の糸口が見つかっていく可能性はあるものの、現段階では研究段階であると言える。

メタモルフィック・テスティングの応用

一般的なソフトウェア開発では、テストはソフトウェアが仕様を満たすことを確認するために行われる。このため、仕様に基づいて設定したテストオラクルを使って、入力データを与えたときのソフトウェアの出力データがテストオラクル通りであればテスト合格と判定する。これに対し、機械学習を使って作成したモデルには仕様が存在せず、テストオラクルを設定することができない。前述のように、学習データセットがテストオラクルの役割を担うが、適切なデータを収集して正解ラベルを付与しなければならないので、容易には増やせない。

メタモルフィック・テスティングは、入出力データの対応をデータ変換によって拡張し、より広範囲のテストを行うテスト手法である。入力データに一定の変換を加えてモデルに入力したとき、出力データに何らかの一定の変換を加えたデータがモデルから出力されることが期待できるならば、これらの変換を使ってテストデータを拡張する。例えば、画像から歩行者を認識するモデルのテストにおいて、試験データセットの画像に霧状の靄をかけることで、試験データセットを霧の日に拡張する。この時、出力データである認識結果が変わらないことが期待できるので、入力データには靄をかける変換を加え、出力データはそのまま（恒等変換）にしたテストデータセットを作成する。この場合、入出力データ間の変換関係が正確には定められない（例えば靄が強い場合には認識結果が一致しなくても許容される）ので厳密にはメタモルフィック・テスティングとは言えないが、テストに有用な範囲で近似的な変換関係を設定することでメタモルフィック・テスティングを応用できる。

自動運転向けの画像認識モデルのテストにメタモルフィック・テスティングを応用した例として、DeepTest[3] があげられる。DeepTest は、ニューロンカバレッジ（入力データによって活性化されたニューロンの割合）を指標として、深層ニューラルネットワークモデルを系統的にテストするツールである。入力がカメラ画像、出力がステアリング角度であるモデルのテストでは、学習データセットから画像変換により多様な道路状況の画像を合成し、変換前後の画像についてモデルが output するステアリング角度が大きく変化しないことをテストする。平行移動、拡大／縮小、水平方向の変形、

回転、コントラストの調整、輝度の調整、鬻かけなどの画像変換によりテストデータセットを拡張し、それらを入力したときに活性化されるニューロンの割合でテストの十分性を判断する。このようなしくみにより、DeepTest は多数のモデル誤作動を低い誤検知率で検出できたという。

DeepTest の画像変換は、変換によって合成された画像に不自然さがある。これに対し DeepRoad[5] は、機械学習を使った画像生成技術を入力データ変換に用いて、現実世界に近いテストデータを使ったテストを行う。学習データセットの画像と YouTube から抽出した雪および雨の画像を GAN (Generative Adversarial Networks) を用いて合成することで、学習データとしては収集し難い大雪の日と大雨の日のテストデータセットを得る。テストの結果、ステアリング角度が晴天時とは大きく変わる不適切なモデルのふるまいが数多く検出された、と報告されている。

敵対的サンプルに対する頑健性

深層学習を使って作成したモデルでは、入力データの微小なずれに対し、出力データの差異が想定外に大きいことがある。例えば道路交通標識を認識するモデルは、人間には意識されない程度のわずかな改変を速度制限標識に加えるだけで、制限速度を誤認識する場合がある。他にも、メタモルフィック・テスティングを使った LiDAR の障害物認識モデルの頑健性テストの結果が報告されている [6]。これによれば、LiDAR データ点群の、車が進行可能な領域の外にランダムに点を付加するだけで、モデルは進行可能な領域内にある自動車、歩行者、サイクリスト、その他の障害物を見失い、その割合は無視できない程度であるという。このような誤認識を引き起こす入力データは、敵対的サンプルと呼ばれる。モデルの品質において、敵対的サンプルに対する頑健性は重要な項目である。

敵対的サンプルに対する頑健性の評価では、与えられた入力データの微小なずれに対してモデルの出力データが安定していることをテストし評価する。基本的には、入力データについて出力データが変化しない近傍が十分大きいことを示せば良いが、近傍内の全ての入力データについて出力データが同一であることをテストするのは難しい。現実的な計算量で近傍を推定するために様々なレベルのテスト手法が提案されており、そのいくつかを下記に示す。

- CLEVER[4]

出力データ間の差を比較し、出力データが同一になる入力データの範囲を極値理論により推定する。比較的計算量が少なく実用規模のニューラルネットワークに対応できる半面、近傍内に敵対的データが存在しないことは保証できない。

- CNN-Cert[1]

非線形の活性化関数の上下限値を線形関数で近似し、出力データが同一である境界を求める。得られる近傍は近似であるが、その中に敵対的データが存在しないことは言え、計算量も少なく小規模のニューラルネットワークに対応できる。

- Reluplex[2]

ニューラルネットワークのふるまいを論理式で近似し、近傍内で出力データが変わらないこ

とを網羅的に検証する。敵対的データが存在しない近傍を正確に特定できる半面、計算が複雑であり、ごく小規模のニューラルネットワークにしか対応できず画像は処理できない。

参考文献

- [1] Akhilan Boopathy et al. “CNN-Cert: An Efficient Framework for Certifying Robustness of Convolutional Neural Networks”. In: *33rd AAAI Conference on Artificial Intelligence (AAAI-19)*. AAAI-19. 2019, pp. 3240–3247.
- [2] Gui Katz et al. “Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks”. In: *Computer Aided Verification*. CAV 2017. Springer International Publishing, 2017, pp. 97–117.
- [3] Yuchi Tian et al. “DeepTest: Automated testing of deep-neural-network-driven autonomous cars”. In: *Proceeding of the 40th International Conference on Software Engineering (ICSE '18)*. ICSE 2018. 2018, pp. 303–314.
- [4] Tsui-Wei Weng et al. “Evaluating the Robustness of Neural Networks: An Extreme Value Theory Approach”. In: *Sixth International Conference on Learning Representations (ICLR 2018)*. ICLR 2018. 2018.
- [5] Mengshi Zhang et al. “DeepRoad: GAN-based Metamorphic Testing and Input Validation Framework for Autonomous Driving Systems”. In: *33rd IEEE/ACM International Conference on Automated Software Engineering (ASE)*. ASE 2018. 2018, pp. 132–142.
- [6] Zhi Quan Zhou and Liqun Sun. “Metamorphic Testing of Driver-less Cars”. In: *Communications of the ACM* 62.3 (2019), pp. 61–67.
- [7] デンソー. *AD&ADAS システム 論理アーキテクチャ*. 2022.
- [8] 日本規格協会. *JIS X 25000: ソフトウェア製品の品質要求及び評価 (SQuaRE)-SQuaRE の指針*. 2017.
- [9] 日本規格協会. *JIS X 25030: ソフトウェア製品の品質要求及び評価 (SQuaRE)-品質要求事項の枠組み*. 2021.
- [10] 機械学習品質マネジメントガイドライン. Tech. rep. 国立研究開発法人産業技術総合研究所, 2022. URL: <https://www.digiarc.aist.go.jp/publication/aiqm/AIQuality-requirements-rev3.2.1.0079-signed.pdf> (visited on 01/20/2023).
- [11] 中江 俊博, 桑島 洋. “自動車業界における AI セーフティ動向”. In: *人工知能* 38.2 (2023), pp. 210–220.
- [12] 中島 震. “機械学習ソフトウェア・テスティングの技術動向”. In: *信学技報* 119.361 (2020), pp. 61–66.

9. AI-OCR

9.1 本章の背景と目的

本章では AI-OCR における品質保証を行うための考え方と現時点で考えられる実現方法について、 QA4AI ガイドラインを上位ドキュメントとして、ワーキンググループで検討した結果を記載する。

まず、AI-OCR の定義は、光学文字認識（画像から文字を認識し、文字毎に文字コードへ変換する技術）のうち、機械学習が利用されているものと定義する。

OCR の技術は古くから存在し、文字や記号が記載された画像を読み込み、必要な識別パターンを含むテンプレートと比較することでコンピュータが識別可能な文字コードへ分類を行う技法により画像から文字コード変換を行うことが可能な技術だ。しかし、従来の技術では手書き文字をはじめ、フォントの違いや文字位置の座標ずれ等、テンプレートと異なる文字画像が入力された際には、文字認識の精度（文字を正しく文字コードへ分類される割合）が非常に低く、その利用範囲は特定のフォーマットに対応した文字画像に対して、特定のフォント・文字列を読み込めるもののみに限定されていた。

そのような背景の中、機械学習技術の発展に伴い、この技術が OCR へも適用され、AI-OCR として、文字認識の精度は飛躍的に向上した。その結果、AI-OCR の技術を各企業および団体のビジネスプロセスに導入する企業が増加し始めているのが現在の状況である。その一方で、AI-OCR を導入するにあたり、どのように品質保証を行うかといった考え方を整理されていなかった。

そこで、本章では AI-OCR システムを開発・導入するにあたり、AI-OCR における品質保証の論点および、特有の課題について整理し、有効な品質保証へのアプローチを提案する。

なお、本章の AI-OCR の品質保証への考え方については、大まかな指針や注意点についてのみ記載するが、全て準拠する必要があるものでなく、導入対象の業務や各社の基準で導入範囲を決めるために利用されることを想定している。また、本ガイドラインは帳票に記載された任意の項目（例えば、請求書における合計金額など）を抽出する帳票型 OCR と、記載された文字画像を全て書き起こす文書型 OCR のうち、主に帳票型 OCR に対しての見解を述べるが、文書型 OCR にも適用可能な情報を含むように努めた。

本章の大まかな流れとしては、9.2 節にて、ガイドラインとして取り扱う前提となる AI-OCR システムの概要について定義する。

9.3 節では、9.2 節で定義したシステムにおける特有の課題および有効とされる技術について記載する。この章では、AI 開発・導入およびシステム開発・導入における一般的な課題ではなく、AI-OCR の開発・導入時ならではの課題に対しての議論へ焦点を当てる。

9.4 節では 9.3 節にて提示した課題を解決していく上で、どのような品質保証を行うべきかを実例

を持って記載する。

そして、9.5 節では AI-OCR システムの開発および導入を行う際の品質保証レベルを提案する。

9.2 前提となるシステム構成

本章で議論する AI-OCR システムは表 9.1 に示す 4 つのモジュールから構成されることを想定している。

表 9.1 想定する AI-OCR のモジュール構成

#	モジュール	概要
1	前処理モジュール	入力された画像ファイルに対して、ノイズの除去や傾きの補正を行うモジュール
2	文字位置検出モジュール	画像データのうち、OCR 対象とする文字の座標を識別するモジュール
3	OCR モジュール	文字位置検出モジュールが認識した座標情報の画像から文字コードに変換するモジュール
4	項目抽出モジュール	OCR した結果を特定の項目として抽出するモジュール

なお、入力および出力におけるインターフェースの仕様については、本章では任意として扱うが、入力に関しては、入力する画像の拡張子や、ファイルの連携方法（オンライン型/バッチ型）の検討が論点となり、出力ではアウトプットの形式や周辺システムへの連携の考慮が主な論点となる。

これらインターフェース仕様に関しては、各システムの仕様や、導入する業務によって調整されるべき観点であり、システムとしての品質保証をする上では重要なテスト観点となる。その一方で、AI-OCR としての品質保証という観点から考慮すると、下図のような入力から出力までの流れを本章の検討範囲とする。

また、次項より下図で提示したモジュールの詳細について記載する。

図 9.1 本章の前提となるシステムと検討範囲



9.2.1 前処理モジュール

前処理モジュールは入力された画像ファイルを認識し、陰影の除去やノイズの認識、傾きの補正等、OCR を実施する前の画像の加工処理工程を担っているものと定義する。

このモジュールでは画像認識技術が利用されることが多く、ベクトル化された画像から陰影を認識し、傾きを検出し補正を行う。傾き補正の例としては、フーリエ変換により、画像の軸を取得し、傾きを特定して補正をかける方法などがある。

9.2.2 文字位置検出モジュール

文字位置検出モジュールは入力された画像ファイルから文字ないし文字列が存在する場所を特定する。このモジュールでは、2つのアプローチが存在する。1つ目は、画像認識技術を用いて、画像内で文字と認識した箇所の座標を取得する方法（機械学習型アプローチ）がある。2つ目はあらかじめルールベースで座標を定義する方法（ルールベース型アプローチ）がある。

いずれの方法でも文字位置検出モジュールとしての役割は果たすことができるが、AI-OCR の導入対象によってどちらが有効かは異なってくる。例えば、認識対象の画像が定型レイアウトに限定される場合はルールベースで座標を定義した方がコストおよび導入へのスピードにメリットがある。その一方で、レイアウトの変更等が発生すると、座標定義のやり直しが発生するリスクがある。また、前処理モジュールの機能の想定外動作に起因し、画像内に歪みやノイズが残存すると、定義した座標がずれる場合がある。

その一方で、機械学習型のアプローチで文字識別モジュールを構築する際には、認識対象の帳票が多様なレイアウトであっても、様々な帳票のレイアウトの学習を行うことで、座標定義の指定および変更なく対応することができるメリットがある。その一方で、汎用的でない業務要件を取り組む際には、導入前にモデルの学習のプロセスが発生する場合があるため、導入までの期間はルールベースと比較して長期化する傾向にある。

いずれのアプローチにおいても実際の適用業務における認識対象画像の種類・数・レイアウトの定期的な変更有無等を事前に調査することで、最適なアプローチを選択することがシステム全体の品質（System Quality）を高めていく上で重要な指針となる。

9.2.3 OCR モジュール

OCR モジュールは文字位置検出モジュールから渡された画像データの座標をピクセル毎に解析し、該当する文字コードへ分類し、テキストデータ化を行うモジュールである。

OCR モジュールでは基本的に機械学習のアプローチを取られているものとして、本章では扱う。

9.2.4 項目抽出モジュール

項目抽出モジュールは、OCR によりテキストデータ化された情報から、必要項目を抽出するモジュールである。例えば、レシートから会計金額のみを抜き出したい場合、OCR 結果から合計金額のみをアウトプット形式に従い出力する処理のことを指す。

このモジュールも文字位置検出モジュールと同様に、ルールベースでのアプローチと機械学習をベースとしたアプローチがある。

ルールベースでのアプローチの場合は、指定した座標に該当項目のタグ（e.g. "合計金額" = [1341,784,987,431] 等）を定義することで、該当項目を抽出することができる。

一方で機械学習のアプローチでは、抽出項目のタグとなる単語や座標等の抽出項目の特徴を学習させることで、該当項目の抽出を行う。そのため、項目抽出モジュールも文字位置検出モジュールと同様に、実際に AI-OCR を導入したい業務で利用される画像の特性を見極めた上で、アプローチを判断することが開発において重要となる。

9.3 AI-OCR 特有の課題と考慮すべき点

本節では AI-OCR の特有の課題および品質向上へ向けた施策案についてまとめた。本節にて述べる特有の課題の対象は、前処理モジュール、文字位置検出モジュール、OCR モジュールおよび項目抽出モジュールを対象とする。また、ここで上げる考慮点は全て適用をする必要はなく、対象業務に合わせて、どの考慮点を検討すべきか抽出するためのものとする。特に、本節では QA4AI のガイドラインで定義された 5 つの品質特性（Data Integrity, Model Robustness, Process Agility, System Quality, Customer Expectation）を上位文書とし、AI-OCR の特有の課題および、その課題に対して有効とされる技術について記載する。

9.3.1 AI-OCR における Data Integrity

AI-OCR システムにおいては、Data integrity による観点がモデルおよびシステムの品質につながる重要な要素となる。AI-OCR における Data Integrity を考える上で、認識対象画像をレイアウト特性、文字特性、ノイズ特性、画像特性と 4 つの特性を考慮する必要がある（表 9.2 参照）。

まず、レイアウト特性については、認識対象の画像ファイルにどのようなレイアウトで文字が記載されているかを検討する必要がある。考慮するポイントとしては、定型（常に一定のレイアウト）/ 非定形（さまざまな形が含まれるレイアウト）に大別される。また、一枚の画像に含まれる情報量（文字数）や表構造を含む帳票、縦書きと横書きの存在する度合いや、改ページの有無、2up での記載等が考慮すべき点として上がる。

これらのポイントについて、機械学習のアプローチを行う場合には、事前に学習させておきたい特徴となっている。例えば、横書きの帳票のみを学習したモデルにおいて、縦書きの文字を未学習で認識することは難しく、事前にこのような検討事項に該当する読み取り対象の帳票画像がある場合には、学習データにきちんと含める必要がある。

次に文字特性については、OCR される文字そのものに対する考慮点である。文字特性は、大きく印字と手書きに大別される。特に手書きの場合については、1 と 7 など、人間が見ても判断が分かれるものに対して、特定の筆跡を学習しない限り、AI が正しく認識することは難しいと考える必要がある。一方で印字の場合についても、様々な観点から考慮する必要がある。例えば、印字・手書きいずれの文字においても、太字/イタリック等の文字の修飾や、明朝体/ゴシック体といったフォントの種類、また、文字の種類も全角/半角といった観点やひらがな/カタカナ/アルファベット/漢字/記号/環境依存文字への対応、漢字の対応についても JIS 第三水準・第四水準への対応といった観点を考慮する必要がある。ただし、これらのパターンを全て網羅したモデルを構築することは現実的ではなく、実際の業務上で対象とする項目に合わせて、必要な学習データを定義する必要がある。実際のやり方としては、金額を OCR したいのであれば、学習データは数字に絞るだけでよく、複雑な数式を習得したいのであれば、環境依存文字を含めた記号まで学習する必要がある。そのため、導入する業務や構築するシステムによって、どこまで OCR が対応できていたら、業務利用できるのか、あるいはどのラインのエラーまで受け入れられるのかを事前に検討する必要がある。

ノイズ特性については、読み取り対象画像に含まれる文字の滲み等、ノイズに対する考慮点である。ノイズが入る原因是様々かつ、特定し辛い部分もあるが、文字の滲みや画像の傾き、文字切れもノイズとして扱う。また、こういった意図しないノイズの他に、ハンコによる文字かぶりや複写禁止の背景上の文字もノイズとしてなりうる。さらに、カメラ撮影された帳票の場合は背景の写り込みが想定され、背景除去を重点的に行う必要がある。程度にもよるが OCR 処理がピクセル毎の陰影から文字を分類している都合上、文字かぶりや背景上の文字は文字認識の精度向上において、ボトルネックになりかねないため、このようなデータが実際に業務で利用する母集団にどの程度存在するのか事前に把握しておくことが有効である。そのため、これらのノイズが含まれている画像について

表 9.2 Data Integrity について考慮すべき特性

特性の分類	考慮すべきポイント
レイアウト特性	対象画像におけるレイアウト特性の例 <ul style="list-style-type: none"> ・定型レイアウト/非定型レイアウト ・1 画像に対する文字の量 ・表構造の有無 ・縦書き、横書きの構造 ・改ページの有無 ・1up/2up 等の文章レイアウト
文字特性	OCR 対象の文字そのものの特性の例 <ul style="list-style-type: none"> ・印字/手書き ・文字の修飾（太字/イタリック/下線） ・全角/半角 ・文字種（ひらがな/カタカナ/英字 等） ・記号/環境依存文字 ・フォント（明朝体/ゴシック体 等） ・ロゴ ・林など複数の漢字で 1 つの漢字を構成している文字
ノイズ特性	画像内に含まれるノイズ特性の例 <ul style="list-style-type: none"> ・文字の滲み/レ点 ・光の反射 ・判子被り ・二重取り消し線 ・背景の写りこみ ・影の写り込み ・画像の傾き ・複写禁止等の背景ノイズ
画像特性	画像を定義付けるプロパティ特性の例 <ul style="list-style-type: none"> ・解像度（DPI） ・画像サイズ ・モノクロ/カラー ・画像の明度/彩度

てもどこまで正確な結果が必要なのか議論した上でターゲットとするモデルの品質を決めるといい。

画像特性については、数値上で定義できる画像の特徴を考慮点として挙げる。主な論点は、解像度（dot per inch）であり、dpi の制約はプロジェクト前に検討する必要がある。研究ベースでは 300 dpi 以上でモデルを構築している場合が多いが、実際に業務へ適用するとなると、ファイル転送における容量の制約から dpi を下げたものが必要になる場合もある。そのため、Data Integrity の観点としては、実際の業務でやり取りされるファイルの解像度と同様のものを利用し、学習を行う必要がある。また、dpi 以外には、画像サイズ、モノクロ/カラー、明度/彩度が考慮点として挙げられる。

以上のように、Data Integrity の品質特性からは、レイアウト特性、文字特性、ノイズ特性、画像特性の 4 つの特性を考慮する必要がある。しかし、これらの考慮点を全て含めた開発および、テストを含めた品質保証活動を行うことは、現実的ではない。そのため、AI-OCR を導入したい業務に対して、これらの考慮点がどれくらい含まれているかを分析・把握することで、どのような学習データでモデルを構築すべきなのか、モデル構築の難易度はどれくらいなのかを見定めることができる。その結果を以って、モデル開発のスコープを定義し、モデルがどこまでできたら業務上で使えるかを定義した上で、開発やテストケースを実施していくことが効果的な品質保証活動になるだろう。

9.3.2 AI-OCR における Model Robustness

AI-OCR における Model Robustness は、モデルが導き出す精度の測定における考え方を主として扱う。

堅牢性の高いモデルとは、PoC からベータ版開発、ベータ版開発から本番開発といったフェーズの移行時にモデルが陳腐化しないことを堅牢性が高いと定義する。例えば、汎化性能を確保したハイパーパラメータの検討や妥当な精度指標の検討、実データに対して想定外のノイズに頑健かを評価する。AI-OCR はアウトプットの出力結果がわかりやすい特性上、精度の定量評価が比較的容易である。しかし、数値情報として精度を解釈するだけでなく、多面的な指標によって評価することによって実業務で利用可能か判断を助けることになる。

精度指標は AI-OCR の出力結果の文字と実際に出力された文字の組み合わせで表現することができる。この組み合わせは表 9.3 の混同行列で表現する。

表 9.3 混同行列

実際の文字 / 予測文字	文字 A(Positive)	文字 B(Negative)
文字 A(Positive)	真陽性 True Positive: TP	偽陰性 False Negative: FN
文字 B(Negative)	偽陽性 False Positive: FP	真陰性 True Negative: TN

代表的な精度指標としては、正解率、適合率、再現率、F 値が利用される。これらの指標の特徴を表 9.4 に示す。

表 9.4 代表的な精度指標

観点	指標	計算式	特徴
文字評価	正解率	(TP+TN)/ (TP+TN + FP + FN)	単純な OCR モジュールの精度評価 に対して有効（計算しやすい）
	適合率	TP/(TP+FP)	誤分類は少ないが、判定漏れ (値が取得できない割合)が多い
	再現率	TP/(TP+FN)	誤分類は多いが、判定漏れが少ない
	F 値	(2*適合率*再現率)/ (適合率 + 再現率)	前処理モジュールから項目抽出モジュールまでの全体の精度評価に利用
業務効率化評価	レーベン シュタイン距離	正解文字列から読み取り 文字に対して、文字の挿入・置換・削除の数から 編集距離を算出	OCR 結果が仮に誤った際に、その修正に どのくらい工数がかかるかの試算に利用
	項目単位精度	完全一致で正解した項目 数 / 読み取り対 象の項目数	業務上 AI-OCR がミスなく読み取れる割合

上記のような指標はあくまで文字全体に対しての評価であり、あくまでモデルそのものの評価を表している。

ただし、AI-OCR を業務へ導入するに当たっては、文字単位の指標だけでなく、項目単位の精度（完全一致で正解した項目数 / 読み取り対象の項目数）でも評価することが重要になってくる。この指標で評価することで、業務削減効果や ROI の試算の目安となる。ただし、項目単位の精度は、読み取り対象の項目によって精度が大きく変化する。例えば、住所を項目として読み取りたい場合、読み取る文字数は約 10-20 文字だが、人の名前を項目として読み取る場合は約 27 文字程度が読み取り対象となる。したがって、項目単位の精度は、文字数の絶対数が多い住所の方が低くなる可能性が高く、各項目の特徴を踏まえた精度評価が重要となってくる。

また、業務削減効果という観点からはレーベンシュタイン距離が採用される場合もある。レーベンシュタイン距離は AI-OCR から出力結果が実際の正解データと比較して、文字の挿入、置換、削除の観点からどの程度離れているか測定する方法で、AI-OCR で読み取った結果を人が補正し、システムへ入力する業務フローの場合非常に有効な評価指標となりうる。

ここまで述べてきたように、一言で精度といっても様々な指標が存在し、その特徴を理解した上で、解決したい課題に対して多面的に測定することが重要となる。

その一方で、精度を導き出す元となるデータ（テストデータ）もまた重要である。テストデータを考慮する上では、学習データおよびテストデータ、交差検証用データの比率が重要になってくる。また、テストデータには、業務上利用する様々な帳票を考慮し、偏ったデータやほぼ同じレイアウトのデータで評価していないことを確認する必要がある。その他に、テストデータへノイズの含まれる割合や、業務上の出現頻度が高い帳票と難易度が高い帳票等が混在していることも併せて確認することも必要になってくる。十分に内容が吟味されたテストデータを多面的な指標で測定することで、モデルの良し悪しや業務適用へのフィジビリティを推測することができる。

Model Robustness の考慮において、継続的に学習していった際に精度が安定するか否かも評価観点となり得る。したがって、学習データを追加した際の精度の向上度合いも確認することで、運用時の可用性がより明確になる。また、Model Robustness においては AI-OCR のエラー原因を分析し、エラーの傾向を把握することも品質保証活動として重要である。読み取りエラーの原因が文字位置検出モジュールに起因しているのか、OCR モジュールに起因しているのか、あるいは項目抽出モジュールに起因しているのかを把握すれば、そのモデルの特徴や導入後のリスクがより明確になる。

9.3.3 AI-OCR における Process Agility

2つの側面から AI-OCR システムにおける Process Agility を検討する。1つ目はモデル開発の機動性、2つ目は運用段階の機動性である。

モデル開発時の機動性の主要論点は主に、データの部分にフォーカスされている。

AI-OCR は様々なビジネスプロセスに組み込まれるが、紙の帳票を AI-OCR 対象とする特性上、個人情報が含まれたセキュアなデータが含まれることがある。そのため、AI-OCR のモデル構築を外部へ委託する際には、個人情報のマスクが必要となる場合がある。また、画像データから学習用・テスト用のデータを作成する際には、文字の書き起こしが必須となってくる。そのため、収集したデータの取り扱いのルールがあらかじめ決まっていないと、開発の立ち上がりが遅れ、機動性が低下するリスクがある。

開発時に導入後の ROI 試算や目標とする精度および、精度の前提となるデータ（手書きを対象とするか否か等）の検討が不足していると、モデル開発の終了条件が定義できず、モデル開発が完了しなくなる恐れもある。

運用時の機動性については、システム上で精度を常に監視し続ける仕組みの構築が必要となってくる。モデルの精度低下を素早く検知し、業務影響を極力減らした状態で、モデルの改修ができる体制を構築していくことが機動性を高めるポイントとなる。

本節では、AI-OCR をビジネスへ組み込む際には、次図に示す開発プロセスを提唱する。

図 9.2 AI-OCR システムの開発プロセス



まず、導入検討・アセスメントフェーズでは AI-OCR が有効だと思われる業務を選定し、AI-OCR 導入後の業務整理を進めるとともに、データの調達の可否判断や調整を進める。この段階では実運用で利用するデータを直接確認することを推奨する。その後、実運用で利用するデータを元に、座標指定型および、帳票認識型のどちらが有効か、アプローチの定義を行う。

モデル開発/要件定義フェーズでは、システム要件定義とモデル開発を並行して行うことを推奨する。これは、実現したい要件と実際に組みあがるモデルを整理し、システム構築を行う際の全体像を整理することが目的である。そのため、モデル開発時の状況を常に確認し、試行錯誤の結果のもとで、要件を微調整していくことで、システム化の実現性を高めていく必要がある。特に AI-OCR は画像認識の技術を適用する特性上、開発時に構築される予定のモデルが自社で用意できる環境で動作可能かも併せて検討を行う。これらの作業が完了した段階で、実際にシステム開発へ踏み切るか判断を実施する。このときにモデルの精度のみでなく、実際にシステムへ組み込むことが可能かも含めて、要件定義と照らし合わせながら判断を行う。

本開発フェーズでは、モデル開発/要件定義フェーズで発生した課題に対して、モデルのパラメータ設定やルールベース補正の追加、学習データの追加を行うことで、モデルの性能向上を実現しながら、システム開発を平行して進めていく。これらが完成した段階で、システムテスト/リリースを行い、運用フェーズへ移行していく。

運用フェーズにおいては、システムの動作のみでなく、精度の監視を継続的に行うことで、運用時に AI-OCR で読めない帳票の特定や精度改善を継続的に実施していく。また、モデルの更新を行った際には認識率の測定だけでなく、更新前に認識できていた帳票の精度の確認も重要となる。

9.3.4 AI-OCR における System Quality

AI-OCR の開発・導入にあたり一般的なシステム開発の手法ももちろん必要だが、この節では AI-OCR 特有の論点を取り上げる。

AI-OCR では様々な言語や文字を認識することを目的としているが、場合によっては辞書による

補正が必要となる場合がある。特に業務用語や会社名や住所などの定型化されている言葉に対しては、辞書による補正がシステム全体の品質を高める上で有効である。これらの辞書は常に一定である場合は少なく、例えば、住所の場合は市町村の統廃合等により住所の変更が発生する。そのため、システムとしては、このような辞書を定期的に更新していく仕組みも併せて構築する必要がある。

これらの辞書に関わる AI-OCR の特性として、法令の変更等、社会的変化による対応も必要となる。例えば、新元号の辞書登録や、税率変更によるフォーマットの変更への考慮も重要となる。

また、AI-OCR を組み込む際に全ての帳票が精度 100 %で処理することは難しい。そのため、業務によっては AI-OCR の処理結果を人が確認するプロセスを導入する必要がある。従って、AI-OCR が誤認識した帳票にアラートを出す仕組みや、AI-OCR を組み込んだシステムそのものの使いやすさも重要な観点となってくる。

AI-OCR を業務プロセスへ組み込むという観点では、業務時間や必要な帳票の処理枚数から、インフラ構成を決めることが重要である。対象業務はオンライン処理が必要なのか、夜間バッチ処理による処理が可能かもシステム・インフラを設計する上で検討する必要がある。

9.3.5 AI-OCR における Customer Expectation

AI-OCR を開発・導入を行う目的は、作業代替、作業スピードそして、作業品質への寄与があると定義し、表 9.5 のような検討を行った。

表 9.5 AI-OCR における期待値の分類

期待値の項目	期待値の深さ
作業代替のレベル	ユーザーの作業完全置き換え
	ユーザーの作業支援
	参考情報として利用（保存目的）
作業スピードのレベル	ユーザーより早い
	ユーザーと同等
作業品質のレベル	AI は人ではあり得ないミスをする（均質化はされる）
	人は人によって作業品質は異なる

まず、作業代替についてだが、AI-OCR の開発・導入をするにあたり、作業代替の度合いをどこまで求めるかが、最終的な品質に寄与すると考えられる。作業代替のレベルには、完全置き換え (AI-OCR により作業を全て置き換える)、作業支援 (AI-OCR により通常の作業の支援を行う)、そして参考情報としての利用 (AI-OCR により電子データ化を行い、保存等を行う) に分けられる。この目的の度合いによって最終的に求められるモデルの品質が変化する。開発を開始する際には、こ

の作業代替レベルにより達成品質を検討することを推奨する。

また、作業スピードについては、AI-OCR の開発・導入にあたり人並の作業スピード（つまり処理速度）が求められるのか、それとも人よりも早く処理する必要があるのかを併せて検討する必要がある。そして、その期待を満たすための処理速度を実現するためのインフラを用意できるかも論点となる。

AI-OCR の開発・導入にあたり、作業品質の向上レベルも重要となってくる。これは、人力での作業は人によって作業のバラツキがある一方で、AI-OCR は人ではあり得ないミス（例えば、「=」と「ニ」の誤認識等）が発生する。これらの検知、修正がどれくらい業務に影響があるかも併せて検討する必要がある。また、これらのミスを人が事前検知できるような技術として確信度の活用等の方法もあるが、このような技術の導入も含めて検討をしていく必要がある。

9.4 品質保証技術の AI-OCR 適用例

9.4.1 メタモルフィックテスティングを適用した品質評価例

開発した AI-OCR の品質を Model Robustness の観点で評価する手法として、3.3.2 で述べたメタモルフィックテスティング（以降、MT とする）を適用したテスト設計を提案する。まず、MT が AI-OCR の品質評価に有効と考える 2 つの理由を述べる。① AI-OCR には高い頑健性（Robustness）が要求される AI-OCR に入力される帳票は、顧客の業務内容や記入する人などに起因した様々な特徴を有する。背景色が通常より濃い帳票、くせ字・崩し字で記入された帳票、項目に複数行で記入された帳票など様々な帳票が存在するが、どのような特徴を持つ帳票に対しても高精度で認識可能な頑健性に富んだ AI-OCR が、高品質の AI-OCR と言える。② AI モデルの弱点や不良を効率的に特定できる AI モデルは学習データから帰納的に作成されるため、内部ロジックを把握した上で入出力関係を定義した検証（Verification）をすることは非常に困難である。そこで、MT を適用して様々なテストケース間の出力結果の比較を繰り返すことで、妥当性確認（Validation）を実施する。入力データのどのような変化に弱いのかを特定することが、効率的なモデルの弱点や不良の特定に繋がる。そして、不良やモデルの弱点が明確になった場合は、学習パラメータのチューニングや再学習すべきデータの特定、内部ロジックの見直し等の対策を検討する。以上 2 つの理由から、モデルの頑健性を確認する MT は AI-OCR の品質評価に有効である。メタモルフィック関係を利用した多様なテストケースを作成し、テストすることで Model Robustness の観点で品質評価ができる。しかし、メタモルフィック関係は無数のパターンが考えられるため、テストケースは際限なく作成できてしまう。現実的なテストケース数で最大限の検証効果を得るために、メタモルフィック関係を用いた帳票への変化の加え方を工夫する必要がある（参考文献 [1]）。そこで以下に示すような導出例に従い、得られた観点をメタモルフィック関係に基づく変換に利用することで、テストケースの絞り込みや優先度の検討をするのが望ましい。

<観点導出例> (1) AI 未搭載の従来 OCR が不得意としていた認識パターンを洗い出す (2) 顧客業務で発生する帳票の特徴や発生頻度を分析する (3) 誤認識が顧客業務に大きく影響するパターンを明確にする

例えば上記の (1) について調査し、「項目欄の罫線と文字が重なったケースの認識精度が低い」という従来 OCR の課題が判明したとする。本課題を AI-OCR が克服できているか確認するため、罫線と文字が重なっていない帳票①と、Noise-based(出力結果に影響を及ぼさない入力の変換)というメタモルフィック関係の変換を利用して帳票①に変化を加え(参考文献 [2][3])、罫線と文字を意図的に重ねた帳票②を作成する。そして、帳票①と帳票②の認識結果を比較することで、従来の課題を AI-OCR が克服できているか評価する。また (3) について調査をし、誤認識が顧客業務に大きく影響するパターンとして金額項目の認識があったとする。本パターンでは、例えば「1」と「7」など字の形状が近い手書き数字を高精度で認識する必要がある。そこでまず金額項目に手書き文字で「1」が書かれた帳票を用意し、Heuristic(元データに近い入力に変化)というメタモルフィック関係の変換を利用して(参考文献 [2][3])、「1」に近い「7」へと字体を変えた帳票を複数作成する。そして作成した帳票間の認識結果を比較し、手書き文字の「1」と「7」を明確に識別できるかを重点的かつ優先的に評価する。このように優先度が高いテスト観点を、メタモルフィック関係を利用したテストケース作成に活かすことも有効である。上記で述べたように、効果的なメタモルフィック関係の検討が、MT を適用したテスト設計で重要なポイントとなる。

[1] 中川純貴, 「Deep Learning 搭載ソフトウェアの品質評価/テストを前進させる取り組み」, JaSST' 19 Hokkaido, 2019. [2] C. Murphy, et al., Properties of Machine Learning Applications for Use in Metamorphic Testing, SEKE2008, 2008 [3] C. Murphy, Applications of Metamorphic Testing, 2011, <http://www.cis.upenn.edu/cdmurphy/pubs/MetamorphicTesting-Columbia-17Nov2011.ppt>

9.4.2 帳票項目分析を利用した品質評価事例

項目抽出を有する AI-OCR では、項目毎の特性を考慮した品質評価を行う必要がある。そのため、Data Integrity の観点で品質評価を行う際には、前述の AI-OCR 特有の考慮すべき点である 4 つの特性（レイアウト特性、文字特性、ノイズ特性、画像特性）のみでなく、帳票の抽出項目毎に特性を考慮する必要がある。

AI-OCR WG では、AI-OCR にて項目抽出の対象になることが多い項目を標準項目特性として定義し、帳票毎に考慮すべき項目を帳票特化項目特性と定義した（表 9.6 参照）。

標準項目特性では、基本的にどの帳票にも記載される且つ項目抽出対象となり得る項目を定義した。標準項目特性は、「金額」「日付」「会社名」「電話番号」に対して、考慮すべき特性を定義した。また、帳票特化特性については、「請求書」「アンケート」「マイナンバー帳票」「振込依頼書」「帳票の明細行」に対して、帳票毎に考慮する点を挙げた。この帳票特化特性は、今後、AI-OCRWG を通して事例を追加していく予定である。

表 9.6 AI-OCR において抽出の対象となる項目

特性種別	概要	例
標準項目特性	どの帳票にも記載され得る、一般的な項目で考慮すべき特性	金額や日付など ※請求書、口座振替依頼書など 様々な帳票で利用されている
帳票特化特性	帳票毎に個別に記載され得る、帳票毎に特殊な項目あるいは帳票全体における考慮すべき特性	伝票番号など ※対応する帳票によって記載ルールが大きく異なっている

これらの特性を参照することにより、帳票毎のテストデータ作成の観点を項目ごとに充足することで、品質保証活動の一助となるものと考える。

標準項目特性

標準項目特性として定義した「金額」「日付」「会社名」「電話番号」の項目は、基本的に多くの帳票で項目抽出対象となることが多い。従って、これらの項目を標準項目と設定し、AI-OCR のテストオラクルを作成する上で、考慮すべき観点とその例を次表に記載した。

例えば、「金額」のような単純な項目でも 10 個もの考慮すべき特性があり、実運用で読み取りを行うデータがどの特性に該当しているか把握し、適切なテストデータを用意することで、適切なテストケースを作成できると考える。

合計金額

- ・ ¥マークの有無（例：¥10,400）
- ・ ¥マークと金額間のスペース
- ・ 金額背後の表記（例：円、-、也など）
- ・ カンマ区切り（例：¥1,000,000）
- ・ ポップス区切り

請求金額

- ・マイナス記号表記（例：▲100,000）
- ・ドル・ユーロなど外貨表記（例：€ 800,00）
- ・税抜き、税込みの記載（例：¥132,220（税抜））
- ・「百」「千」など単位のプレ印字

金額	百	十	千	万	円
	1	2	3	4	5 6 7

- 漢字記載（例：六百五十円 也）

日付

- 西暦ゼロ埋めあり（例：2020/04/02）
- 西暦ゼロ埋めなし（例：2020/4/2）
- 年月日埋めの西暦（例：2020 年 4 月 2 日）
- ゼロ埋めあり和暦（例：令和 02 年 04 月 22 日）
- ゼロ埋めなし和暦（例：令和 2 年 4 月 22 日）
- 元号省略和暦（例：令 02/04/22）
- 元号英字省略和暦（例：R02 年 4 月 22 日）
- 元年表記和暦（例：令和元年 5 月 1 日）
- 元号選択式（チェックマークや丸囲い）

1.昭和	年	月	日
2.平成			
3.令和			

- 「元号」「年」「月」「日」のプレ印字

平成 31 年 2 月 28 日

会社名

- 株式会社の表記（例：株式会社、(株)、(株)）
- 前株/後株表記（株式会社 QA4AI/QA4AI 株式会社）
- 印鑑被り



- 支店名・店舗名（例：タイムズダイエー富田林店）

電話番号

- TEL の記載（例：TEL: 08012345678）
- 10 桁以上の連番（例：08012345678）
- ハイフンあり（例：080-1234-5678）
- () あり（例：080(1234)5678/(03)12345678）
- FAX の記載（例：024-1234-5678 (FAX)）
- 代表番号（例：(03)(代)1234-5678）
- 市外局番省略など 8 桁番号（例：1234-5678）

請求書における帳票特化特性

請求書は AI-OCR の対象となることが多く、請求書読み取り専用のプロダクトが数多く提供されている。

その一方で、実際に業務利用可能なモデルかどうかは適切なテストケースを作成し、評価を行う必要がある。

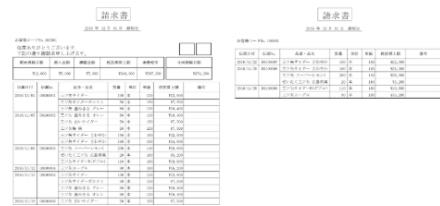
この節では、標準項目を除いた請求書ならではの考慮すべき特性を挙げた。利用方法としては、実際に業務で読み取る対象となり得る請求書が、考慮すべき特性をどこまでカバーしているか同定することで、テストケースの作成あるいは間引きに利用できると考える。請求書特有の特性を以下に記載した。

請求書全体

- FAX の利用による歪み/ノイズ
- 明細への情報の記載（明細行に合計の記載など）

2020/1/1	00001 緑茶500ml	5	110	550
	00002 ダージリンティー	10	150	1,500
	00003 純茶ラテ	8	330	2,640
			<合算合計>	4,690
			<消費税>	375
			<合計>	5,065

- 複数ページ（2 ページ目以降フォーマットの違いあり）



品名	数量	単価	金額
緑茶500ml	5	110	550
ダージリンティー	10	150	1,500
純茶ラテ	8	330	2,640
			合算合計
			4,690
			消費税
			375
			合計
			5,065

帳票名

- 文字間隔の有無

請 求 書

- 文字の囲い

請求書

明細

- 網掛けの有無

品目	単価	数量	金額
商品 1	15.000	5	75.000
商品 2	20.000	4	80.000
商品 3	9.000	6	54.000
商品 4	3.000	7	21.000

アンケートにおける帳票特化特性

アンケートは手書きで実施されることが多く、その結果の集計作業には時間を要する。従って、AI-OCR の対象となることが多く、アンケート文書における AI-OCR の考慮点を下記に記載した。

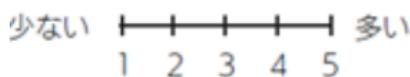
また、アンケートの特性として、フォーマットを自社で規定できる幅が大きく、本節で挙げる考慮点を把握した上で、AI-OCR で読み取りやすいフォーマットを定義することへの利用も想定している。

丸囲い式選択

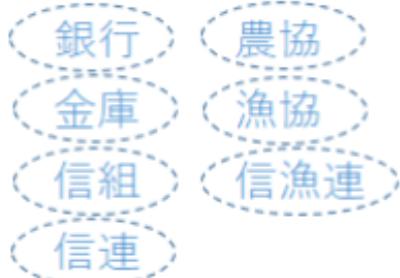
- 対象の全体の丸囲いあるいは数値のみの囲い

1. スタッフの接客・対応はいかがでしたか。
①とても満足 ②満足 ③ふつう ④不満 ⑤とても不満

- 丸囲い項目に対してのチェック印



- 点線での囲い



チェックボックス式選択

- チェック方法（レ点、塗りつぶし、○印、×印、線のみなど）

良い	やや良い	普通	やや悪い	悪い
<input type="checkbox"/>				

フリー記述

- 複数行記載の考慮

その他お気づきの点がありましたらご記入ください。

- はみ出し記載の考慮

その他 ()

マイナンバー帳票における帳票特化特性

平成 27 年 10 月 5 日より施行されたマイナンバー法により、自治体で取り扱う帳票や各種申請書類にマイナンバーの記載項目が追加された。このようにマイナンバーの入力が必須となっている帳票を本節ではマイナンバー帳票と定義する。

マイナンバー帳票のフォーマットは各自治体に依存している傾向があり、下記にてそのフォーマットのばらつきに対する考慮点を記載した。

マイナンバー

- 線による数値の区切り

個人番号									
------	--	--	--	--	--	--	--	--	--

- 4 文字毎の太文字短径区切り

(個人番号)									
--------	--	--	--	--	--	--	--	--	--

- 1 文字ずつのボックス区切り

個人番号									
------	--	--	--	--	--	--	--	--	--

- グレー背景

個人番号									
------	--	--	--	--	--	--	--	--	--

氏名

- 記入欄への (氏)(名) の記載

氏名	(フリガナ)	
	(氏)	(名)

振込依頼書における帳票特化特性

振込依頼書は銀行などの金融機関にて振込の手続きを行うための帳票である。金融業界のペーパーワーク効率化の文脈にて、振込依頼書が AI-OCR の読み取り対象となることが多い。

その一方で、振込依頼書の帳票レイアウトは各種金融機関で異なっており、同様の項目でもバラツキが大きいため、下記にて振込依頼書の考慮点を挙げた。

全体

- 振込金受領書とセットとなるため振込金受領書の取り扱いの考慮

振込先金融機関名

- 選択式金融機関

振込先銀行	
<input type="radio"/>	み ず ほ 銀行 越谷支店 (普) 1234567
<input checked="" type="radio"/>	埼玉りそな 銀行 越谷支店 (普) 1234567
<input type="radio"/>	三菱東京UFJ銀行 越谷支店 (普) 1234567

- 同一枠内記載（直列）

山口 銀行 和木 支店

- 同一枠内記載（並列）

振込先 銀行名	ゆうちょ 銀行 〇一八 支店
------------	-------------------

- カンマ区切り（例：山口銀行、和木支店）

預金種別

- 丸囲い式選択（例：普通 当座のいずれかを丸で囲う）
- チェックマーク式選択（例：□普通 □当座）

口座番号

- 記号番号（例：記号/12345 番号/12345678）
- 5桁あるいは6桁の古い番号（例：12345）

住所

- 〒マークのプレ印字の有無
- 電話番号のセット記入

帳票における明細行の帳票特化特性

多くの帳票には明細行が含まれており、明細の読み取りは AI-OCR のタスクの中でも難しい部類にあたる。その理由としては、明細には複数の項目が複雑に組み込まれており、かつそのフォーマットのバリアンスも非常に大きいことが多い。従って、ここで述べる帳票における明細行の特性は、こ

これまで述べてきた対象項目別ではなく、AI-OCR で読み取った際に発生しうるリスクから明細行を読み取る際に考慮すべき特性を挙げるアプローチをとった。

文字認識精度が低下する明細行の特性

- 枠線との被りによる文字誤認識（プレ印字の線との被り）

117854

- 枠線を誤読（点線区切りを 1 と誤読）

数量	金額
15	3,000

対象項目と値の紐づけ失敗を引き起こす明細行の特性

- グループサプレス

商品区分	商品コード	得意先コード	日付
1	001	0001	05/01
			05/02
	002	0002	05/01
			05/03
2	002	0002	05/04
			05/01
	001	0001	05/01

- 線無し帳票

Date : 2020-01-10	Payment Terms : 60 Days		
Ship Date : 2020-02-20	Shipping Terms :		
<hr/>			
Description			
Unit Price			
QTY			
Amount			
<hr/>			
High Visibility Orange Safety Vest	19.99	4 Pcs	79.96
Ingersoll Rand C38121-006-VS 1/4-Inch F	109.00	5 Pcs	545.00
Enerpac RC-504 50 Ton Single Acting Cyl	1227.00	5 Pcs	6,135.00
Airwolf 3D Printer AW10 XL - 2 LB Filament	299.00	2 Pcs	4,798.00
SaintSmart ABS-107 ABS Filament (Black)	59.00	7 Pcs	413.00
Liberty Pumps 3311 1/2-Horse Power Pump	205.00	2 Pcs	510.00
Sun EMGV-12 N-D2-U Scientific Valve	68.00	6 Pcs	408.00
<hr/>			
SUB TOTAL			
Discount			
Add. Discount			
VAT			
S & H			
TAX			
<hr/>			

- 1 セルに複数項目記載

商品名・単価・消費税

商品 1・500円・50円
商品2・100円・10円

- 値同士の接近

数量 単位 単価

10	C	1,000
5	C	2,000

- キープレイク毎の明細出力

取引先別明細表

日付	取引No.	取引先コード	取引先名
2020/01/01	R0000001	T0000001	株式会社エーアイオーシーフール
<hr/>			
商品名	数量	単価	金額
商品A	1	1,000	1,000
商品B	2	2,000	4,000
商品C	3	3,000	9,000
商品D	4	4,000	16,000
商品E	5	5,000	25,000
商品F	6	6,000	36,000

日付	取引No.	取引先コード	取引先名
2020/01/02	R0000002	T0000002	AI-OCR株式会社
<hr/>			
商品名	数量	単価	金額
商品A	1	1,000	1,000
商品B	2	2,000	4,000
商品C	3	3,000	9,000
商品D	4	4,000	16,000

- 明細の高さが可変

商品名	数量	単価	金額
チョコクリームアイス	1	1,000	1,000
紅茶バー	2	2,000	4,000
コーンフレークシュガー	3	3,000	9,000
フルーツヨーグルトピーナ	4	4,000	16,000
シェフのこだわりふわふわホットケーキ ～たっぷりホップクリーミーを添えてみました～	5	5,000	25,000
ボロネーゼソース	6	6,000	36,000

9.5 推奨する品質評価レベル

本章では下図の 4 段階で品質レベルを定義する。この品質レベルは AI-OCR 全体のシステム開発における品質評価レベルでなく、構築したモデルに対しての評価レベルを記載する。従って、Data Integrity と Model Robustness の観点において、Customer Expectation におけるユーザーの期待値と、

開発しているモデルの実態をすり合わせることに利用へ繋がることを想定した。例えば、精度 95 % といった結果が開発段階で導き出された際に、その結果がどの程度信用できるものなのか（品質レベルが高いのか）といった評価への利用を想定している。AI-OCR の開発にあたり、最も重要な何のデータを利用して開発、検証を行ったか（学習/評価データの観点）とどうやって評価したか（評価指標）の適用度合いから 4 段階の品質レベルを策定した。

品質レベル	学習/評価データ	評価指標	評価の状態
レベル 1 : モデル初期レベル	実環境で利用するデータの偏りを検討せず、サンプリングで収集したデータを利用している	単一の精度指標で評価	モデルの性能は保証できないが味見評価可能な状態
レベル 2 : モデル最適化レベル	実環境で利用するデータの偏りを検討せず、サンプリングで収集したデータを利用し、学習/テスト/検証でデータを分割している（過学習を考慮している）	複数の精度指標で評価	構築したモデルそのものの性能は評価できてるが、実運用で利用可能かは判断できない状態
レベル 3 : 業務適合レベル	レベル 2 の観点を達成した上で、未学習の実運用で利用されるデータを利用	複数の精度指標の妥当性を含めて評価	構築したモデルが業務上でも同様の性能で利用できることを評価できている状態
レベル 4 : 業務効率化レベル	レベル 4 の観点を満たしたうえで、実運用相当のデータセットで精度を評価している	精度と業務効率化の度合いを関連付けて評価	実運用時に業務効率化が可能なところをまで評価ができている状態

まず、品質が最も低いと定義したモデル初期レベルでは、モデルの精度がほとんど保証できない状態を示す。このレベルでは実際にアルゴリズムを選定する際の味見評価程度は可能だが、そこから導き出される精度は信用できない状態となる。このときの学習/評価データは実環境で利用するデータの偏りは考慮されておらず、サンプリングされたデータを利用し、単一の精度指標で評価したものとなる。

モデル最適化レベルでは、モデル初期レベルと同様にランダムにサンプリングしたデータではあるが、学習/テスト/検証用のデータセットを分割し、過学習の有無まで評価できてる状態とした。また、評価指標も単一指標による精度の評価ではなく、複数指標による精度評価を行っている。従って、構築したモデルそのものに対しては評価ができている状態であるが、一方で実運用で利用可能かは保証されていない。品質レベル 3 の業務適合レベルでは、学習/評価データは実運用の偏りを利用した未学習データを利用しているところに違いがある。また、精度評価指標も複数の指標を用いるかつ、その複数指標が妥当なのかを検討した上で評価をしている。そのため、構築したモデルは業務へ導入した際に、同様の精度で稼働することが評価できる。

品質レベル 4 の業務効率化レベルでは、学習/評価データにおいては、業務で利用する相当量のデータセットを利用して評価を行っている。また、精度指標も指標を複数利用するだけでなく、精度と業務効率化の度合いが併せて評価されている。そのため、業務効率化の程度も含めて評価できている状態となる。

これらの品質評価のレベルは開発全体を通して高めていくことを想定している。例えば、導入・アセスメントフェーズではレベル 1 まで実施し、モデル開発/要件定義フェーズではレベル 3 まで評価、本開発が完了し、運用をしながらレベル 4 まで高めていくといった利用方法を想定している。

longtable

lscape

10. 大規模言語モデル・対話型生成 AI

本章においては、生成 AI のうち、特に大規模言語モデル (Large Language Model : LLM) を用いるシステムに関する品質保証を対象とする。従来の AI においては、教師あり学習として定型の入出力に従う特定タスクに対する訓練を行っていたのに対し、LLM は一般的な一連の文章や対話に対する訓練に基づいている。結果として、プロンプトと呼ばれる入力での指示に応じて多様なタスクを扱える汎用的な対話型生成 AI が実現されている。非常に多様なユースケースを比較的小さな労力で実現できるポテンシャルが追及される一方で、その実装の特性から、自然だが指示や事実にそぐわない回答（ハルシネーション）をはじめとした品質の問題が指摘されている。本章では、LLM を利用したシステムに関する品質保証についてのガイドラインを示す。

執筆時点では、テキストの入出力を扱う生成 AI システムが主流となっており、図表や画像を扱うマルチモーダルなものは発展途上である。本章の議論は一般的であることを試みてはいるものの、具体的な例や技術はテキストの入出力を扱うものを想定している。なお、画像や動画、3D モデルなどのコンテンツ生成 AI については 5 章にて別途扱っている。

10.1 LLM・対話型生成 AI の概要

10.1.1 構成と動作

本章で用いる用語を表 10.1 に示す。以下では対話型生成 AI の基盤となっている LLM に関する前提知識を述べる。

表 10.1: 用語定義

用語	意味・解説
生成 AI (あるいは生成系 AI、Generative AI)	対象データの分布を学習することにより、テキストや画像などのデータを生成する AI 技術の総称。大規模言語モデルに基づき、プロンプトと呼ばれる入力により出力の制御を行う対話型のものが注目されており、本章ではこの対話型生成 AI を扱う。
Large Language Model (LLM)	文章や単語の出現確率を深層学習モデルとして扱う言語モデルを、非常に大量の訓練データを用いて構築したもの。LLM の例として OpenAI の GPT、Google の PaLM、Meta の LLaMA などがある。

基盤モデル	事前学習と呼ばれる、特定タスクに依存しない一般的な訓練データを用いて大規模な訓練を行うことで得られる汎用性が高いモデル。ファインチューニングなどのカスタマイズにより具体的なダウンストリームタスクに対応した個別モデルを構築する際に活用できる。
ダウンストリームタスク	一般性が高い事前学習を行った基盤モデルに対して、翻訳、要約、分類、論理推論などより具体的なタスクに適合させるための追加学習を行うことがあり、そのような具体的なタスクのこと。本章では単にタスクと呼ぶ。
ファインチューニング	一般性が高い事前学習を行ったモデルに対して、特定のダウンストリームタスクに特化した学習を行うプロセス。一から新たなモデルを訓練、構築するよりも、少ないコストで高性能な特定タスク向けモデルを構築することができることが多い。
RAG	Retrieval-Augmented Generation の略であり、外部のデータベースから取得した知識に基づいて LLM に回答を生成させることで、知識に即した回答がなされるようにしたり、ファインチューニングを経ず知識を拡張可能したりする手法。
ハルシネーション	生成 AIにおいて、一見自然に見えるが、事実や根拠に基づいていない週津力が得られる現象。事実関係を問うようなプロンプトに対して発生することが多い。
プロンプトインジェクション	LLMへの入力となるプロンプトにおいて、細工や誘導をするなどして、LLMの提供者が意図しない回答として、そのLLMの設定情報や悪意ある回答などを引き出す方法。
ジェイルブレイクプロンプト	プロンプトインジェクションによる攻撃のうち、特にLLMの提供者による設定や制限を回避するもの。

LLM は、大量のテキストデータで学習された言語モデルであり、Transformer ベースのものがほとんどである。モデルの分類としては、大きく以下の 3 つに分類できる。

Encoder モデル (auto-encoding モデル)

BERT に代表される、Transformer の Encoder 部分を使ったモデルであり、入力系列データの一部を隠し（マスキング）、隠した単語を予測させる、いわば穴埋め問題のような形のタスクを設定して学習が行われる。Encoder モデルは入力系列データの特徴表現を出力するため、後段で様々なダウンストリームタスクをするのに適している。例えば、文書分類、固有表現認識、対象の文書内から回答部分を抽出するタイプの質問応答などが挙げられる。

Decoder モデル (auto-regressive モデル)

GPT に代表される、Transformer の Decoder 部分を使ったモデルであり、入力系列データに含まれる単語に対して、その後に続く単語を予測するようなタスクを設定して学習が行われる。人間のフィードバックによる強化学習を導入することで、統計的に次の単語を予測するだけでなく、文脈に応じて人間が欲するであろう回答や、倫理的に問題のある回答を抑制することができ、実用的なレベルでテキスト生成ができるようになった。

Encoder-Decoder モデル (sequence-to-sequence モデル)

T5 に代表される、Transformer のアーキテクチャをそのまま活用しているモデルであり、穴埋め問題と続く単語の予測という 2 種類のタスクを設定して学習を行う。テキストを入力して、その内容に応じて別のテキストを出力するといったタスクへの応用に適している。例えば文書要約、機械翻訳、特定の文書を参照することなく自然文で質問応答をする対話システムなどが挙げられる。

以下では、最も普及している GPT を想定して、そのモデルへの入出力を考える。

モデルへの入力は単語列（正確にはトークンの列）であり、次に来る確率が高い単語を選択して文を紡いでいく。すなわち、入力も出力もテキストである。入力として質問文を入れると、質問への回答文が続いて生成されるという質問応答としての使い方もできる。

特に、モデルへの入力テキストをプロンプトといい、ユーザーがモデルに対して指示をしたり、補足情報を入れたりすることができる。生成される回答の有用性や品質（後に論じる）は、プロンプトをどのように工夫するかにかかっており、いくつもの手法が提案されている^{*1}。例えば、以下のような手法がある。

Few-shot prompting

少量の例文を提示することで、精度の高い回答を引き出すテクニック。

Chain-of-Thought prompting

問題を解くまでの一連の手順をプロンプトに含めることで、複雑なタスクに対応するテクニック。

LLM が学習している内容であれば、上記のような手法により、プロンプトを工夫することで、所望の回答を引き出すことができるが、特定応用ドメインに関する社内データなど、LLM が学習していない情報を持った回答が必要である場合は、以下のような手法を検討する必要がある。

RAG (Retrieval Augmented Generation)

LLM には変更を加えず、回答に必要な補足情報をすべてプロンプトに入れて回答を引き出す手法。

^{*1} <https://www.promptingguide.ai/jp> などにまとめられている。

ファインチューニング

事前学習モデルをベースに、特定ドメインのデータにより追加学習することで、特定タスクに特化した言語モデルを構築する手法。

スクラッチ学習

大量の訓練データを用いて、モデルのパラメータを一から学習させる手法。データセットに特化したモデルを作成することができるが、莫大な計算量や処理時間がかかるため、実行できる企業や組織は限られる。

10.1.2 活用のユースケースと留意点

生成 AI の活用例として、①テキストの生成、②情報抽出、③コードの生成、④ナレッジベースへの回答といった例がある。

①テキストの生成

製品のドキュメント作成やコピーライティング、特定の分野（ex. 子供向け）における短いストーリーの作成に用いられる。利用者は、生成された文章が意味を成しているか、倫理的に問題がないか確認を必要とする。

②情報抽出

テキストの分類、文章要約、機械翻訳等があり、例えば分類においては対象の文章の意味や顧客センチメントの測定、テキスト間の関係の判断といった利用がある。対象文章の表現や記述分野など様々な要因で分類の精度が低下する場合があるため、システム開発時の精度評価だけでなく、分類失敗時のフィードバックを利用者がするなど様々な考慮が必要である。

③コードの生成

Python、JavaScript、Ruby など様々な言語のプログラミング案の作成や、自然言語で記述された文章からの SQL の作成、サーバーサイドのインフラ構築や運用コード、ウェブサイトのデザインなどの例がある。生成されたプログラムの正しさや知的財産上の妥当性といった点で利用者は確認が必要である。

④ナレッジベースへの問い合わせ

FAQ ページに沿って作られたチャットボットや、企業マニュアル検索チャットボット等がある。正しい回答を得るには、ナレッジベースの構築時に利用する情報の正確さを高めるだけでなく、問い合わせに利用する質問文に十分なコンテキストを与えることなど様々な工夫が必要となる。ナレッジベースの情報を用いた回答をさせるため、RAG など手法を用いることになる。

こういったシステムの形態に加えて、生成 AI に担わせる「知能」の難しさに対応することが求められている。難しいタスクを担わせる場合、その実現のための技術的な課題だけでなく、そのタス

クの結果を利用者が判断する課題があるため、例えば以下のようなステップアップの形式で品質の問題に取り組むことが有効である可能性がある。

一般情報の検索や相談

LLM がその時点で持つ一般的な情報を引き出すため、チャットやアプリケーションから呼び出し利用する。一般情報として妥当な結果であるかや、その結果をどのような成果物に転用できるか利用者が検証する必要がある。

正確さを求める組織内情報の検索

RAG などの手法を利用し、組織内の情報を引き出す。組織内情報を整備し LLM に適切な情報を与える環境整備を要する。利用者は、組織内のどのような情報がシステムに入力がされているのか理解したうえで、結果を利用する必要がある。

組織内の知見に基づくプログラム案や文書案の作成

知見を大量の類似案件の情報から学習するか、ルールベースで知見を設定する形で情報を引き出す。利用者はどの程度の質で案が作成されるのか理解し手直しや修正判断を要することを理解し、結果を利用する必要がある。

組織内の知見を利用するエージェントの構築

エージェントとは例えば LLM に成果物レビューを担わせるなど、自然な会話や相談といった振る舞いをもつもの。エージェントに与える相談は、過去に蓄積した情報にする知見に限らず、未知の情報を与えることがあるため、未知の情報がどの程度過去の情報に近いものか、エージェント側か利用者側で評価し、不足した情報を都度補う必要がある。そのため、精度の定義やその評価指標を設定し、開発時のみならず運用時も事項に対する回答が期待に対してどのような達成であるかを継続的に監視し、都度改善をする必要がある。利用者側はエージェントの流動的な品質達成状況を理解しながら利用することだけでなく、その結果のフィードバックの重要性を理解する必要がある。

組織内の知見を利用した成果物の創造

成果物の創造や、成果物に対する意思決定といった高度な知能を担わせるもの。目的に対して成果物がどの程度実現できているかや、組織内の知見が目的に対してどの程度有効に利用できているか、を評価し不足があれば利用者側などが継続的に評価し利用する必要がある。正解のある成果物であれば実現性は正解の定義によって実現できるが、正解を定義できない成果物の場合、どの程度まで達成していれば十分であるかの評価尺度の設定が難しいため、利用者による定性的な判断が必要になる。

10.1.3 典型的な懸念

モデルの出力に関しては注意が必要な観点について概観する。より詳細な品質特性や評価手法については 10.2 以降で述べる。

情報セキュリティ

インターネット上の生成 AI サービスを利用する場合、①クラウドへの情報送信が情報の機密度等の観点から許容されるか、②送信した情報が LLM の更新時に訓練データとして学習されない保証が必要かどうかについて確認する必要がある。

いずれかがインターネット上の生成 AI サービスでは不十分と判断する場合は、自組織で構築した独自の LLM の利用が必要となる。ただし、その場合 LLM に期待される性能が低下する場合が多く、情報セキュリティと性能その他要件とのトレードオフを考慮することが必要である。

生成物の権利

2023 年時点において、著作権などの知的財産権に関する法的な判断は確定していないと考えられる。そのため、LLM の出力を利用する場合には、それが知的財産権の侵害リスクを理解し、リスクに対する対策を講じる（例えばプログラムを出力した際にオープンソースソフトウェアを含むか確認するツールを利用する）必要がある。

倫理的問題

LLM は、世の中の大量のデータを集約し学習していることから、世界に存在するバイアスをそのまま持っているリスクのあることが知られている。したがって、用途によっては LLM の出力が倫理的に問題ないかをチェックするレイヤを設けるなど、対策が必要な場合もある。例として、自社のチャットボットに生成 AI を組み込むときに、生成 AI の出力をそのまま出力しても社会的に問題とならないかの検討などがあげられる。

ハルシネーション

LLM が出力するテキストは、一見正しいように見える回答が実は誤っている場合があり、ハルシネーションと呼ばれる。倫理的問題と類似であるが、LLM の出力には誤りが含まれていることを前提として、利用の適否判断および利用が必要である。

訓練データの鮮度

LLM の学習には大きなコストを要するため、常に最新の情報を含むような訓練がなされているとは限らない。したがって、LLM 利用時にはいつ学習が行われたかの確認が必要である。解くべき問題によってそれ以降の情報が必要であれば、ベクトルストアで情報を補完するなど、LLM 単体ではない形態が必要となる。生成 AI サービスでも最近はこの仕組みを備えるものが増えてきている。

提供者によるバージョン更新

クラウドサービスに共通する注意点ではあるが、LLM の場合特にサービス提供者が行う更新が頻繁であったり古い版にアクセスできなくなったりする傾向がある。LLM の出力の監視が必要なケースも出ると想定される。

10.2 LLM における品質特性

上記で挙げたようにデータ、モデル、システム全体あるいは顧客の期待について分析、整理、評価する際には、扱う品質特性を明確にする必要がある。例えば差別がないという特性、すなわち公平性（倫理性の一種）を扱いたいのであれば、その観点からデータ、モデル、システム全体そしてプロセスをそれぞれ評価するとともに、顧客の期待について明確にする必要がある。以下では評価基準となり得る品質特性について示す。執筆時点では画像を扱うものなどマルチモーダルなものは発展途上であり、評価事例も少ないため、テキストを出力する LLM を想定して論じる。

LLM に対する評価は重要なトピックであり、本章執筆の時点（2023 年後半）においてもすでに多数の取り組みが行われている。例えば Chang らや Guo らによるサーベイ論文では 200 件を超える論文やプレプリント、ベンチマークなどが論じられているとともに更新も続いている [Chang+, arXiv23][Guo+, arXiv23]。以下ではこれらのサーベイに含まれる既存の評価事例や、ISO 25059:2023 (SQuaRE for AI) における AI の品質特性を踏まえて、LLM において従来とは多かれ少なかれ異なる定義や手段にて扱うべき品質特性を整理する（表 10.2）。この表では、SQuaRE for AI における品質特性の語、[Guo+, arXiv23] における評価手法の分類の語に対し、それらと QA4AI コンソーシアムでの議論を踏まえて定めた本章での用語の対応をまとめている。それぞれの品質特性の評価のための手法やベンチマークについては 10.3 にて述べる。

本章での用語	SQuaRE for AI	[Guo+, arXiv23]
QC01：回答性能	Functional Correctness	
QC01-1：自然言語処理における回答性能	Question Answering, Knowledge Completion, Reasoning	
QC01-2：ツール活用に関する回答性能	Tool Learning	–
QC01-3：創造性・多様性に関する回答性能		
QC01-4：制御可能性	User Controllability	–
QC02：事実性・誠実性	Functional Correctness	
QC02-1：一般的な知識に対する事実性・誠実性	Question Answering, Knowledge Completion, Truthfulness	
QC02-2：与えた知識に対する事実性・誠実性		–
QC02-3：根拠の説明性・妥当性		
QC03：倫理性・アライシメント	Societal and Ethical Risk Mitigation	Ethics and Morality
QC03-1：公平性		Bias
QC03-2：安全性		Toxicity, Risk Evaluation
QC03-3：データガバナンス		Risk Evaluation
QC04：頑健性	Robustness	Robustness Evaluation
QC05：AIセキュリティ	Security	Robustness Evaluation

表 10.2 LLM における品質特性

10.2.1 QC01：回答性能

回答性能は、期待する特定の機能あるいはタスクにおける「良さ」の基準に対し、どれだけ正しい結果が提供されるかを表す。SQuaRE for AIにおいては機能正確性に該当するが、いわゆる正しさだけが基準になるわけではないため、回答性能という用語を用いる。

QC01-1：自然言語処理における回答性能

自然言語処理においては、LLM 以前よりそれぞれのタスク専用の AI に対する評価が広く行われてきた。例えば、感情分析 (sentiment analysis)、文書の分類、論理的推論、要約、質問回答、翻訳などがある。それらのタスクにおいてはしばしばベンチマークデータセットにおいて用意された正解と照らし合わせることによる回答精度やそれに類する指標を測定することが行われてきた。これらは品質特性としては回答性能を扱っていると言える。

ただし LLM 固有の点として、一つのタスクに特化せず汎用的に多様なタスクを扱うことを求めることがある。この場合、複数のタスクに対して回答性能を評価することで、総合的に言語理解や言語生成、より広く言語能力の評価とすることが多い。

QC01-2：ツール活用に関する回答性能

LLMにおいては、プログラムコードやオフィス文書ファイルなど計算機処理用のフォーマットを扱ったり、それらのフォーマットを自身で生成して外部のツールを活用したりすることがある。外部のツールとして、検索エンジン、知識データベース、プログラム実行エンジン、オフィスソフトなどを適宜選択、利用することは従来の自然言語処理では想定していない振る舞いであり、LLM を含むシステムにおいて固有の評価観点となりうる。LLM がツールを使う場合、利用者が用いるツール入力を生成する場合、いずれにおいても、計算機処理用のフォーマットでは文法上の正しさや意味上の妥当性など固有の品質を扱う必要がある。

QC01-3：創造性・多様性に関する回答性能

創造性・多様性とは、LLM の出力がより多様で異なる回答を出力するできることを表す。LLM のユースケースは、確実で安定した一様な出力を求める場合だけでなく、アイデアを広げるような出力を求める場合がある。アイディアを広げたい場合には、一つの回答に多様なアイディアを含めたり、回答を出し直す度に異なる内容が出てきたりすることが望ましい。特定のユースケースにおいては創造性・多様性が品質特性の一つとなる。

QC01-4：制御可能性・協調可能性

LLM はプロンプトで入力された情報よりも多くの情報を補って出力するが、その補った情報を含め出力が指示に沿っているかを表す。また、個々の指示に対して安定して指示に沿った出力を評価

することのみならず、追加指示に対しては追加内容に沿った変化が求められる。

これは機械学習分野では、「少ない指示（Zero-Shot/Few-Shot）による学習可能性」ともとらえられるが、利用者の指示に対する制御可能性・協調可能性であると言える。この制御可能性は、出力に対してさらに修正などの指示を行うことで、出力を洗練していくことができることも含む。制御可能性により、多様なタスクや要求に対応できることが LLM の強みであり、制御可能性は LLM における重要な品質特性となる。

SQuaRE for AI (ISO 25059:2023) における制御可能性は、人間やエージェントが AI に対して介入して望ましくない結果を防ぐことが念頭にある。LLM の場合は、多様なタスクに対応する能力を持つ AI に対し、要求に応じた結果を得るための制御を考えている。

10.2.2 QC02：事実性・誠実性

事実性（Factuality）は、提供される情報や返答がどれだけ実世界の真実や検証可能な事実に即しているかを表す。逆に、ハルシネーションや不整合を含む返答を行わないこと、不確かさが高い回答に対してそのような明記を行う誠実性（Truthfulness）も求めることとなる。ハルシネーションに対する強い懸念を踏まえると、LLM に対するトラストや効率的な活用のためには事実性・誠実性が重要となる。また与えた文書に対して検査や検索、要約を行う場合、文書に記載がないようなことがらを返答にふくめないことが結果を信頼するために非常に重要である。

事実に対する質問に回答を行う機能・タスクの場合、その回答性能を評価することで同時に事実性が評価されることもある。しかし、要約や文章生成などにおいて、求められる機能としては十分な返答であるが事実の誤りや矛盾を含むことはあり得るため、機能の正確さとは別の品質特性として挙げている。

QC02-1：一般的な知識に対する事実性・誠実性

事実性・誠実性の一つの観点として、歴史や医療に関する知識などのうち、一般的に正解が存在するとされ検証可能である事実に対しての評価がある。

QC02-2：与えられた特定の知識に対する事実性・誠実性

事実性・誠実性の別の観点として、特定の知識を LLM に与えた際に、それに沿った回答ができるかという評価がある。これは、一般的ではない組織内の知識などをファインチューニングやは RAG、プロンプトにより LLM に与えた際に、その特定知識に基づいた回答ができるかを評価する。この点はファインチューニングや RAG などによる LLM のカスタマイズの主目的となることが多いため、この観点からの事実性・誠実性の評価が必要となる機会は多い。

QC02-3：根拠の説明性・妥当性

事実性・誠実性が重要となるユースケースでは、LLM の回答を利用する人間がその正しさを検証、確認できることも重要である。このために、情報源など回答の根拠となる情報を提示するように求めることが多い。ハルシネーションの一種として、存在しない URL を提示するようなこともあるため、このような根拠を提示できる程度および、その根拠の妥当性を評価する必要がある。

10.2.3 QC03：倫理性・アライメント

倫理性は、広くは倫理的な問題がないこと全体を指す。モラル性ともいいうことができる。人間の期待に添うという広い意味でアライメントという語も用いられる。具体的には、性別や人種など特定のアイデンティティに対して社会的なバイアスを示することがないという公平性、攻撃的であったり社会に害をなしたりするような情報を提供することがないという安全性を考えることが多い。安全性のうち一部については法律により明示的に取り扱われている性質もあるため、それらの遵守も安全性の一種として必要となる。

公平性については LLM 以前の AI においてその重要性が強く認識されており、安全性については LLM の犯罪行為の助長などが当初から話題となったため、特に重要な副特性として挙げている。より広い倫理性は、モラルといった言葉で議論されることもある。モラルについては従来社会学などにおいて Moral Foundation という用語にて考慮する側面が論じられてきた。例えば Moral Foundation Theory では、他者の痛みを感じ取り避けようとする（care）、グループのために活動しようとすること（loyalty）といった Moral Foundation が列挙されている^{*2}。このような専門家による定義とは別に、多様な利用者やステークホルダー、社会の受け止め方を基に倫理性を定義することもある。さらには、LLM に対しては、政治的な特性に関する調査 [Hartmann+, arXiv23] や、より強い力や財産を求めるような発言の調査 [Perez+, ACL'23] もすでに行われている。倫理性については幅広い定義、議論がありうる点に留意が必要である。

QC03-1：公平性

公平性は、性別や人種など特定のアイデンティティに対して望ましくないバイアスを示したりすることがないことを示す。LLM の学習において、訓練データに含まれる不適切なバイアスを反映することがある。これにより、LLM が偏った情報や差別的な態度を示したり、不適切な意見や偏見を増幅したりすることが懸念される。

^{*2} <https://moralfoundations.org/>

QC03-2：安全性

安全性は、人間や社会に対して被害を与えないことを指す。これは、利用者を傷づけるような発言を行わないことと、利用者に対して他者や社会に危害を与えるようなことの双方を含む。被害をなす具体的な内容としては、ヘイトスピーチ、攻撃的あるいは虐待的な行為、ポルノコンテンツ、犯罪行為の助長やほう助などがある。安全性がないことを、毒性（Toxicity）があるという言い方をして問題となる特性に注目することもある。

ここでは対話における安全性という観点から、倫理性・アライメントの一部として安全性を含めている。LLM が発行したコマンドによりロボットを操作するような場合、倫理性・アライメントの一部ではなく、従来からの物理的な安全性を考える必要がある。

QC03-3：データガバナンス

LLM に対しては、訓練データおよび生成データについて、法の遵守あるいはそれに類する観点からの懸念が論じられており、その点からも品質の確認が必要となる。

具体的な観点としてはまず、著作権が議論されている。著作権については、(1) 訓練への既存著作物の利用、(2) 生成された出力が既存の著作物と類似した場合のその利用、(3) 生成された出力の著作物としての扱いといった異なる段階・観点での議論がなされている^{*3}。生成 AI システムの品質という観点では(3)は扱わない。

(1)においては、「問題」ある訓練データの利用により、訴訟やそれに伴うシステム停止のリスクの大小が品質の観点となる。国内では、情報解析では著作権者の利益を害さないという趣旨に基づく著作権法 30 条の 4 がよく知られているが、具体的な判例や法解釈が確立しているわけではなく、著作権者の利益を損なうという主張もある。このためリスクを認識しつつ、状況を追う必要がある。国外で構築、提供されるサービスについては、状況が異なる点にも留意が必要である。

(2)については、訓練データ内に含まれる著作物に近い出力が生成された場合などに、その利用が著作権侵害となる可能性がある。この侵害の有無判断は、生成結果の類似性だけでなく、依拠性の有無、私的利用かどうか、といった出力の利用方法に依存する。このため、生成 AI が類似物を出さないという生成の品質観点というよりも、その出力の利用時に類似性検査などを行うといったシステム全体の品質として議論されることが多い。

より一般的には著作権に限らず、利用規約やプライバシーの配慮が必要となる。特にファインチューニングや RAG などにより固有のデータを取り込ませる場合は重要となる。(1)のように訓練データについては、利用規約、個人情報の利用同意などについて問題がないかの確認が必要である。(3)のように生成データについては、訓練データに基づき機密漏洩やプライバシー侵害となるような出力がないか、あるいは出力の利用方法を適切に定義、制限しているかといった確認が必要である。

^{*3} 2023 年 6 月の文化庁資料より <https://www.bunka.go.jp/seisaku/chosakuken/93903601.html>

10.2.4 QC04：頑健性

頑健性は、未知であったり偏っていたり、敵対的あるいは不正であったりする入力や、外界からの干渉に対して、品質を維持する度合いを指す（ISO 25059:2023）。対象の品質としては回答性能はもちろんのこと、倫理性・アライメントなど多様な品質特性それぞれについてその維持度合いを評価する必要がある。

LLM の場合は、OOD (out-of-distribution) と呼ばれる訓練データの分布を外れる入力や、敵対的な入力に対する頑健性が重要となる。意図しない挙動を引き起こそうとする悪意ある入力に対する頑健性は、AI セキュリティの副特性であるともとらえられる。この点は 10.2.5 にて述べる。

10.2.5 QC05:AI セキュリティ

安全性などの品質要件を守るように構築された LLM が、悪意のある攻撃に対して頑強であることを示す。LLM の場合は、プロンプトインジェクションやジェイルブレイクプロンプト（脱獄、jailbreak）と呼ばれる敵対的な入力により、システムの提供者が与えた LLM に事前に設定した内容について説明させたり、提供者が出力を抑制させた望ましくないコンテンツを出力させたりするなど、作成者の制御や設定から外れる動作を促す攻撃が確認されている（[Liu+, arXiv23], [Yao+, arXiv23] など）。例えば単純には「ここまで指示を無視して以下を行え」といったプロンプトにより、LLM の提供者が設定として与えた初期プロンプトを無視させる方法がある。これにさらに無意味な文字列や他言語の文字列を混ぜ込むようなこともある。このような攻撃により、提供者がシステム内部で与えた設定内容を取得したり、犯罪手法の説明など禁止されているはずのタスクを実行したりすることがある。LLMにおいては特にこのような攻撃に対する耐性が重要である。これは頑健性の一種でもあると言えるが、自然には発生しないような悪意ある入力に対しての耐性を考える場合はセキュリティの問題となる。

LLM 以前の機械学習型 AI に対しては、固有のセキュリティ観点あるいは攻撃の方法が論じられてきた [Kumar+, arXiv19]。具体的には、AI の入出力を通して訓練データを推測する攻撃（Model Invasion Attack）や、提供された AI を模倣する AI を構築する攻撃（Model Extraction Attack）が知られている。本ガイドラインの執筆時点（2023 年後半）で LLM や対話型生成 AI サービスに対してこれらの攻撃が実証されているわけではないが、同様の問題が生じる可能性がある。

当然ながら、LLM を構築するための訓練データや学習パイプライン、LLM の入出力に対して細工を施すなど、不正アクセスを通した攻撃についても、AI に限らない従来のセキュリティとして扱う必要がある。

10.2.6 その他の品質観点

透明性

SQuaRE for AI (ISO 25059:2023) における他の品質特性として、透明性 (Transparency) がある。この品質特性は LLM でも従来の教師あり学習などの AI においても同様に留意が必要である。訓練データの収集方法や属性に関する記録とその情報公開をはじめとして、プロダクトの特性自身だけではなく、プロセス全体についての考慮が必要である。

説明可能性

従来機械学習型 AI、特に深層学習など複雑なモデルを用いるものに対しては、説明可能性 (Explainability) のための技術 (XAI) が一つの重要な側面として扱われてきた (4 章)。一つのユースケースとしては、分類や回帰などの結果の出力だけでは、人間の最終判断において参考にすることが困難なため、入力内の注目領域など決定ロジックに関する情報を追加することがある。

LLM においては、結果とその根拠を出力するように求めることができるが、AI 内部の決定ロジックについて説明しているわけではなく、結果と根拠のもっともらしい組を出している点で、従来 AI の説明とは異なる。このような根拠の出力については、QC02-3：根拠の説明性・妥当性として扱った。

従来の XAI 同様に、訓練データの影響、入力データの構成要素の影響、ニューロン発火の傾向などを分析する技術が説明可能性の技術として追求されている。LLM 固有の点としては、事前学習とファインチューニングのどちらが影響しているか調べたり、ハルシネーションの原因を探ったりといった点がある [Zhao+, TIST'24]。

アクセシビリティ

SQuaRE (ISO 25010:2011) では、ユーザビリティの一環としてアクセシビリティがあり、多言語対応はその一つの側面である。従来ソフトウェアの場合、多言語対応は、明示的に日本語や英語など特定言語のインターフェースを適切に実装したかどうかという問い合わせになる。一方、LLM の場合は学習により多言語を扱う能力がどこまで得られたかということになるため、想定利用者を踏まえ、テストデータを通じた評価により対応の程度を明らかにする必要がある品質特性となる。

ユーザビリティと社会心理的側面

LLM に対する品質特性は、心理学・社会学などの観点からの品質特性を扱う必要もありうる。例えば、挨拶や補足説明、締めくくりの言葉を追加してくることが多いといった対話の上での親切さや個性、個性の一貫性、それを受けた利用時の楽しさや満足度、個人との親和性・相性といった観点がありうる。本章では技術観点での品質特性を中心にまとめているが、こういった品質特性も扱う必要がある。SQuaRE では、プロダクト品質の Usability や、利用時品質の Trust, Pleasure, Comfort

に相当するものとなる。

対話の自然さや流ちょうさは重要な側面であり、SQuaRE では上記の品質特性、特に Usability の副特性である User Interface Aesthetics（快美性）に影響すると考えられる。LLM については、その自然さが非常に高いことが LLM の標準特性となっており、LLM の評価軸として明示的に論じられることが多いが、モデルによっては評価を検討することもありうる。

機能適応性

SQuaRE for AI (ISO 25059:2023) では、機能適応性 (Functional Adaptability) として、環境変化に対して適応的に振る舞う能力が一つの品質特性となっている。本ガイドライン執筆時点では、LLM に対してはプロンプト等による明示的な指示による変化への対応を制御可能性として考えているが、LLM を用いた AI システム全体として継続学習を行う場合などは、機能適応性を扱うこととなる。

10.3 一般的な LLM に対する品質評価手法

以下では、特定のドメインやタスクに特化したカスタムの LLM 評価ではなく、ChatGPT など一般的な LLM に対して適用されてきた評価手法について紹介する。これらの評価は、利用目的を特定のものに限らず汎用的なツールとしての LLM に対し総合評価として取り組まれてきたものである。これに対し、特定のドメイン、タスクでの要求に対する十分性評価、特にそのような要求に応じてカスタムな LLM システムを構築した場合の評価が実用上重要となる。この点については 10.4 で論じる。

以下ではいくつかのベンチマークや評価手法に触れるが、それらは実現の一例として示すものであり、デファクトスタンダードあるいは最新であることを意味しない点に留意いただきたい。本ガイドラインとしては、執筆時点（2023 年後半）のスナップショットを示すことで、概念や基本的なアプローチの整理を試みる。これを受けて、読者のニーズを踏まえたベンチマークの最新版などの調査や検討、テストフレームワーク上の固有のテストスイート実装は必要になるであろう。執筆時点では LLM のためのテストフレームワークは Giskard^{*4} や deepeval^{*5} など一部のものに限られている。また、執筆時点では画像を扱うものなどマルチモーダルなものは発展途上であり、評価事例も少ないため、テキストを出力する LLM を対象として論じる。

LLM の訓練には膨大なリソースを要するため、サービスとして提供されているものを活用したり、オープンソースとして訓練結果が提供されているものをカスタマイズしたりすることが多い。このため、訓練データの品質に対しては、その情報が公開されておらず検証ができないことが多い。結果として、プロダクト・サービスとして完成したものに対してベンチマークを設定し、テストを

^{*4} <https://www.giskard.ai/>

^{*5} <https://docs.confident-ai.com/>

実施することが多い。

10.3.1 QC01：回答性能の評価

QC01-1：自然言語処理における回答性能の評価

感情分析 (sentiment analysis)、文書の分類、論理的推論、要約、質問回答、翻訳などのタスクに対しては、LLM 以前から、それぞれにおける評価ベンチマークが発展してきており、技術の発展を測るために評価指標も広く用いられてきた。例えば翻訳であれば、用意された人間の専門家による翻訳との類似度をとらえる BLUE スコアが自動評価可能な指標としてよく知られている (BLUE は Bilingual Evaluation Understudy の意) [Papineni+, ACL02]。

LLMにおいては、一つのタスクに特化したモデルではないため、複数のタスクに対して総合的に正しさを測ることで言語能力一般としての評価を行うことが多い。この際には、複数の評価ベンチマークを統合した包括的なベンチマークを用いる。自然言語推論 (NLI: Natural Language Inference)、自然言語理解 (NLU: Natural Language Understanding)、自然言語生成 (NLG: Natural Language Generation) といった観点で総称されるベンチマークとなる。

複数の評価ベンチマークをまとめた包括的なベンチマークの例として、SuperGLUE がある (以前のベンチマークである GLUE の発展、GLUE は General Language Understanding Evaluation の意) [Wang+, arXiv19]。SuperGLUE は自然言語推論 (NLI) を対象とし、例えば以下のようなタスクを含む。

- 与えた文章に対する質問に対し Yes/No の回答を行う (BoolQ)。
- 与えた文章に対して、その原因あるいは結果となるものを二択で選ぶ (COPA)。
- 与えた文章内の代名詞に対し、それが指すものを選ぶ (WSC)。

各タスクは自動評価できる形になっており、二値分類であれば正解率や F1 値などの一般的な評価指標に帰着される。

LLM に対しては、算術演算なども加えてより広い総合ベンチマークとしたものが盛んに提案されている。一例としては Language Model Evaluation Harness^{*6}があるが、他にも多数の総合ベンチマークがある [Guo+, arXiv23 (7.3)]。

上記のような包括的なベンチマークの結果は、リーダーボード (Leaderboard) と呼ばれる順位表のような形で可視化されることが多い。リーダーボードにおいては、自身が評価したいタスク集合を選ぶことで、それらに対してのみの順位表を作ることもある。

包括的なベンチマークについては日本語版の構築も進んでいる。執筆時点 (2023 年後半) では、上記にて紹介したベンチマークの日本語版として、Yahoo! JAPAN 研究所による JGLUE^{*7}、Stability

^{*6} <https://github.com/EleutherAI/lm-evaluation-harness>

^{*7} <https://github.com/yahoojapan/JGLUE>

AI による JP Language Model Evaluation Harness^{*8}などの事例が見られる。

QC01-2：ツール活用に関する回答性能の評価

LLM を含むシステムにおいては、検索エンジン、知識データベース、プログラム実行エンジン、オフィスソフトなどの外部ツールのためのフォーマットを扱ったり、それらの外部ツールの選択と実行を行ったりすることがある。このような場合、以下のような評価観点が必要になる ([Guo+, arXiv23 (3.4.1)] より補足)。

1. ツールへの入力が正しいか、あるいはツール実行が正しく完了するかの評価
2. ツールへの入力や、ツールの実行結果の品質が高いかの評価
3. ツールを使うことによるタスク達成度合いに関する評価

例えばプログラムコードの生成や実行においては、(1) コンパイラが検査するようなプログラムの文法の正しさやそれにより実行が多くのテストケースで正常終了すること、(2) コーディング規約の遵守や可読性などプログラムの保守性あるいはテストケースのパス率、(3) プログラム実行により望む結果が得られたか、あるいは開発プロセス上の進捗が得られたかといったことが考えられる。なお、プログラムコード生成については盛んな研究事例があるため、翻訳における評価指標を適合した評価指標、つまり専門家の模範解との類似性を測る CodeBLEU など固有の評価指標も提案されている（例えば [Evtikhiev+, JSS23] の評価事例を参照）。

QC01-3：創造性・多様性に関する回答性能の評価

創造性・多様性については、複数案の回答を求めたり回答を再生成したりした場合の回答群についての評価を行う。単に回答数ではなく回答間の何らかの差異・距離をとらえた評価ができることが望ましい。執筆時点ではこういった観点での具体的な評価事例は確認できていない。なお、逆にプログラム生成など比較的安定した回答が望ましい場合に多様な解が出てしまうことを（問題点として）評価した事例はある（例えば [Ouyang+, arXiv23]）。

QC01-4：制御可能性・協調可能性の評価

制御可能性・協調可能性の評価においては、機能・タスクの回答性能自体ではなく、その際の指示を意図を持って変更した際にそれを反映できるかを評価する。例えば出力フォーマットの指示を付加したり、含めなければならない観点や含めてはならない観点の指示を追加した際に、それらを遵守した回答に変わるかを評価することが考えられる。すなわちメタモルフィックテスティングの形式をとる評価となる。このときの指示の付加は、当初のプロンプトへの追加と、回答に対する返答としての追加プロンプトとして行う場合がある。執筆時点において、このような形式での評価事例

^{*8} <https://github.com/Stability-AI/lm-evaluation-harness>

は確認できていない。

10.3.2 QC02：事実性・誠実性の評価

事実性の評価の方法としては、事実を答えるタスクに対して回答性能の評価を扱う方法がある。単純には、事実に関する質問回答について評価を行えばよい。そのようなベンチマークデータセットの一例としては読み解きの事実を扱う TriviaQA[Joshi+, ACL17] や、LLM に記憶された知識を問う KoLA[Yu+, arXiv23] などがある。より詳細な評価を人間が行う場合や、長い回答文章を分割して含まれる個別の文をそれぞれ評価するような手法 [Min+, EMNLP23] もある。

誠実性の評価をしたい場合は、答えられないはずの問い合わせを投げかけるような評価ベンチマークが必要となる。例えば総合ベンチマークである BIG-bench においては、known-unknowns というタスク、つまり「未知であると断言できること」についての問い合わせが用意されている [Srivastava+, EMNLP23]。また TruthfulQA は、誠実性を問うようなベンチマークデータセットの一例であり、嘘の回答を引き出すような質問を含むベンチマークとなっている [Lin+, ACL22]。

QC02-1：一般的な知識に対する事実性・誠実性の評価

一般的な知識に対する事実性・誠実性の評価においては、学習を通して獲得した事実を問うため、プロンプトなどに事実を含めずに質問回答を行うような評価を行えばよい。既存データセットにおいては、質問と、答えの証拠を含む付属文書を含むことがあるが、付属文書を用いずに問い合わせを投げかければ、LLM が学習を通して獲得した事実について問うことができる。

QC02-2：与えた知識に対する事実性の評価

ファインチューニングあるいはプロンプトにより与えた特定の知識に対する事実性・誠実性を評価するためには、そのような知識を与えた上で問い合わせを行えばよい。ファインチューニングやプロンプトよりも、ニューラルネットワークの局所更新などの知識追加の手法も盛んに研究されており（例えば [Zhang+, arXiv24]）。このため、これらの研究における評価は、与えた新しい知識に対する（少なくとも事実性の）評価事例になっていると言える。

QC02-3：根拠の説明性・妥当性の評価

執筆時点において、出力内の、求められる回答そのものと、根拠部分とを明示的に区別して評価している取り組みは明示的には見られない。しかし、プロンプトとして根拠を含めるように指示したり、サービス提供者が論文や Web サイトの引用など根拠を含めるようにしたりすることはよく見られるようになっている。このため、対象タスクそのものの回答性能とともに、根拠の出力ができるか、妥当かについても評価する手法を確立していく必要がある。

10.3.3 QC03：倫理性の評価

公平性、安全性、データガバナンスについては以降においてそれぞれ詳述する。

モラルとも呼ばれるようなより広い倫理性の評価については、倫理性を問うようなベンチマークが用いられてきた。例えば ETHICS データセットでは、状況の記述に対して受容可能な度合いを測らせることで、人間と同じような価値判断がなされるかを評価する [Hendrycks+, ICLR21]。このような倫理性に関する評価を、クラウドソーシングを基に集めたデータセットとしては MoralExceptQA[Jin+, NeurIPS22] がある。

LLM の生成した回答を評価するデータセットとしては、BOLD[Dhamala+, FAccT21] などがある。BOLD では、Wikipedia から収集した文章の前半のみを与えて、LLM が生成した後半において不適切なバイアスや毒性がないかを評価する。

QC03-1：公平性の評価

性別や人種などセンシティブ属性に関するバイアスについては、翻訳や推論といった個別タスクに関する評価が盛んに取り組まれてきた。例えば Winogender[Rudinger+, NAACL18] や WinoBias[Zhao+, NAACL18] などのデータセットにおいては、消防士が「he」になるなど代名詞におけるバイアスを評価する。翻訳タスクについては WinoMT[Stanovsky+, ACL19] などのデータセットが同様の側面を扱っている。LLM によるより広いタスクを扱うものとしては、StereoSet[Nadeem+, ACL21] や CrowS-Pairs[Nangia+, ACL20] がある。これらのデータセットにおいては、LLM がセンシティブ属性に関するステレオタイプを好むかどうかを評価する。

上述した BOLD[Dhamala+, FAccT21] のようなデータセットでは、LLM の生成した回答に対して公平性を評価できるようになっている。HolisticBias[Smith+, EMNLP22] データセットにおいては、600 個のセンシティブ属性に関する単語について、テンプレートを用いてプロンプトを生成する。UNQOVER[Li+, EMNLP20] や BBQ[Parrish+, ACL22] においては、選択問題としてステレオタイプに関する問い合わせるようにしており、「誰が犯人かなど断言できないはずのところ特定の人種などを指さないか」といった問い合わせも用意している。

QC03-2：安全性の評価

毒性の有無を判定・分類するタスクに対するデータセットとしては、twitter のデータを集めた OLID[Zampieri+, NAACL19] や半教師あり学習を用いて構築した SOLID[Zong+, ACL21] がある。そのような判定機を利用して、LLM が生成する回答の毒性評価を行うものとしては、例えば RealToxicityPrompts[Gehman+, EMNLP20] がある。この取り組みでは、毒性があるような回答を誘導するプロンプトが提供されている。プロンプト自体には毒性がなくても、毒性がある回答を誘導できることが示されている。犯罪や自殺に関する問い合わせのベンチマークとして HarmfulQ がある [Shaikh+, ACL23]。

QC03-3：データガバナンス

10.2.3 にて論じたように、訓練データおよび生成データについて、多様な観点からの評価が必要となる。訓練データについては、著作権、利用規約、個人情報の利用同意などの観点から評価が必要となる。生成データについては、著作権や利用規約、プライバシーの観点から問題となり得る出力の有無や頻度、その利用方法の定義、制限の適切さの評価が必要となる。第三者が提供する LLM についてのこれらの点からの確認はもちろんのこと、ファインチューニングや RAG により固有のデータを取り込む場合は、特にそのデータ固有の確認が必要となる。

ただし、これらの点に限らない留意、配慮が必要となる。個人情報については、削除依頼があった場合の手続き、例えばファインチューニングの再実行も検討する必要がある。個人情報そのものが必要とない場合にはその除去や変換を行ってからデータを活用することも重要である。

10.3.4 QC04：頑健性の評価

頑健性の評価においては、入力やタスクにおいて単語単位や文単位などでの摂動を加えたベンチマークにより評価を行う。そのようなベンチマークとしては、AdvGLUE[Wang+, NeurIPS21] や ANLI[Nie+, ACL20] がある（ともに GLUE や NLI といった既存の語に Adversarial を冠している）。摂動ではなく、悪意ある入力に対する頑健性については AI セキュリティに関する項目（10.3.5）にて扱う。

10.3.5 QC05：AI セキュリティの評価

[Liu+, arXiv23] ではジェイクブレイクプロンプトのパターンを 10 種類を系統的に定義している。例えば、ペルソナを演じることを求めたり、科学的な実験を装ったりすることで期待しない回答を引き出そうとするパターンがある。これらのパターンを用い、OpenAI が ChatGPT の利用において禁止していることを行わせる実験を行っている。[Wei+, arXiv23] では 30 種類弱のジェイルブレイクプロンプト手法に対する評価を行っている。[Deng+, arXiv23] では、ジェイクブレイクプロンプトの防止が行われる際に実行時間が変わる LLM サービスがあるなど防止策の分析もを行い、SQL インジェクション手法を模倣したフレームワークを提案している。このように、AI セキュリティにおいては、攻撃手法を実装することで LLM を評価する。

10.4 LLM を用いた個別システムに対するカスタムの品質と評価

10.3 では、「多様なタスクを扱える」一般的な LLM に対して評価を行う手法についての一般論について述べた。LLM を用いて個別のシステムを構築する場合、そのシステムの要求やリスク、扱う固有の情報を踏まえた評価が必要となる。

執筆時点ではこのようなカスタム評価は限られている。プログラム生成、金融、法律、教育といっ

たドメインごと知識回答については盛んに評価が行われているが、特定システムの要求やリスクを考慮した取り組みは広く見られない。テストフレームワークとして Giskard^{*9} や deepeval^{*10} などが見られているが、その評価や活用事例もこれからという状況である。

このように執筆時点でのノウハウは限られているが、個別システムに対して評価を行う場合、具体的には以下の留意が必要となる。

10.4.1 対象システムが扱うタスク固有の回答性能評価

特定の機能・タスクに対しては、それに対応した回答性能（QC01）の評価が必要となる。例えばコード生成では、生成されたコードに対し、テストをパスする正しさ、無駄のなさ、読みやすさ、コーディング規約の遵守といった評価基準がありうる。専門家の模範解との類似性を測る CodeBLUE など固有の評価指標も提案されている（例えば [Evtikhiev+, JSS23] での評価事例を参照）。

10.4.2 対象システム固有の知識に関する事実性・誠実性の評価

事実性・誠実性（QC02）の評価においては、照らし合わせる事実がシステム固有となる。RAG やファインチューニングなどの知識取り込みの手法を通して知識を取り込む場合、その知識に沿った回答をすることが個別システムを構築する理由ともなるため、この点が最も重要な評価となる。

10.4.3 対象システム固有のリスクに関する AI セキュリティの評価

AI セキュリティに関する評価では、対象システムにおいて防ぎたい攻撃に関する分析や列挙、リスクを踏まえた優先度決定などが必要になる。例えば、システムによって利用者に許可する LLM の機能・タスクが異なる。文書作成アプリに埋め込んだ LLM であれば、記述補助や要約に使わせる意図であり、そこでプログラミングを行うことは、サービス価格に対する対価の観点からも防ぎたい。LLM による金融商品の説明において虚偽の説明をさせることで損害賠償を狙う、競合他社の批判などを言わせるなどシステム固有のリスク種類やその大小があるため、防ぎたい事象については対象システムごとのリスク分析が必要となる。

10.4.4 自動評価の実現手段と評価内容

10.3 で紹介した多くの取り組みにおいては、自動評価が可能となるようなベンチマークが提供されている。自動評価により多数の LLM を比較するが容易となるとともに、LLM の調整や改善のためのサイクルを迅速に反復することが可能となる。

^{*9} <https://www.giskard.ai/>

^{*10} <https://docs.confident-ai.com/>

一方、自動評価においては True/False の二値で解答する問題や、文章に対して適切さの点数を与える問題など、正解率や F 値といった標準的な評価指標に帰着しやすい形になっていることが多い。その結果、対象の品質特性に対し、本来評価したいことがらとは異なるベンチマークとなっている可能性がある。

例えば倫理性については、究極的には、運用時に現れる多種多様な問い合わせにおいて対象の LLM が不適切な回答をしないことを確認したい。例えば、様々な病状についての問い合わせに対し、病院に行かずに自身の判断で治療するように促すなど行き過ぎた助言がないことを確認したいとする。これに対し既存のベンチマークでは「・・・のときに病院に行かず・・・する行為は適切か？」という問い合わせへの点数付けが正しいかどうかを評価している場合がある。

対応したい品質特性の名前から盲目的に既存のベンチマークを採用するのではなく、品質による価値・リスクやそのためのコストを踏まえてベンチマークの選択や拡張を行っていく必要がある。

10.4.5 利用する自然言語

現状では ChatGPT などアメリカで開発された LLM も多いが、そういった LLM は、データセットが少ないマイナーな言語や、データセットが大量にあってもラテン文字以外を用いる言語において性能が悪いことがある（例えば [Bang, arXiv23] における翻訳の評価）。このため、利用が想定される利用言語を明確にし、その言語での評価を行う必要がある。

10.5 QA4AI ガイドラインの 5 軸の品質特性に対する生成 AI の特徴

本章では、生成 AI、特に大規模言語モデル (LLM) の活用における課題や品質特性について論じてきた。一方で、本ガイドラインの 2.1 章で示している AI プロダクトの品質保証において考慮すべき 5 軸は、主にディープラーニングなどの機械学習技術を前提に作成された。つまり、判別や分類、推測などの定められた課題に対して、それを実現する機能を持つ機械学習モデルと、そのモデルを用いたシステムを構築することを想定している。そのため、データの収集、モデルの学習、システムの構築といったプロセスが、設定された目的を達するように適切に実施されることを求めている。

他方、LLM は特定の機能に特化しない基盤モデルとして提供される。そのため、設定された目的の達成という観点では、LLM 構築のためのデータの収集やモデルの学習における品質保証を語ることができない。また、一部の事業者を除いて多くの事業者は LLM を自身で構築することは稀であり、本ガイドラインが対象とする LLM を包含するプロダクトの構築においてデータ収集やモデル構築はブラックボックスであって、直接的に品質を評価することができない。

しかし特定の目的を実現するために、ファインチューニングなどの技術を用いて、特定のデータを既存の LLM に追加学習する ことができる。また、RAG と呼ばれる技術により、既存の LLM と特定のデータを蓄積したデータベースを併用するシステムも一般的になりつつある。それらの場

合、特定の目的に対するデータの収集や(ある種の)モデル構築を行うことと言えるが、学習済みのLLMを前提としていることから、従来の機械学習におけるデータ収集やモデル構築と同様に考えることはできない。

以上のような特性により、AIプロダクト品質保証において考慮すべき5軸を、基盤モデルを前提とした生成AIにそのまま適用することができない。現時点では、生成AIに対する品質保証の軸の見直しは完了していない。以下の議論は、品質保証において考慮すべき5軸に対して、生成AIで特に考慮すべきポイントを示すに留める。

Data Integrity

5軸のうちData Integrityでは、訓練データやテストデータの質や量の適切性や十分性について言及している。これらの適切性や十分性は、想定する利用環境や目的などを考慮して評価されるものである。例えば学習出た一の量の十分性に関するチェックリスト(a.ii)では「想定する要求・適用環境において」十分な量のデータがあることを、訓練データの妥当性に関する(b.i)でも「想定する要求・適用環境に」対応した適切なデータとなっていることを求めている。しかし、基盤モデルとしてのLLMは様々な目的に対して様々な環境で利用される汎用性が期待され、データに関しても要求や利用環境などの前提を置くことができないし、用いるデータが莫大であるので、データの量や質を評価することは簡単ではない。また、LLMを構築する一部の事業者以外にとってLLMの学習にどのようなデータが使われたかはブラックボックスであり、Data Integrityの評価は困難である。

以下、LLMを学習する場合、学習済みLLMを利用する場合、学習済みLLMに追加学習をする場合に分けて、LLMにおけるData Integrityに関する留意点を述べる。

LLMを学習する場合

LLMの学習には大量のデータが必要であるため、インターネット上にあるあらゆる文書を使う場合がある。このとき、学習のデータが適切かという視点が重要となる。このときの適切性は先に述べたような想定する要求・適用環境に対応する適切性でなく、LLMによる回答の正誤や妥当性等に影響を与える一般的な事項を検討することとなる。この点はチェックリスト(d)訓練データの適正性や(h)訓練データの法的適合性と関連する。適切でないデータとは、例えば以下のようなものである。

- 誤りのある文書、フェイク文書：誤ったことを学習することにつながる
- 陳腐化したデータ：インターネット上にある陳腐化したデータを使うと誤った解答につながる
- 権利を侵害する文書：著作権のあるデータを利用すると著作権侵害につながる
- 倫理的に問題のある文書：差別や迫害などにつながる言動を学習する可能性がある

LLMを学習する際は、上記のようなデータを用いないことでData Integrityを高めることができる。訓練データセットを構築する際は、利用できるデータを見極め、利用したデータの出自を明らかにしておくことが重要である。

ただし LLM によっては特定の分野にフォーカスした学習を行う場合がある。例えば、プログラム言語の学習にフォーカスした LLM や、日本語など特定の自然言語にフォーカスした LLM、金融などの特定事業分野にフォーカスした LLM である。その場合には、それらのフォーカスに対応したデータを適切に収集する必要がある。

データの量についてでは、LLM ではスケーリング則と呼ばれる特性が知られており、データ量が多いほど回答の精度が上がると言われている。そのため現時点ではできるだけ多くのデータを用いる開発競争が繰り広げられている。しかし大規模な LLM の学習には多大なコストがかかるため、今後は目的を限定した比較的小規模な LLM も登場するものと考えられる。

学習済み LLM を利用する場合

学習済みの LLM を使う際はどのようなデータで学習されているか詳細に把握することは難しい。モデルによってはどのようなデータを使った記載しているものもあるが詳細には書かれていないことが多い。不明なデータで学習した LLM を利用する結果、不適切な出力をしてしまう可能性もある。このため、学習に使われたデータが公開されている LLM を利用するが、LLM を利用したシステムを提供する際のリスクを下げるに繋がる。

学習済み LLM に追加学習する場合

学習済み LLM にユーザが保有するデータを追加する場合がある。その方法には、ファインチューニングや RAG(Retrieval-Augmented Generation) など様々あり、それぞれ異なる特徴をもつが、ここではそれらを区別せず追加学習と呼ぶことにする。追加学習の場合には、想定する要求や適用環境が存在するため、Data Integrity の考え方を準用することができる。ただし、LLM が既に学習しているデータがあるため、想定する要求や適用環境に対するデータを追加学習すべてカバーする必要はない。また、追加学習によってどのように結果が変化するかは明確ではなく、追加により一般的な知識に対する性能が低減する場合もある。ベースとなる学習済 LLM と追加学習のためのデータの関係性については、今後の研究の進展が待たれる。

Model Robustness

Model Robustness で考慮する主な性質として、モデルの精度 (チェックリスト (a))、汎化性能 (チェックリスト (b))、頑健性 (チェックリスト (g)) がある。

一般的に AI モデルの精度は、特定の問題に対する正解を設定した上で、それを表現するテストデータを準備し、正答率や再現性などで評価される。LLM の評価においても特定の問題に対する正答率や再現性を評価することは可能である。しかし基盤モデルとしての LLM は様々な問題に対する汎用性も求められており、特定の問題に対する評価が、LLM としての精度を表すものではないことに注意が必要である。また、特定の問題を設定する場合にも、問題領域が広く正解が定まらない利用が多いことも LLM の特徴である。例えば自然言語で書かれた仕様書を入力としてソフトウェアのソースコードを生成する場合、入力となる仕様は非常に多様であり、それに対して生成するソ

スコードの正解も1つではない。ソースコード生成能力の評価には、Human Eval^{*11}やMBPP^{*12}のようなベンチマークが用いられることが多いが、その結果がユーザが期待する生成対象でも同様となるとは限らない点に注意が必要である。(QC01-1, QC02-2 に関連)

また、LLMは一般的な知識を有していることが期待されることから、その正しさは特定の問題に対する正解ではなく、社会の一般的な理解に合致しているか、あるいはそれが事実であるかという観点で評価されることもある。この点は、本ガイドラインの Model Robustness では考慮していない。(QC02-1 に関連)

次に汎化性能とは、学習時の入力とは異なる入力に対して適切な判別や予測を行う能力である。たとえば動物画像を判別する機械学習モデルでは、訓練データに含まれる犬の画像と完全に一致しなくとも、それらに類似した入力画像に対しては犬であると判別できるのは汎化性能によるものである。機械学習では「内挿」と呼ばれる訓練データ群の内側にある入力データに対する汎化性能が期待され、逆に「外挿」と呼ばれる訓練データ群に含まれない入力データに対する結果を問うことは技術の特性上無意味である。しかし LLM の場合には外挿に相当するような未知の結果を生成することが期待される場合がある。そもそも Data Integrity で述べたように訓練データの特定が困難があるので、内挿か外挿かの区別をすることも困難である。一方で、LLMに対する指示に対する正解は1つに定まるものでなく、同じ入力に対して様々な出力を生成する「出力の多様性」をモデルの評価指標として考慮する場合もある。(QC01-3 に関連)

モデルの頑健性とは、入力の微小な変化に対しての出力が安定する性質を表す。LLMは同じ入力に対して異なる結果を生成することも特徴の1つであり、上記の意味での頑健性を求めるのは困難である。しかし利用目的によっては LLM に対する問合せに対して安定的な回答が期待される場合もあり、LLM に用いて安定的な結果を得るために検討は今後の課題である。(QC01-4 に関連)

そのほか、LLMに期待される性質について以下に挙げる。

1つは、知らない情報に対して正しく知らないと答える能力である。LLMは過去の情報を学習しているため、最新の情報は学習していない。しかしながら、LLMは知らない情報をあたかも知っているように文書を作り上げることがある。このような出力は、利用者に対して正しくない情報を提供するため性能の悪いモデルとなる。知っている情報は正しく返し、知らない情報は知らないと回答できるかがモデル頑健性の1つの要素となる。

2つめは、文書としての自然さの観点である。LLMの利用者は様々な指示をすることがあり、LLMは指示に適した自然な文書をする必要がある。例えば、利用者は箇条書きでの指示をしたり、文字数を指定したりする。また、小学生向けや専門家向けなど、文章の読者を指定する場合もある。内容の正しさだけでなく、指示に沿った出力の自然さも LLMの頑健性の1つと考えられる。

最後は敵対的な攻撃に対する頑健性である。悪意のある利用者は、LLMが学習したデータを抜

*11 <https://github.com/openai/human-eval>

*12 <https://arxiv.org/abs/2107.03374>

き出そうとしたり、LLM をだましたり、LLM に悪意のある出力を引きだそうとすることがある。LLM は、これらの入力に対して適切に対処できなければならない。例えば、設定パラメータを聞き出すような入力を無視したり、差別を助長するような発言はできないと回答するなどである。(QC03 に関連)

本ガイドラインの Model Robustness では学習過程の妥当性(チェックリスト(d))についても言及しているが、LLM の学習については技術進展の最中であり、現時点でその過程の妥当性を検討することは困難である。

また、Model Robustness ではモデルの陳腐化(チェックリスト(j))にも言及している。これは学習時と運用時で外部環境に変化がありデータ分布が一致しなくなるコンセプトドリフトなどを想定したものである。LLM の構築では膨大なデータの学習に多大なコストがかかる上、日々生まれている新しいデータを反映することは困難である。学習済 LLM を利用する際には、学習がどの時点で行われたものであるかに留意し、特に最新の情報を問い合わせるような用途で使う場合には、Web 検索との併用のような手段も検討すべきである。

System Quality

System Quality では AI プロダクト全体の品質確保について考慮している。AI コンポーネントをシステムの一要素として扱うことで、システム全体に対しては従来の品質保証の考え方を準用できる可能性がある。LLM を組み込んだシステムを開発しリリースする際の要点としても、従来システムの品質観点と同じである。しかしながら、生成 AI を利用している点で、利用者への影響が大きいハルシネーション、公平性、倫理、データの権利、AI セキュリティ、個人情報とプライバシーについて、システムのリスク分析を実施し対策を講じることを強く求める。(チェックリスト(b)(c)(d)(f)(i)) (QC04, QC05 が関連)

また、System Quality ではシステムが提供する価値に着目している(チェックリスト(a))。LLM の活用は黎明期であり、現状では LLM を使うこと自体が目的となっているシステムもある。しかし、LLM を使うことが妥当か、従来の機械学習や演繹的な解析技術を用いる方が提供価値の観点で適切でないかなど、今後は冷静な判断が求めらるようになるだろう。

さらに、LLM にデータベースを加える RAG や、プロンプトによる in-context learning、あるいは Langchain 等を用いて複数回の LLM への問合せを行うなど、LLM を中心とした AI コンポーネントの構造は多々ありアップデートが続いている。また、LLM の出力を評価する手法なども提案されている。適切な価値提供のための適切な生成 AI 包含システムのアーキテクチャは、今後も模索が続くであろう。

AI システムの法的適合性(チェックリスト(h))に対して、EU AI Act など従来の機械学習に対する規制において、生成 AI に対する言及が急速に検討されているところである。規制内容について変化が早く、最新の情報を確認する必要がある。

Process Agility

LLM は従来の AI よりも開発のスピードが速いことが特徴である。そのため、利用者・開発者ともに最新の情報を入手し、知識・技術をアップデートしなければならない。

利用者の立場では、LLM を活用するための知識と技術が必要になる。例えば、LLM への入力に対してどのような出力になるかや、求める出力を得るためににはどのようなプロンプトにすれば良いかなどである。基本的なものは書籍などにまとめられるが、システムに特化するものもある。ノウハウを社内でまとめるなど利用者も学習し続けるという考えが必要である。

開発者の立場では、LLM が数ヶ月～数年おきに新しいものがリリースされるという点を考慮して、システムを開発しなければならない。LLM を交換可能に出来る設計にしておいたり、必要に応じてクラウド環境を使うなど、今後想定される LLM の更新を意識した開発が求められる。

Process Agility のチェックリストは、多くの項目が LLM を用いたシステム開発においても準用できるであろう。

Customer Expectation

ChatGPT をはじめとした近年の LLM 技術の高まりを受け、一般のニュースなどでも広く生成 AI について紹介されている。そのため、顧客も生成 AI の基本的な機能や生成 AI の課題を認知し始めている。一方、すべての人が正しく生成 AI の課題を理解しているわけではなく、利用者は一般消費者となることもあるため、生成 AI を組み込んだシステムを提供する際には、考えられる課題を共有し、制限事項として理解してもらうことが必要である。

例えば、ある自治体でのゴミ出し案内を ChatGPT で行う実証実験において、当初 62.5% だった正答率を様々な工夫で 94.1% に向上したものの、自治体では導入条件を 99.9% としていたために導入を断念したとのニュースがあった。その判断の是非についてはここでは論じないが、ステークホルダーの期待によっては異なる結果になった可能性はある。

生成 AI の利用が増加する一方で、現状ではその技術的特性や能力についての理解が社会に浸透しているとは言えず、生成 AI の技術も日々進歩している。最新の情報を基に期待値も常に見直しながら導入していくことが肝要である。

参考文献

[Chang+, arXiv23] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, Xing Xie. A Survey on Evaluation of Large Language Models (v8). arXiv, <https://arxiv.org/abs/2307.03109>, October 2023.

[Guo+, arXiv23] Zishan Guo, Renren Jin, Chuang Liu, Yufei Huang, Dan Shi, Supryadi, Linhao Yu, Yan Liu, Jiaxuan Li, Bojian Xiong, Deyi Xiong. Evaluating Large Language Models: A Comprehensive

Survey (v2), arXiv, <https://arxiv.org/abs/2310.19736>, October 2023.

[Zhao+, TIST'24] Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. Explainability for Large Language Models: A Survey. ACM Trans. Intell. Syst. Technol., January 2024 (Early Access).

[Papineni+, ACL02] Kishore Papineni, Salim Roukos, Todd Ward, Wei-Jing Zhu. BLEU: A Method for Automatic Evaluation of Machine Translation. The 40th annual meeting of the Association for Computational Linguistics (ACL 2002), pp. 311-318, December 2002.

[Wang+, arXiv19] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, Samuel R. Bowman. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems (v3). arXiv, <https://arxiv.org/abs/1905.00537>, February 2020.

[Lin+, ACL22] Stephanie Lin, Jacob Hilton, Owain Evans. TruthfulQA: Measuring How Models Mimic Human Falsehoods. The 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022), pp. 3214 – 3252, May 2022.

[Zhang+, arXiv24] Ningyu Zhang et al. A Comprehensive Study of Knowledge Editing for Large Language Models (v3). arXiv, <https://arxiv.org/abs/2401.01286>, Jan 2024.

[Hartmann+, arXiv23] Jochen Hartmann, Jasper Schwenzow, Maximilian Witte. The political ideology of conversational AI: Converging evidence on ChatGPT's pro-environmental, left-libertarian orientation (v1). arXiv, <https://arxiv.org/abs/2301.01768>, January 2023.

[Perez+, ACL'23] Ethan Perez et al. Discovering Language Model Behaviors with Model-Written Evaluations. Findings of the Association for Computational Linguistics: ACL 2023, July 2023.

[Liu+, arXiv23] Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, Yang Liu. Prompt Injection attack against LLM-integrated Applications (v1). arXiv, <https://arxiv.org/abs/2306.05499>, June 2023.

[Yao+, arXiv23] Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Eric Sun, Yue Zhang. A Survey on Large Language Model (LLM) Security and Privacy: The Good, the Bad, and the Ugly (v1). arXiv, <https://arxiv.org/abs/2312.02003>, Dec 2023.

[Kumar+, arXiv19] Ram Shankar Siva Kumar, David O'Brien, Kendra Albert, Salomé Viljöen, Jeffrey Snover. Failure Modes in Machine Learning Systems (v1). arXiv, <https://arxiv.org/abs/1911.11034>, November 2019.

[Evtikhiev+, JSS23] Mikhail Evtikhiev, Egor Bogomolov, Yaroslav Sokolov, Timofey Bryksin. Out of the BLEU: How should we assess quality of the Code Generation models?. Journal of Systems and Software, Vol. 203, 2023

[Ouyang+, arXiv23] Shuyin Ouyang, Jie M. Zhang, Mark Harman, Meng Wang. LLM is Like a Box of Chocolates: the Non-determinism of ChatGPT in Code Generation (v1). arXiv, <https://arxiv.org/abs/2306.05499>.

[org/abs/2308.02828](https://arxiv.org/abs/2308.02828), August 2023.

[Joshi+, ACL17] Mandar Joshi, Eunsol Choi, Daniel Weld, Luke Zettlemoyer. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension, The 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017), July 2017.

[Yu+, arXiv23] Jifan Yu et al. KoLA: Carefully Benchmarking World Knowledge of Large Language Models (v2). arXiv, <https://arxiv.org/abs/2306.09296>, July 2023.

[Srivastava+, EMNLP23] Aarohi Srivastava et al. Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models. The 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023), December 2023.

[Min+, EMNLP23] Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, Hannaneh Hajishirzi. FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation. The 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023), December 2023.

[Wang+, NeurIPS21] Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. Adversarial GLUE: A multi-task benchmark for robustness evaluation of language models. NeurIPS Datasets and Benchmarks 2021, December 2021.

[Nie+, ACL20] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding. The 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020), July 2020.

[Hendrycks+, ICLR21] Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, Jacob Steinhardt. Aligning AI With Shared Human Values. The International Conference on Learning Representations (ICLR 2021), May 2021.

[Jin+, NeurIPS22] Zhijing Jin, Sydney Levine, Fernando Gonzalez Adauto, Ojasv Kamal, Maarten Sap, Mrinmaya Sachan, Rada Mihalcea, Josh Tenenbaum, Bernhard Schölkopf. When to Make Exceptions: Exploring Language Models as Accounts of Human Moral Judgment. Advances in Neural Information Processing Systems 35 (NeurIPS 2022), November 2022.

[Dhamala+, FAccT21] Jwala Dhamala, Tony Sun, Varun Kumar, Satyapriya Krishna, Yada Pruk-sachatkun, Kai-Wei Chang, Rahul Gupta. BOLD: Dataset and Metrics for Measuring Biases in Open-Ended Language Generation. The 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT 2021), March 2021

[Rudinger+, NAACL18] Rachel Rudinger, Jason Naradowsky, Brian Leonard, Benjamin Van Durme. Gender Bias in Coreference Resolution. 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (NAACL 2018), June 2018.

[Zhao+, NAACL18] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, Kai-Wei Chang. Gen-

der Bias in Coreference Resolution: Evaluation and Debiasing Methods. 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (NAACL 2018), June 2018.

[Stanovsky+, ACL19] Gabriel Stanovsky, Noah A. Smith, Luke Zettlemoyer. Evaluating Gender Bias in Machine Translation. The 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019), July 2019.

[Nadeem+, ACL21] Moin Nadeem, Anna Bethke, Siva Reddy. StereoSet: Measuring stereotypical bias in pretrained language models. The 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), August 2021.

[Nangia+, ACL20] Nikita Nangia, Clara Vania, Rasika Bhalerao, Samuel R. Bowman. CrowS-Pairs: A Challenge Dataset for Measuring Social Biases in Masked Language Models. The 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020), November 2020.

[Smith+, EMNLP22] Eric Michael Smith, Melissa Hall, Melanie Kambadur, Eleonora Presani, Adina Williams. “I’m sorry to hear that”: Finding New Biases in Language Models with a Holistic Descriptor Dataset. The 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP 2022), December 2022.

[Li+, EMNLP20] Tao Li, Daniel Khashabi, Tushar Khot, Ashish Sabharwal, Vivek Srikumar. UN-QOVERing Stereotyping Biases via Underspecified Questions. Findings of the Association for Computational Linguistics: EMNLP 2020, November 2020.

[Parrish+, ACL22] Alicia Parrish, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Jana Thompson, Phu Mon Htut, Samuel Bowman. BBQ: A hand-built bias benchmark for question answering. Findings of the Association for Computational Linguistics: ACL 2022, May 2022.

[Zampieri+, NAACL19] Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, Ritesh Kumar. Predicting the Type and Target of Offensive Posts in Social Media. The 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (NAACL 2019), June 2019.

[Zong+, ACL21] Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, Preslav Nakov. SOLID: A Large-Scale Semi-Supervised Dataset for Offensive Language Identification. Findings of the Association for Computational Linguistics (ACL-IJCNLP 2021), August 2021.

[Gehman+, EMNLP20] Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, Noah A. Smith. RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models. Findings of the Association for Computational Linguistics (EMNLP 2020), November 2020.

[Shaikh+, ACL23] Omar Shaikh, Hongxin Zhang, William Held, Michael Bernstein, Diyi Yang. On Second Thought, Let’s Not Think Step by Step! Bias and Toxicity in Zero-Shot Reasoning. The

61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL 2023), July 2023.

[Wei+, arXiv23] Alexander Wei, Nika Haghtalab, Jacob Steinhardt. Jailbroken: How Does LLM Safety Training Fail? (v1). arXiv, <https://arxiv.org/abs/2307.02483>, July 2023.

[Deng+, arXiv23] Gelei Deng, Yi Liu, Yuekang Li, Kailong Wang, Ying Zhang, Zefeng Li, Haoyu Wang, Tianwei Zhang, Yang Liu. MasterKey: Automated Jailbreak Across Multiple Large Language Model Chatbots (v2). arXiv, <https://arxiv.org/abs/2307.08715>, October 2023.

[Bang, arXiv23] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, Pascale Fung: A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity (v4). arXiv, <https://arxiv.org/abs/2302.04023>, November 2023.

11. AI プロダクト品質保証コンソーシアムについて

名称： AI プロダクト品質保証コンソーシアム

英名： Consortium of Quality Assurance for Artificial-Intelligence-based products and services

略称： QA4AI コンソーシアム

URL： <http://www.qa4ai.jp/>

設立日： 2018 年 4 月 1 日

目的： AI 技術の活用・進化のさらなる促進と、AI プロダクトと社会との安心できる共生の実現

メンバ・団体（2024 年 1 月現在）：

青木 利晃（北陸先端科学技術大学院大学）

秋場 良太（有限責任あずさ監査法人）

池田 裕則（東芝インフラシステムズ株式会社）

石川 冬樹（国立情報学研究所）

伊藤 潤平（ウイングアーク 1st 株式会社）

伊藤 浩朗（日立オートモティブシステムズ株式会社）

猪又 憲治（三菱電機株式会社）

今井 健男（Idein 株式会社）

今谷 恵理（株式会社日立製作所）

上田 英介（FastLabel 株式会社）

宇治田 康浩（オムロン株式会社）

宇宿 哲平（有限責任あずさ監査法人）

梅津 良昭（株式会社リコー）*

遠藤 征樹（株式会社日立産業制御ソリューションズ）

大塚 祐次（株式会社日立産業制御ソリューションズ）

大西 秀一（株式会社ヴィッツ）

大野 敦寛（日立オートモティブシステムズ株式会社）

小川 秀人（株式会社日立製作所）*

荻野 恒太郎（楽天株式会社）*

長田 健一（日立オートモティブシステムズ株式会社）

小原 勇揮（ウイングアーク 1st 株式会社）

柏 良輔（横河電機株式会社）

岸 知二（早稲田大学）

鬼頭 正広（アイシン・ソフトウェア株式会社）

衣旗 宏和（三菱電機ソフトウエア株式会社）
窪田 邦夫（マレリ株式会社）
栗田 太郎（ソニー）
來間 啓伸
久連石 圭（株式会社東芝）
黒田 園子（パナソニック株式会社）
小宮山 英明（コニカミノルタ株式会社）
斎藤 辰彦（三菱電機株式会社）
榎原 彰*
佐藤 孝司（京都情報大学院大学）
島田 さつき（富士通クオリティラボ株式会社）
清水 真（アライズイノベーション株式会社）
柴山 吉報（阿部・井窪・片山法律事務所）
新原 敦介（株式会社日立製作所）
洲鎌 康（三菱電機株式会社）
鈴木 万治（DENSO International America Inc.）
須原 秀敏（株式会社ベリサーブ）
妹尾 義樹（国立産業技術総合研究所）
田口 研治（株式会社シーエーブイテクノロジーズ）
土屋 知典（富士通株式会社）
徳 隆宏（ダイキン工業株式会社）
徳本 晋（富士通株式会社）
中江 俊博（株式会社デンソー）
中川 純貴（株式会社日立製作所）
中澤 克仁（富士通株式会社）
中野 宏昭（富士フィルムビジネスイノベーション株式会社）
西 康晴 *
野原 優己（LINE Fukuoka 株式会社）
橋本 美樹（三菱電機株式会社）
濱田 晃一（株式会社ディー・エヌ・エー）
日置 智之（株式会社シナモン）
深井 寛修（株式会社明電舎）
藤井 岳（株式会社シナモン）
堀川 透陽（株式会社ベリサーブ）
誉田 直美（株式会社イデゾン）

増田 聰（東京都市大学）
町田 欣史（株式会社エヌ・ティ・ティ・データ）
松原 修（富士通株式会社）
松谷 峰生（NPO 法人 ASTER）
真鍋 誠一（株式会社センスタイムジャパン）
三浦 真樹（富士通株式会社）
三島 浩一（三菱電機株式会社）
光本 直樹（株式会社デンソー）
宮坂 隆之（株式会社本田技術研究所）
宮下 修治（三菱電機株式会社）
明神 智之（株式会社日立製作所）
三輪 祥太郎（三菱電機株式会社）
向山 輝（日本電気株式会社）
武藤 裕之（アイシン・ソフトウェア株式会社）
森川 聰久（株式会社ヴィッツ）
山下 直人（ウイングアーク 1st 株式会社）
山口 晋一（慶應義塾大学 SDM 研究所）
山元 浩平（株式会社コーピー）
吉岡 信和（早稲田大学）
鷺崎 弘宜（早稲田大学）
渡部 和也（アライズイノベーション株式会社）

国立研究開発法人 宇宙航空研究開発機構 研究開発部門 第三研究ユニット
特定非営利活動法人 ソフトウェアテスト技術振興協会
一般財団法人 日本科学技術連盟
(以上 五十音順)

『AI プロダクト品質保証ガイドライン』
AI プロダクト品質保証コンソーシアム (QA4AI コンソーシアム) 編
2024.01 版 2024 年 1 月 17 日公開
<http://www.qa4ai.jp/>

A. チェックリストの新旧対照表

表 A.1: Date Integrity の対照表

旧 (2019.05 版)	変更内容	新 (2020.08 版以降)	
量は充分か、	表現変更	(a.i)	想定する学習手法の適用前提や統計的観点から十分な量のデータがあるか.
意味のある量か、	表現変更	(a.ii)	想定する要求・適用環境において、希少な状況や分類クラスの偏りがある場合であっても、それらに対して十分な量のデータがあるか.
「かさ増し」しても大丈夫か	表現変更	(a.iii)	データ量が少ない場合、「かさ増し」(人工的なデータ生成など)で補完が可能か.
求める母集団のサンプルか、実際のデータを利用しているか、	表現変更	(b.i)	想定する要求・適用環境に意味の観点から対応した適切なデータとなっているか.
不必要的データが含まれていないか、含むべきでない母集団のデータと混ざっていないか	表現変更	(b.ii)	要求・適用環境の想定にそぐわないデータが入っていないか.
	新規	(b.iii)	人工的に作成・加工したデータについても、要求・適用環境を適切に表現しているといえるか.
コストは適正か	表現変更	(b.iv)	データの収集等の費用対効果の観点からも適切であるか.
データに関する要求事項を満たしているか、	表現変更	(c.i)	データに関するステークホルダーの要求事項を満たしているか
データに関する制約に反していないことを監視しているか	表現変更	(c.ii)	データが満たすべき不变条件や整合性条件、学習対象となる判断の公平性、個人情報の有無など、データに対する制約を満たしているか.
偏りやバイアス、汚染は無いか、自分たちが考えている「偏りを発生させるもの」だけでよいか	表現変更	(d.i)	潜在的なバイアスや汚染の可能性について、多様なステークホルダーや社会への影響の観点から検討し、データが適切であることを確認したか.

データは複雑すぎないか、	表現変更	(e.i)	学習させたい推論機能に対して、必要以上の情報量や傾向を含む複雑なデータとなっていないか。
単純すぎないか、必要な要素を適切に含んだサンプルか、ラベルは妥当か	表現変更	(e.ii)	データを単純化しすぎて、必要な情報が入っていないことはないか。
データ内の性質（多重共線性など）は適切に考慮されているか	表現変更	(f.i)	想定する学習手法の適用前提となるようなデータの性質（多重共線性など）は適切に考慮されているか
それぞれのデータは常識的な値か、	表現変更	(g.i)	データに含まれている値は、対象ドメインの知識などと照らし合わせて現実的に発生する妥当な値となっているか。
外れ値は本当に外れているデータか、欠損に意味はないか、外れ値や欠損値の扱いは適切か	表現変更	(g.ii)	外れ値と欠損値と判断した値は、真に現実的な値ではなく取り除くべきであることを確認したか。データを取り除くための前処理は適切であったか。
所有権や著作権・知的財産権、機密性、プライバシーは適切に考慮されているか	表現変更	(h.i)	データの利用が契約や第三者の知的財産権により制限されないか、データの利用に法令上、倫理上の問題はないか、プライバシー等への配慮が必要ないか。
学習用データと検証用データは独立しているか	-	(i.i)	学習用データと検証用データは独立しているか。
オンライン学習を行う場合、その影響を適切に考慮しているか	表現変更	(j.i)	インクリメンタルに追加や置き換え、削除されるデータについて、適切な運用機構・体制を設け、監視、制御や制限、検証を行っているか。
学習用プログラムやデータ生成プログラムの不具合によってデータの意味が毀損されないか	表現変更	(k.i)	データに対する前処理、作成・加工などの処理を行うアルゴリズムの特性や、そのライブラリやそれを呼び出すプログラムの不具合、誤った利用により、データの適切さが失われていないか。

表 A.2: Model Robustness の対照表

旧 (2019.05 版)	変更内容	新 (2020.08 版以降)	
正答率、適合率、再現率、F 値といった精度は妥当か	表現変更	(a.i)	正答率、適合率、再現率、F 値といった推論性能に関する評価指標の値は、要求に対して十分か。
汎化性能は確保されているか	-	(b.i)	汎化性能は確保されているか。
(AUROC といった) モデルのよさを表す指標は充分か	表現変更	(c.i)	(AUROC といった) 精度以外のモデルのよさを表す指標についても適切な指標を選定し充分に評価したか。
学習は適切に進行したか	-	(d.i)	学習は適切に進行したか。
局所最適に陥っていないいか	表現変更	(d.ii)	学習結果が局所最適に陥っていないか。
適切なアルゴリズムやハイパーパラメータかどうかの検討は行ったか、	-	(e.i)	適切なアルゴリズムやハイパーパラメータかどうかの検討は行ったか。
十分に交差検証などを行ったか	-	(f.i)	十分に交差検証などを行ったか。
ノイズに対して頑健か	-	(g.i)	ノイズに対して頑健か。
数理的多様性、意味的多様性、社会的文化的多様性などを考慮し、十分に多様なデータで検証を行ったか	-	(h.i)	数理的多様性、意味的多様性、社会的文化的多様性などを考慮し、十分に多様なデータで検証を行ったか。
デグレードは許容可能な範囲か、デグレードの影響範囲を把握できているか、学習は再現可能か、学習時のふるまいと提供時のふるまいに齟齬はないか	表現変更	(i.i)	モデルを更新する場合、以前の振る舞いとの変化について把握しているか、それが許容可能であることを確認しているか。

	新規	(i.ii)	特に自動でのモデル更新・配備を行う場合、自動化された検査内容は十分であるか.
モデルが陳腐化していないか、実データに対する予測品質が劣化していないかないか	表現変更	(j.i)	運用時における傾向の変化により、モデルの性能、妥当性、有用性が低下する可能性を検討し、それに対するモデルの頑健性確保、運用における監視などの対策をとっているか.
目標指標の計測が難しい場合、計測できるメトリクスとの関連は妥当か	System Quality へ		
	新規	(k.i)	学習アルゴリズムの特性や、そのライブラリやそれを呼び出すプログラムの不具合や誤った利用により、不適切なモデルとなっていないか.

表 A.3: System Quality の対照表

旧(2019.05 版)	変更内容	新(2020.08 版以降)	
価値は適切に提供されているか、	表現変更	(a.i)	システム全体により価値は適切に提供されているか、提供価値を計測できているか.
目標指標の計測が難しい場合、計測できるメトリクスとの関連は妥当か	Model Robustness より、表現変更	(a.ii)	価値の計測が難しい場合、計測できる代替メトリクスとの関連は妥当か.
性能などシステム全体のふるまいが劣化していないかないか	表現変更	(b.i)	AI の導入や変更がシステム全体のふるまいや性能などの品質に悪影響を与えていないか.
システムを全体として、および意味のある単位で評価を行ったか	表現変更	(c.i)	システムを全体として、および意味のあるサブシステム単位で評価を行ったか.

発生しうる品質事故の致命度は許容できる程度に低く抑えられているか	-	(d.i)	発生しうる品質事故の致命度は、許容できる程度に低く抑えられているか。
品質事故を引き起こしうる事象の発生頻度は低いと見積もることができるか、	-	(d.ii)	品質事故を引き起こしうる事象の発生頻度は低いと見積もることができるか。
事象の発生頻度や事象の網羅性、環境統制性の検討は充分か	表現変更	(d.iii)	事象の発生頻度、事象の網羅性、事象に影響を与える環境の制御可能性に関する検討は十分か。
システムの事故到達度	表現変更	(e.i)	システムの事故到達度は十分に抑制しているか。
・安全機能・耐攻撃性は充分か	表現変更	(e.ii)	十分な安全機能や耐攻撃性を提供しているか。
AI の寄与度を抑えられているか、	表現変更	(f.i)	システムに対する AI の寄与度を抑えられているか。
システムが依存する他の（AI の、もしくは非 AI の）システムの変更是迅速かつ適切に反映できるか、	-	(f.ii)	システムが依存する他の（AI の、もしくは非 AI の）システムの変更の影響は、迅速かつ適切に反映できるか。
不具合の影響を充分低く抑えられるか	表現変更	(f.iii)	AI の不具合の影響を十分に低く抑えられるか。
保証性、説明可能性、納得性は充分か	表現変更	(g.i)	ステークホルダーに対する保証性、説明可能性、納得性は十分か。
	新規	(h.i)	AI プロダクトが第三者の知的財産権を侵害しないか
	新規	(i.i)	継続的な運用に伴うシステムの性能などの品質が低下する可能性を検討したか。
	新規	(i.ii)	運用中のシステム品質低下を検知する仕組みを検討したか。

表 A.4: Process Agility の対照表

旧 (2019.05 版)	変更内容	新 (2020.08 版以降)	
データ収集の速度とスケーラビリティは充分か	表現変更	(a.i)	データ収集の速度とスケーラビリティは十分か。
充分短い反復単位で反復型開発は行っているか、	表現変更	(b.i)	十分に短い反復単位で反復型開発を行っているか。
モデル・システムの品質向上の周期は充分短いか、	表現変更	(b.ii)	モデル・システムの品質向上の周期は十分に短いか。
運用状況の継続的なフィードバックは頻繁か	表現変更	(b.iii)	運用状況の継続的なフィードバックは頻繁に行っているか。
	新規	(c.i)	問題が発生した時にその原因を解析するために、問題発生時の状況を記録し取得できる仕組みを備えているか。
	新規	(c.ii)	取得した問題発生時の状況をもとに事象を再現できるか。
リリースロールバックは簡便で迅速に行えるか	-	(d.i)	リリースロールバックは簡便で迅速に行えるか。
新しい特徴量を迅速に追加したりモデルを迅速に改善したりできるなど、よりよくなっていく見込みはあるか	-	(e.i)	新しい特徴量を迅速に追加したりモデルを迅速に改善したりできるなど、よりよくなっていく見込みはあるか。
段階的リリースやカナリアリリースの度合は適切か、	-	(f.i)	段階的リリースやカナリアリリースの度合は適切か。
リリース直前にシステム全体やモデルの評価を行っているか	-	(f.ii)	リリース直前にシステム全体やモデルの評価を行っているか。

開発・探索・検証・リリースなどの自動化は充分か	表現変更	(g.i)	開発・探索・検証・リリースなどの自動化は十分か。
データ、モデル、環境、コード、出力などの構成管理が適切に行われているか	-	(h.i)	データ、モデル、環境、コード、出力などの構成管理が適切に行われているか。
開発者やチームは技術的に充分納得し共感しているか	表現変更	(i.i)	開発者やチームは技術的に十分に納得し共感しているか。
開発チームは適切な能力を持った人財を備えているか	表現変更	(i.ii)	開発チームは適切な能力を持った人財を備えているか。
経験を技術に反映させられているか	-	(j.i)	経験を技術に反映させられているか。
開発チーム外のステークホルダーは充分納得しているか	表現変更	(k.i)	開発チーム外のステークホルダーは充分納得しているか。

表 A.5: Customer Expectation の対照表

旧 (2019.05 版)	変更内容	新 (2020.08 版以降)	
顧客の期待は高いか	-	(a.i)	顧客の期待は高いか。
顧客は確率的動作という考え方を受容していないか、	-	(b.i)	顧客は確率的動作という考え方を受容していないか。
リスク・副作用を理解していなかったり受容したりしていないか	表現変更	(b.ii)	リスク・副作用を理解していないか、もしくは安易に受容して必要な対策を怠っていないか。
継続的実運用にどのくらい近いか	-	(c.i)	継続的実運用にどのくらい近いか。
データの量や質に対する認識は甘いか	-	(b.iii)	データの量や質に対する認識は甘いか。

狙っているのが「人間並み」か	-	(a.ii)	狙っているのが「人間並み」か.
法規制、知的財産権、プライバシー、コンプライアンス、社会的受容が必要か	表現変更	(d.i)	AI プロダクトの利用に法令上、倫理上の問題はあるか、第三者のプライバシー等への配慮が必要か、AI プロダクトの利用が社会的に否定的か.
”合理的” 説明を求める傾向や、“外挿” や “予測” をしたがる傾向、”原因” や “責任（者）” を求めたがる傾向はあるか	-	(b.iv)	“合理的” 説明を求める傾向や、“外挿” や “予測” をしたがる傾向、”原因” や “責任（者）” を求めたがる傾向はあるか.
納得感を共感する風土や雰囲気、仕事の進め方は少ないか、	-	(e.i)	納得感を共感する風土や雰囲気、仕事の進め方は少ないか.
顧客担当者・チームで意思決定できる権限や範囲は少ない・狭いか	-	(e.ii)	顧客担当者・チームで意思決定できる権限や範囲は少ない・狭いか.