

対話型生成 AI を活用した業務システム向けテストデータ生成の実 践的検証

Practical Validation of Test Data Generation for Enterprise Systems by Conversational AI

研究者：高橋 昭光（アクセンチュア株式会社）

主査：石川 冬樹（国立情報学研究所）

副主査：徳本 晋（富士通株式会社）

アドバイザー：栗田 太郎（フリー株式会社）

研究概要

業務システム開発におけるテストデータ生成は、複雑な参照整合性の維持や分布制御に多大な工数を要し、また本番データの利用には法的制約がある。本研究では、対話型生成 AI を活用したテストデータ自動生成手法を確立した。架空のペット保険システム（8 テーブル約 140 万件）を題材に、第 1 段階：構造整合性、第 2 段階：分布制御、第 3 段階：自然性向上という段階的生成戦略と、事前割当方式による高精度分布制御を提案した。評価の結果、参照整合・一意制約違反率 0%、ドメインルール違反率 0%、最大分布誤差 0.59%（目標 5%）を達成し、複雑な要件を両立できることを実証した。さらに、インプット資料の標準構成 9 種類と段階的な対話手順を定義し、非決定的な LLM を用いた手法の再現性を担保する基盤を確立した。

Abstract

Test data generation for business systems faces challenges in effort, quality, and legal constraints. We developed an automated test data generation method using Large Language Models (LLMs). Our approach employs a three-phase strategy (structural integrity → distribution control → naturalness enhancement) and a pre-allocation method that achieves high-precision distribution control. Evaluation using a virtual pet insurance system with eight tables across 1.4 million records demonstrated 0% referential integrity and uniqueness constraint violations, 0% domain rule violations, and maximum distribution error of 0.59% (target: 5%). We established a standardized input structure with nine document types and a systematic dialogue procedure to ensure reproducibility with non-deterministic LLMs. These results demonstrate that LLM-based test data generation can simultaneously satisfy complex relationships, business logic, and statistical distributions.

1. はじめに

1.1 研究の背景と目的

現代の業務システムは多様化・大規模化が進んでおり、開発や保守の現場では、テスト工程において膨大な量のテストデータを迅速かつ整合性を持って準備することが強く求められている。従来のテストデータ生成は、開発者やテスト担当者が手作業でデータを作成するか、本番データをマスキングして流用する方法が主流であった。しかし、システム全

体の複雑なリレーションやドメインルールを反映したデータを人手で用意するのは困難であり、特に多くのテーブルで相互に外部キー制約を持つシステムでは、整合性確保に多大な工数を要する。また、個人情報保護法や GDPR 等の規制強化により、本番データの取り扱いには法的・運用上の制約が大きい。

本研究で提案する手法は、主に結合テスト以降のテストデータ準備フェーズを想定しており、テーブル設計と業務ロジックが確定した段階で適用する。単体テスト段階での小規模データ生成にも適用できるが、テーブル設計が安定していることが前提となる。利用者は QA エンジニアを想定し、生成したテストデータは開発者に CSV 形式で配布する。本手法における反復プロセス (5.1 節参照) は、初回のテストデータ生成時に LLM との対話を通じて段階的にスキプトの品質を向上させる過程を指す。一度要件を満たすスキプトが完成すれば、同一要件でのデータ再生成は容易に実行できる。

こうした課題を背景に、本研究では対話型生成 AI、特に大規模言語モデル (Large Language Model: LLM) を活用し、システム全体に整合性の取れた、かつ本番データに近い分布や属性を備えたテストデータ自動生成手法を確立することを目的とする。具体的には、実践的な検証を通じて、その有効性や現場適用上の課題、今後の展望を明らかにする。

1.2 研究の概要と貢献

本研究では、架空のペット保険システム (8 テーブル, 約 140 万件) を題材とし、LLM との対話により Python スクリプトを生成してテストデータを自動生成・評価した。

本研究の貢献は次の 3 点である。第一に、要件を段階的に分離する生成戦略により、複雑な参照整合性、ドメインルール、統計的分布を同時に満たすデータ生成を実現した。第二に、目標値から逆算して割り当てる分布制御方式により、高精度な統計的分布の再現を達成した。第三に、標準化されたインプット資料と対話手順により、非決定的な LLM を用いた手法の再現性を担保する基盤を確立した。これらにより、LLM を単なるコード生成ツールではなく、技術的課題の共創的パートナーとして活用する新しいアプローチを提示した。

1.3 対象システムの概要

本研究で題材とする架空のペット保険システムは、申込者、ペット、申込、申込状態履歴、申込支払、支払履歴、証券、申込関連書類の 8 テーブルで構成される。主キーは業務的意味を持たないシーケンシャル ID (例: APL-0001 形式) で設計され、外部キー制約により参照整合性が定義されている。申込者からペットおよび申込への 1 対多の関係、申込から申込状態履歴および申込支払への親子関係により、複雑な参照構造を形成している。業務プロセスは次の 3 ステップからなる。(1) 申込者がペット情報を登録し保険を申込、(2) 審査・承認を経て保険証券を発行、(3) 保険料の定期支払いである。

1.4 用語の定義

本研究で使用する主要な用語を以下に定義する。

- ・ テストデータ生成: スキーマ適合性, 参照整合性 (PK (Primary Key) 一意性・FK (Foreign Key) 整合), ドメインルール適合性, 統計的妥当性の 4 要件を満たすデータの自動生成。
- ・ PK 一意性: 主キーに重複値が存在しないこと。
- ・ FK 整合: 外部キー値が参照先テーブルの主キーに必ず存在すること。
- ・ ドメインルール適合: 業務上の制約条件 (状態遷移, 金額計算等) への準拠。
- ・ 分布の再現: 実運用データの統計的特性を指定誤差範囲内で再現すること。

2. 関連研究と本研究の位置づけ

2.1 従来のテストデータ生成手法

テストデータ生成の従来手法は、実務経験に基づき、(1)手作業によるデータ作成、(2)本番データのマスキング、(3)テンプレートベースの自動生成、の3つに分類できる。

手作業方式は個別の業務要件への柔軟な対応が可能だが、複雑なシステムでは整合性確保に多大な工数を要し、作業者のスキルに品質が依存する。本番データマスキング方式は現実的だが、個人情報保護法・GDPR等により法的制約が厳しい。テンプレートベース自動生成方式は、事前に定義された固定ルールやSQLスクリプト、データ生成ツール(DBUnit, DataFactory等)により効率的にデータを生成できるが、複雑なドメインルールや精密な分布制御への対応が困難である。

2.2 LLM 活用における技術的課題

上記の従来手法の限界を克服する手段として、近年 LLM の活用が注目されている。LLM は自然言語理解とコード生成能力により柔軟なデータ生成が期待されるが、業務システムへの適用において、本研究では以下の課題に着目した。

第一に、複数テーブル間の参照整合性を確実に維持する方法である。LLM は個別テーブルのデータ生成は可能だが、複雑な外部キー制約を持つシステム全体で整合性を保証する体系的な手法は明確ではない。

第二に、複雑なドメインルールと統計的分布を同時に満たす戦略である。業務ロジックに準拠しつつ、実運用を想定した統計的分布を再現する体系的な生成戦略が必要である。

第三に、非決定的な LLM で再現可能なプロセスを構築する方法である。対話型 AI の柔軟性を活かしつつ、現場で再現可能な標準プロセスとして確立する手法が求められる。

これらの課題は、実務上の観点（参照整合性維持の高コスト、業務ロジックと統計的分布の両立困難）と LLM 活用における技術的制約（生成結果の非決定性）の両面から設定した。本研究は上記3つの課題に対し、段階的生成戦略、事前割当方式、標準化されたインプット資料構成により解決を図る。

3. 研究のアプローチと評価基準

3.1 本研究のアプローチ

本研究では、2.2 節で提示した3つの課題に対応する形で解決アプローチを設定した。複数テーブル間の参照整合性維持には、ID 体系の統一管理と生成順序の制御により対処する。複雑なドメインルールと統計的分布の両立には、段階的生成戦略と事前割当方式により対処する。非決定的な LLM での再現可能なプロセス構築には、標準化されたインプット資料と段階的プロセスにより対処する。これらのアプローチにより、LLM 活用の技術的有効性と実用的再現性を両立させる。

3.2 評価基準

本研究の成果は以下の基準で評価する。

表 1: 評価基準と測定方法

#	評価対象	評価項目	説明
1	フォーマット・スキーマ適合	テーブル定義準拠率	データ型、桁数、NOT NULL に準拠しているか
2		余分/欠落列率	余分な列/欠落した列がないか
3	コード・マスタ適合・データ品質	マスタコード適合率	マスタやコード定義に準拠しているか
4		不正値率	コンテキストに定義した制約に準拠しているか
5	参照整合・一意制約	外部キー整合率	ER 図のリレーションとおりに生成できているか
6		一意制約違反率	一意制約違反がなく生成できているか
7	データボリューム・	ボリューム達成率	生成されたデータ数が想定とおるか
8	期間・分布の達成	期間内率	指定した期間とおりに生成できているか

#	評価対象	評価項目	説明
9		分布誤差	設定した分布とおりに生成できているか
10	ドメインルール適合	ルール適合	設定したドメインルールに適合できているか

これらの評価基準は、テストデータの役割から導出した。データベース層での要件として、スキーマ適合性（項目 1-2）、コード・マスタ適合性（項目 3-4）、参照整合性（項目 5-6）を設定した。アプリケーション層での要件として、ドメインルール適合（項目 10）を設定した。統計的模擬の要件として、分布制御（項目 7-9）を設定した。

4. 提案手法：対話型生成 AI を活用したテストデータ自動生成

4.1 生成プロセスの概要

本手法は、プロンプトエンジニアリングによる対話的スクリプト生成を採用しており、AI エージェントや専用フレームワークは使用しない。これらを使用せずチャット型の LLM インタフェースのみで実行可能とすることで、現場への導入障壁を低減し、異なる担当者でも同等の結果を得やすくするためである。本手法では事前準備として、テーブル定義、ER 図、ドメインルール等を記述した Markdown ファイルおよび yaml ファイル群（詳細は 4.3 節参照）を整備する。これらをインプット資料として LLM に提示することで、担当者が異なっても同じ前提知識から対話を開始できる状態を整える。その上で、段階的生成戦略（第 1 段階：構造整合性、第 2 段階：分布制御、第 3 段階：自然性向上）に従い、各段階でスクリプト生成と自動評価を繰り返す。自動評価により導出された評価結果を人間が解釈し、段階目標の達成状況を判断した上で LLM に改善を指示することで、段階的に要件を満たすスクリプトに改善していく。

4.2 生成方式の選択モデル

LLM を用いたテストデータ生成には、(A) LLM が直接データを生成する方式、(B) LLM にスクリプトを作成させデータを生成する方式、の 2 つのアプローチが存在する。[1]本研究ではこれらの選択基準を体系化し、表 2 に示す選択マトリクスを作成した。

表 2：生成方式選択マトリクス

#	データ件数	整合性複雑度	再現性要求	推奨方式	理由
1	小(~1 万)	低(単一)	低(再生成不要)	(A)	迅速・低コスト
2	中(1 万~500 万)	低(単一)	低(再生成不要)	(A)/(B)	(A)を使う場合はモデル非決定性を緩和する工夫が必要
3	小(~1 万)	中(2~10)	低(再生成不要)	(A)/(B)	
4	小(~1 万)	中(2~10)	中(再生成要)	(B)	関係整合と分布制御を保った生成にはパラメトリックな生成と検証ロジックが必要
5	中(1 万~500 万)	中(2~10)	中(再生成要)	(B)	
6	大(500 万~)	高(複数ドメイン)	高(完全再現)	(B)	高度な制約と決定的再現

本研究では、単一テーブルあたり最大 50 万件規模（総計約 140 万件）、整合性複雑度 8 テーブルで、再現性要求は必ずしも高くないため、当初は(A)直接生成を想定していたが、LLM (GPT-5) との対話の中で自発的にスクリプト生成を提案されたため(B)スクリプト生成方式に切り替えた。

4.3 インプット資料の標準構成

本研究を通じて、LLM による効果的なテストデータ生成に必要なインプット資料の標準構成を確立した。

表 3：インプット資料の標準構成と役割

#	インプット資料	形式	役割	必要度
1	テーブル定義	Markdown	スキーマ定義、PK だけでなく FK も指定	必須

#	インプット資料	形式	役割	必要度
2	ER 図	PlantUML	リレーションやカーディナリティを定義	必須
3	コード定義	Markdown	コードと値の組み合わせ一覧	必須
4	マスタデータ	Markdown	参照用マスタ	必須
5	ID 定義	yaml	ID 形式の統一管理	推奨
6	テストデータ指示	Markdown	生成要件, ドメインルール	必須
7	データチェックリスト	Markdown	評価項目	必須
8	分布情報	Markdown	分布目標値, 保守性向上のため外部化	推奨
9	外部定義リスト	yaml	ペット名や住所等の項目のリスト	任意

これらの 9 種類は, 初期構成 (項目 1-4, 6-7) と段階的追加 (項目 5, 8, 9) の組み合わせで確立された. 必要度は, 各項目の欠如が生成結果に与える影響度に基づき分類した.

4.4 段階的生成戦略

複雑な要件を一度に満たすのではなく, 3 段階で段階的に品質を向上させる戦略を確立した.

第 1 段階 (構造整合性):

LLM が生成するスクリプトは単体では各テーブルのデータを生成できるが, 複数テーブル間の生成において生成順序の不適切さによる FK 違反や, ID 形式の不統一による PK 重複が発生しやすい. そこで, ID 定義を一元管理する yaml ファイルを用意して ID 形式を統一し, 生成順序を親テーブルから子テーブルへの順に制御することで, 参照整合性を保証する. 本研究ではこれらの手法を採用し, PK 重複率 0%, FK 違反率 0%, データ型違反率 0%を達成した.

第 2 段階 (分布制御):

ランダム生成後に分類する従来方式は確率的変動により安定した分布精度を達成できなかった. そこで, 事前割当方式 (4.5 節で詳述) により, 年齢帯・地域等の目標分布を高精度に達成する. 従来方式の問題を解決し, 分布誤差 5%以内を実現した.

第 3 段階 (自然性):

LLM に生成方法を具体的に指示しない場合, 氏名に連番が振られるなど不自然なデータが生成されやすい. そこで, Faker (氏名・住所・電話番号等のリアルなダミーデータを生成する Python ライブラリ) を用いた日本語名生成, ペット名や住所の外部定義リストからのランダム選択により自然なデータを生成した. [2] レビューアによる主観評価で違和感の有無を検証し, 定性的な品質を確認した.

この 3 段階の分割は, LLM との対話を通じて段階的に形成された. 全要件を同時に指示すると参照整合性違反と分布誤差が同時発生し優先順位が判断できなかったが, 対話を通じて構造整合性から分布制御, 自然性へと続く品質要件の依存関係が明らかになり, この階層構造に従って 3 段階戦略として体系化した.

4.5 事前割当方式による分布制御

第 2 段階では年齢帯や地域などのカテゴリ分布を目標値に合わせる必要がある. 当初採用した属性値をランダム生成後に分類する方式では, 少量データでの試行で最大誤差が 5%を超える場合があり, 目標の 5%以内を安定して達成できなかった. 原因として, 確率的な変動による累積誤差と複数属性の多重依存が考えられ, 高精度制御には限界があった.

この問題を解決するため, LLM との対話を通じて生成順序を逆転させる事前割当方式が提案され確立された. 本方式は, 目標比率から各カテゴリ (年齢帯, 地域等) の生成件数を事前確定し, 各レコードにカテゴリを割り当てたリストをシャッフル後, カテゴリから具体的な年齢や都道府県等の属性値を逆算する. この方式を採用した結果, 分布誤差は目標 5%以内を安定して達成し (本評価では最大 0.59%), シード固定により完全に再現可能と

なった。本方式は年齢帯以外の離散分布全般に適用可能である。

4.6 再現性の担保

LLM を活用した手法では非決定性が課題となるため、現場での実用性を確保する目的で、対話による生成プロセスの再現性を次の手順で担保した。まず、標準化されたインプット資料（テーブル定義、ER 図、ドメインルール等 9 種類）を LLM に提示し、異なる担当者でも同じ前提知識から対話を開始できる状態を整える。次に、段階的生成戦略に従い、当該段階の目標のみに集中して LLM と対話を進める。自動評価スクリプトの出力を LLM にフィードバックして改善を繰り返し、段階目標達成後に以降の段階に進む。この手順により、段階を限定することで LLM の応答を予測可能な範囲に制御し、異なる担当者でも同等の品質を達成できる枠組みを実現した。

5. 評価と検証

5.1 検証環境と手順

検証環境：

OS - Windows 11, Python - 3.13, 主要ライブラリ - Faker, pandas, PyYAML, 開発環境 - Visual Studio Code with GitHub Copilot (GPT-5)

検証手順：

段階的生成戦略（4.4 節）に従い、各段階では少量データ（申込者 100 件程度）で評価スクリプトを実行し、LLM にフィードバックしてスクリプトを改善した。各段階の完了は表 5 で確認し、全段階完了後に目標件数（申込者 10,000 件, 申込 100,000 件）でデータ生成を実行した。

5.2 評価結果

本研究で生成したデータは、単一テーブルあたり最大 50 万件規模の大規模データに及んだ。なお、自然性（第 3 段階）についてはレビューアによる主観評価（定性所見）として確認したため、表 5 の定量評価項目には含めない。定性所見として、氏名に連番が付与される、記号が混入する、小書き文字（やゅょ等）から始まる等、当該項目として通常ありえない値を指摘した。

表 4：生成テストデータ件数

#	テーブル名	物理名	項目数	親テーブル	件数
1	申込者	t_applicant	16	-	10,000
2	ペット	t_pet	14	申込者	18,600
3	申込	t_application	15	申込者	100,000
4	申込状態履歴	t_application_status_history	9	申込	449,415
5	申込支払	t_application_payment	16	申込	100,000
6	支払履歴	t_payment_history	12	申込支払	508,859
7	証券	t_policy	10	申込	98,000
8	申込関連書類	t_document	11	申込	147,339

表 5：評価結果

#	評価対象	評価項目	評価内容	目標値	実測値	達成
1	フォーマット・スキーマ適合	テーブル定義準拠率	型, 長さ, NOT NULL 準拠	100%	100%	✓
2		スキーマ適合率	余分列, 欠落列無し	100%	100%	✓
3	コード・マスタ適合	マスタコード適合率	マスタ, コード準拠	100%	100%	✓
4	データ品質	不正値率	負数, ID 形式準拠	100%	100%	✓

#	評価対象	評価項目	評価内容	目標値	実測値	達成
5	参照整合・一意制約	外部キー整合率	FK 準拠率	100%	100%	✓
6		一意制約違反率	PK 準拠率	100%	100%	✓
7	データボリューム・期間・分布の達成	ボリューム達成率	目標件数達成	100%	100%	✓
8		期間内率	指定期間遵守, ピーク月	100%	100%	✓
9		分布誤差	年齢帯分布誤差(最大)	Gap:<5%	0.26%	✓
10			地域分布誤差(最大)	Gap:<5%	0.59%	✓
11			犬猫比率誤差	Gap:<5%	0.01%	✓
12			支払頻度分布誤差(最大)	Gap:<5%	0.35%	✓
13			申込状態分布誤差(最大)	Gap:<5%	0.5%	✓
14			ペット数 P 分布 (P25-P99)	全部一致	全部一致	✓
15			氏名重複率	Gap:<5%	25.5%	[注]
16	ドメインルール適合	ルール適合	状態遷移準拠率	100%	100%	✓
17		金額整合準拠率	100%	100%	✓	

注: 氏名重複率 25.5%は Faker ライブラリの制約による構造的制約 (Faker の日本語氏名生成機能は内部辞書として姓 50 パターン×名 51 パターン≒2,500 通りのみを保持しているため, 申込者 10,000 件を生成すると数学的に重複が発生する). 重複を許容できない場合はカスタム辞書の準備が必要.

スキーマ適合・参照整合・コード適合・データ品質・ドメインルールの全項目で目標を達成(違反率 0%). 分布誤差はすべて目標 5%以内を達成し, 特に事前割当方式により最大 0.59% (目標 5%) の高精度を実現した.

5.3 考察

本研究では, 3.1 節で設定した 3 つのアプローチについて評価を行い, すべて目標を達成した. 以下, 各アプローチの技術的意義について考察する.

第一に, 複数テーブル間の参照整合性維持について, ID 定義の一元管理と生成順序制御により, LLM の非決定性の下で 8 テーブル約 140 万件の規模で違反率 0%を達成した. 本手法は LLM の柔軟性を活かしつつ構造整合性を保証する新しい枠組みである.

第二に, 複雑なドメインルールと分布制御の両立について, 段階的生成戦略により要件を分離し, 事前割当方式により高精度分布制御を実現した. 事前割当方式は生成順序を逆転させることで確率的変動を排除し, LLM との対話を通じて創出された解決策である.

第三に, 再現可能なプロセス構築について, 標準化されたインプット資料と段階的対話手順により, LLM の非決定性と柔軟性を両立させつつ再現性を確保した. 段階目標の明示と評価フィードバックにより, 異なる担当者でも同等の品質を達成できる枠組みを確立した.

これら 3 つのアプローチを実現する過程で, LLM 活用の有用性として, LLM 自身が問題の複雑度を認識し, 適切な実装方式 (事前割当方式や ID 一元管理など) を提案できる点が明らかになった. また, 本手法は従来手法と比べて参照整合性の確認・修正に要する工数を大幅に削減できることも確認された. 手作業方式では同規模データの整合性確保に多大な工数を要し, 本番データマスキング方式は法的制約とセキュリティリスクがある. 本手法の総工数は数人日程度であり, 生成スクリプトの再利用により 2 回目以降の更新工数をさらに削減できる.

一方で, 生成結果の品質は, 事前の設計やインプット資料の粒度・正確性に大きく依存する. 十分な資料整備や設計が不十分な場合, 期待する品質を安定して得ることは難しい.

さらに, Faker ライブラリの日本語名生成など外部ライブラリに起因する制約も回避が困難であり, 例えば氏名の重複を許容できない場合はカスタム辞書の準備が必要となる.

6. 結論

6.1 達成された成果

本手法の意義は、単なる作業効率化に留まらず、ソフトウェア開発における人間と LLM の協働の新しいモデルを示したことにある。従来のコード生成ツールが人間の指示を忠実に実装する「実行者」であったのに対し、本研究では LLM が評価結果から問題を認識し解決策を自発的に提案する「共創的パートナー」として機能することを実証した。

また、LLM の非決定性という本質的課題に対し、標準化されたインプット資料と段階的プロセスにより再現性を担保する枠組みを確立したことは、LLM 活用の実用化における重要な方法論的貢献である。非決定的な技術を現場で再現可能なプロセスとして定着させる本アプローチは、生成 AI 技術の産業応用において広く適用可能な知見を提供する。

さらに、本手法により、テストデータという「見えにくいが不可欠な成果物」の品質と効率が飛躍的に向上することで、業務システム開発全体の生産性向上に寄与する可能性がある。特に、法規制によりテストデータ準備に苦慮してきた金融・医療・公共分野において、本手法は現実的な代替手段を提供するものである。

6.2 本研究の限界と今後の課題

本研究は 2.2 節で提示した 3 つの技術的課題（参照整合性維持、ドメインルールと分布の両立、再現可能なプロセス構築）に対し、段階的生成戦略、事前割当方式、標準化されたプロセスにより解決を図り、5.2 節で示した通り目標を達成した。一方で、以下の限界が存在する。

第一に、題材が限定されている。ペット保険という特定ドメインでの検証に留まり、他業種・他システムへの一般化可能性は未検証である。第二に、規模に制約がある。最大 50 万件規模での検証であり、数百万件を超える大規模システムへの適用可能性は確認できていない。第三に、LLM バージョンへの依存がある。GPT-5 での検証結果であり、他の LLM や将来のバージョンでの再現性は保証されない。第四に、外部ライブラリに制約がある。Faker の氏名生成における重複等、外部ライブラリの制約は回避困難である。第五に、適用フェーズに制約がある。本手法は、データモデルとドメインルールが確定している状況において有効性が実証されているが、新規開発初期のようにデータモデル自体が流動的な段階では、頻繁なスキーマ変更により生成スクリプトの改修コストが増大するため適用は推奨されない。

これらの限界に対し、今後の研究方向として次の 5 点を提案する。(1)金融・公共等の異なるドメインでの検証により手法の一般化可能性を評価する。(2)分散処理やストリーミング生成により数百万件規模への拡張性を検証する。(3)複数 LLM (Claude, Gemini 等) での比較検証により、LLM 非依存な標準プロセスの確立を目指す。(4)カスタム辞書の自動構築手法の開発により、外部ライブラリの制約を緩和する。(5)スキーマ変更時のスクリプト修正の効率化手法の確立により、データモデル流動期での適用コスト削減を図る。これらの取り組みにより、本手法の実用性と適用範囲の拡大を図る。本研究で確立した手法と知見が、業務システム開発の現場におけるテストデータ生成の効率化と品質向上に寄与することを期待する。

7. 参考文献

- [1] 鷲崎弘宜 (編集), 生成 AI によるソフトウェア開発 —設計からテスト, マネジメントまでをすべて変革する LLM 活用の実践体系, オーム社, 2025/11
- [2] Benoit Baudry, Khashayar Etemadi, Sen Fang, Yogya Gamage, Yi Liu, Yuxin Liu, "Generative AI to Generate Test Data Generators," IEEE Software, Volume 41, Issue 6, pp. 44-51, Nov.-Dec. 2024, doi: 10.1109/MS.2024.3449408