データ構造化手法とユニケージ開発手法の比較

田渕 智也
USP 研究所
t-tabuchi@usp-lab.com

當仲 寛哲
USP 研究所
tounaka@usp-lab.com

田中 湧也
USP 研究所
y-tanaka@usp-lab.com

要旨

事業の変化や企業間の競争が激しい時代の事業を支える業務システムにおいて，後戻りを行わない開発プロセスと RDB 等の構造化データを用いた従来型の開発手法では開発途中での手戻りが行えず， 迅速な変化への対応が困難である．本論文では，短い時間で解決策を検証し， 事業の変化にも柔軟な対応力を持ち， 技術アーキテクチャが一体である「ユニケージ開発手法」を提案する．

1. 背景

市場の変動が激しくなる今の時代に企業の事業を支える情報システムは変化に迅速かつ安価に追従することが求められている．後戻りをしない段階的な開発プロセスと構造化されたデータを用いる「データ構造化手法」では， 今日まで決められた手順やデータの仕様に従い複数人でシステムを開発できていたが，迅速な変化に対応できることが試されている現状である．[1, 6]

これに呼応する形で，現場では安価で迅速に対応できる開発手法が求められている．例としては，アジャイル開発手法が応用されているが， 現状は実際のシステム開発にどのように落とし込むかが課題となっている．[2] その要因の一つとして， 開発中の変更や完成までの過程が予測できないことが挙げられ， 変更が起きたときにいかに早く検証できるか， 試行錯誤のサイクルを短縮できる最適な技術アーキテクチャが確立されていないことも大きな要因である．[3] アジャイル型開発は既に体系化が行われ，ウォーターフォール型開発の対概念としては広く認知されているが [2]，それに適した技術アーキテクチャを明確にしていない．

そこで， 開発手法と技術アーキテクチャが一体となったユニケージ開発手法（技術アーキテクチャとアジャイル，リーン・TPS 開発手法の規則の融合）を提案する．

この手法では，システム開発工程に変更がある場合に迅速に解決策を検証する技術アーキテクチャ[4]を持つため，これまで課題となっている変化に迅速かつ安価に対応し，企業の情報システムの開発をすることが可能となっている．

本論文では，データ構造化手法とユニケージ開発手法の2つの手法の比較を行い，ユニケージ開発手法が変化に迅速に対応でき，試行錯誤の妨げとなる「分業」を克服した柔軟な手法であることを説明する．まず，2 章ではユニケージ開発手法の代表的な特徴についてデータ構造化手法と比較を行いながら説明する．3 章では開発・設計の各工程における特徴を比較する．4 章ではシステム開発を行うためのプロジェクト体制，推進方法を比較し，5 章では実際の実装技術アーキテクチャの特徴を述べる．6 章ではシステムリリースの運営保守について比較し，7 章ではユニケージ開発手法の適応事例について紹介する．最後に，8 章で全体を総括する．

2. ユニケージ開発手法とデータ構造化手法の比較

2.1. ユニケージ開発手法の特徴

ユニケージ [7] は，UNIX[1]オペレーティングシステムの基本的なアーキテクチャと開発思想を基盤にしたデータ駆動型エンタープライズアーキテクチャとその開発手法を統合したフレームワークであり，企業システムの構築に特化している．[4, 5, 6] ユニケージはビジネスおよび業務設計，データ管理，アプリケーション開発，技術アーキテクチャの 4 つの主要構成要素から成り立っている．

ユニケージ開発手法は，これらの要素を実現するための一連の技法であり，特にアジャイル開発手法，トヨタ生産方式（TPS），およびリーン開発手法の特長を融合させている．この融合により，柔軟性，生産性，および品質の向上が達成される．

[1] "UNIX is a registered trademark of The Open Group"

ユニケージ開発手法の技術アーキテクチャの一つの特筆すべき特徴は，「入力」と「出力」によって整理された工程を UNIX の標準的なコマンドや USP 研究所によって作られたコマンドによる一連の処理によってそのまま実装することである．このため，開発期間が短縮され，データのみで結合するソフトウェアによってシステムが実現し，柔軟な「疎結合アーキテクチャ」のシステムとなる．さらに，UNIX/Linux の基本的なソフトウェアを活用することで「長く使い続けられるシステム」が構築できるのが特徴である．

2.2. 代表的な特徴の比較

「ユニケージ開発手法」が変化に対して迅速な対応ができる開発手法であることを説明するために， データ構造化手法との比較を行う．表 1 に示す主な比較対象とその代表的な特徴を中心に 3 章から 6 章にわたって説明を行う．

本論文の比較対象となるウォーターフォール型開発とリレーショナルデータベース（RDB）のスキーマに基づいて進める開発手法は，従来から広く行われてきた．この手法は，前工程へ後戻りをしない段階的な開発プロセスを採用し，データの初期設計を重視しデータの重複を排除する正規化を行うことで，計画的かつ段階的にシステム開発を進めることを目指している．具体的には、ウォーターフォールモデル型開発と RDB のスキーマに基づいた構造化データを組み合わせたアプローチであり，ここではこの手法を「データ構造化手法」と呼ぶこととして，以下比較を行う。

表 1 ユニケージ開発手法とデータ構造化手法の比較

比較対象	データ構造化手法	ユニケージ開発手法
開発手法	成果物を決められた期間内に確実に作ることに重点を置く手法	成果物が出来上がるまで顧客とともに伴奏し何度も試行錯誤し完成に近づける手法
設計・開発工程	工程間の後戻りをしない	工程間の後戻りができる
プロジェクト推進体制	上流で設計したものを下流で実装する	1人または少数精鋭のエンジニアがほぼすべての工程を担当する
実装技術	データベースモデル(RDB)を使用 データはDBMS等のミドルウェアで管理	ファイルリレーションを持たせたテキストファイル使用 データの管理方法はコマンドによって管理
運用保守体制	設計仕様通りに動作しているかを評価	開発後の事業の変化を評価

3. 開発手法と設計・開発工程の特徴

3.1. 開発手法の特徴

開発工程の特徴を「重視すること」「工程の進み方」「手法の目的」「開発工程」の 4 つの観点で整理し，データ構造化手法と比較した結果を表 2 に示す．

データ構造化手法では，決まった工程で確実に開発を進めることに重点を置くため， 開発工程の後戻りは行わない．後戻りを行う場合は最初から計画をし直すため時間とコストがかかる．

ユニケージ開発手法の場合は業務システムが事業・ビジネスに貢献することに重点を置き，確実に動くシステムを早く，安く作ることを目的するため要件を確認と並行して実装と依頼者による確認を繰り返しながら開発を進行する． ここで開発を繰り返し行える要因として，ユニケージ開発手法が手法と技術アーキテクチャが一体になっていることが挙げられる．[4]

つまり， 開発を確実に慎重に進めたい場合のデータ構造化手法に対して，仕様が不確実な場合の開発でシンプルさと早さ・安さの実現を追い求める手法がユニケージ開発手法である．

表 2 開発手法の比較

比較対象	比較のポイント	データ構造化開発手法の特徴	ユニケージ開発手法の特徴
開発手法	重視すること	決まった工程で開発を確実に進めることに重点をおく	業務システムが事業・ビジネスに貢献することに重点をおく
	工程の進み方	工程間の後戻りはほとんどしない	工程間の後戻りは許可する
	手法の目的	システムを慎重に作る	確実に動くシステムをシンプルに、安く、早くつくる
	開発工程	要件定義 基本設計 詳細設計 プログラミング 単体テスト 結合テスト システムテスト リリース	要件をヒアリングしながら実装 依頼者による確認とヒアリングを繰り返し行う ビジネスに最も重要で価値がある機能から先に開発する 周辺の機能は次のフェーズで開発する 開発段階で単体テストが終了 完成した機能からユーザーに提供し反応を確かめる

3.2. 設計・開発工程の特徴

設計工程の特徴を「エンジニアの関わり方」「アプリケーションの設計」「データ基盤の設計」「データ基盤の構造」「データ基盤の管理」「データの更新と削除」の 6 つ

の観点で整理し，データ構造化手法と比較した結果を表 3 に示す．

データ構造化手法では各設計工程にはアプリケーションの設計工程とデータ基盤の設計工程に分かれ，アプリケーションの設計を行う専門的な知識を持った人物とデータベースの設計を行う専門的な知識を持った人物がそれぞれの工程に割り当てられ，分業が発生する．具体的に発生する作業は表 3 の「データ基盤の設計」や「データ基盤の管理」に記している．

ユニケージ開発手法の場合はアプリケーションの設計もデータ基盤の設計も同じエンジニアで構成されるチームが担当する．データのハンドリングに用いる技術とアプリケーションの開発に用いる技術が統一されているため，エンジニアにとっては技術の学習や考慮する対象が少なく済む．例えば，複数のデータの組み合わせを用いて画面表示を行うアプリケーションが求められてもエンジニアはデータ基盤の構造も把握しているため素早く検証しその場でデモンストレーションをすることができる．

表 3 設計・開発工程の特徴

比較対象	比較のポイント	データ構造化開発手法の特徴	ユニケージ開発手法の特徴
設計・開発工程	エンジニアの関わり方	複数の設計工程がある 各設計工程をそれぞれの専門家が分業して担当する	エンジニアがすべての設計工程に関わる
	アプリケーション設計	仕様書に基づく実装	ヒアリングから編み出した業務の実態とデータ基盤構造に基づく実装
	データ基盤の設計	まず、RDBを設計 ２－３段に分けてデータが重複せず，CRUD処理にコストが少ないように正規化を行う 概念設計→論理設計→物理設計でシステムに必要なデータモデルを作成し、DBMSに実装する	ユニケージのデータ基盤を設計する。依頼者が持つデータを業務で発生する作業の入出力に対応させる。 対象業務で発生する作業の入出力に必要なデータを根源データまで辿って確認・収集しそのままシステムに保管した上で、整理データを設計する
	データ基盤の構造	RDBMSを利用する データはDBMS内で定義・保持されデータの利用はDBMSを経由する。データ管理はDBMSに任せる	生データを恒久保存するデータストアと、そのデータを様々な切り口で整理し下処理した大きく2つのブロックで構成される UNIXファイルシステム上のテキストファイル上にデータを保管する 実業務で発生するデータを時系列など比較的小さな単位で分類分けする。
	データ基盤の管理	処理前後のデータの整合性はRBDMSソフトウエアに任せる 整合性や一貫性などデータベース管理をアプリケーションから分離し効率的に行うためにDBMS(ミドルウェア)が利用される	整合性の確保は開発者のコマンドの処理方法に委ねられる DBMSを利用しない。必要となるDBMSの相当機能は開発者が実装する
	データの更新・削除	整合性確保や管理効率のため、論理設計の過程で”データの冗長性を排除する正規化”を行う	更新・削除はしない追記型のデータ構造なのでデータ重複を許容し、正規化はしない

4. プロジェクト推進・体制の特徴

プロジェクト推進とその体制の特徴を「全体的な体制」「クライアントのプロジェクト関与」「責任分担」「上流工程の役割」「下流工程の役割」「契約形態」「チーム体制」の 6 つの観点で整理し，データ構造化手法と比較した結果を表 4 に示す．

データ構造化手法では上流工程が設計したものを下流工程で実装することを基本とし，プロジェクト管理等はフェーズ毎に異なるそれぞれの専門エンジニアが担当するため分業が発生する．クライアントが関与するのは要件定義までで，開発完了までクライアントの関与は薄くなる．

ユニケージ開発手法では，エンジニアの役割はプロジェクトの段階によって異なるが担当をする人物は変わらず，どの段階であっても傍観者になることはない．エンジニアがすべてのフェーズに関わり続け，分業は依頼者の業務単位で行われる．依頼者もチームの一員として設計も開発も，終始プロジェクトに関わり続ける．

データ構造化手法では大人数で各工程をそれぞれの専門家が分業するが，ユニケージ開発手法は少数精鋭で一丸となって取り組む．

5. 実装技術の特徴

実装技術の特徴を「データ管理の実装方法」「データストアの構造」「データ層とアプリケーション層」「開発中の仕様変更」の 4 つの観点で整理し，データ構造化手法と比較した結果を表 5 に示す．本項で述べるアプリケーションとは主に Web ブラウザにて閲覧・操作が可能なアプリケーションを指す．

データ構造化手法では， ミドルウェアを利用または連携した実装を行う．整合性維持等のデータ管理とパフォーマンスの最適化はミドルウェア側が行う．アプリケーション開発はデータ管理用の API 等の手続きを基にしてアプリケーションに必要なデータ操作を実装する．この実装方法では， データ層とアプリケーション層が分断され，データ設計者とアプリケーション開発者は完全に分業している．確定したデータ仕様に基づき開発が行われているので変更は想定されておらず，仕様変更の場合は前の工程に遡ってやり直す必要がある．

ユニケージ開発手法では RDBMS のような構造化データを管理するミドルウェアは一切使用せず，データ管理の API は設けない．すなわち，データ管理の実装方法はアプリケーション開発時のコマンドによる処理方法に

委ねられ，テキストファイルを基本としたデータに対して直接読み書きを行う．データ管理とアプリケーション構築に用いる技術がユニケージで統一されているので，エンジニアがデータの仕様変更からアプリケーションへの反映も対応することができる． 実装技術を習得する観点においては，データ構造化手法はデータを管理する言語 (SQL 等) とアプリケーションを開発する言語 (java，C 等) が別々であるため，それぞれの技術を学びその上で学習後もそれぞれの技術の最新を追い続ける必要がある．一方，ユニケージ開発手法ではデータハンドリングを行うツールとアプリケーションを開発するツールが同一で，長い歴史のあるシェルコマンドを言語として用いているので学習が容易であり，開発プロジェクトに参画するまでの期間がデータ構造化手法に対して比較的短い．

表 4 プロジェクト推進・体制の特徴

比較対象	比較のポイント	データ構造化開発手法の特徴	ユニケージ開発手法の特徴
プロジェクト推進・体制	全体的な体制	上流工程が設計したものを下流工程で実装する	役割は工程によって異なるが担当をするエンジニアは変わらない
	体制と担当	プロジェクト管理/上流工程担当/下流工程担当などフェーズ毎に異なるそれぞれの専門エンジニアが担当する	各エンジニアがすべてのフェーズに関わり続け、依頼者の業務単位で分業
	クライアントのプロジェクト関与	依頼者は要件定義に強く関与する 要件確定後は開発チームがシステムを構築する。開発完了までクライアントの関与は薄くなる	依頼者もチームの一員として開発に関わる 依頼者もチームの一員として設計から開発、検証まで終始プロジェクトに関わる
	責任分担	全体的な責任は上流設計者にある 要件定義の責任は依頼者・開発チームの両者にある プロジェクト完遂の責任はプロジェクトマネージャにある 各工程の責任は工程のリーダーが責任を負う	責任はチーム全体が持つ エンジニアの役割はプロジェクトの段階によって変わるが傍観者になることはない
	上流工程の役割	上流工程では担当者は依頼者の同意をとることに専念する 要件とスコープを確定し合意する 開発仕様を確定する	上流と下流に分けない 依頼者の確認と実装を繰り返す
	下流工程の役割	下流工程では担当者は仕様書に従い実装する 仕様に基づきシステムを構築する	
	契約形態	上流工程は準委任、下流工程は請負	完成形準委任 出来具合を確認しながら目的達成に貢献できるシステムを実現する
	チーム体制	大人数で分業して進める 各設計工程をそれぞれの専門家が分業して順に担当する	少数精鋭で分業しない

表 5 実装技術の特徴

比較対象	比較のポイント	データ構造化開発手法の特徴	ユニケージ開発手法の特徴
実装技術	データ管理の実装	データの実装は、RDBの設計をそのままRDBMSにセットし、アプリケーション開発にAPIを提供する ミドルウェアを利用してデータベースを管理する データベースに対する操作はSQLで行う	読み書きにミドルウェアを使用しない APIは設けない DBMSを利用しない アプリケーションが直接データ管理とデータ操作を行う
	データストアの構造	DBMS内部に隠蔽されたデータの塊として扱われる 整合性やパフォーマンスの観点で最適化される	テキストファイルを基本とする
	アプリケーションの実装	アプリケーション開発はデータ管理のAPIをもとにアプリケーションを開発するのみ SQLによりデータベースにアクセスしてアプリケーションに必要なデータベース操作を実装する	ユニケージのコマンドでデータ管理と操作を行う
	ータ層とアプリケーション	データ設計・開発者とアプリケーション開発者は完全分業している データ層とアプリケーション層は異なるエンジニアが担当する	データ基盤設計者もアプリケーション開発者は同一人物 データ層とアプリケーション層は同じエンジニアが担当する
	開発中の仕様変更	確定したデータ仕様に基づき開発が行われているので、変更は想定されていない 仕様変更の場合は前の工程に遡ってやり直す必要がある	エンジニアがデータの仕様変更からアプリケーションへの反映も対応することができる

6. 運用保守体制の特徴

運用保守体制の特徴を「システムの評価」「不適合や不満への改修」「システムの改修」「総括して評価する期間」の 4 つの観点で整理し，データ構造化手法と比較した結果を表 6 に示す．

データ構造化手法では成果物が設計仕様通りに動作しているかが重要視され，要件定義で合意したシステム機能が実現できているかを評価する．成果物の不適合や不満に対して，改修を行うには設計からやり直す必要があり，新規開発と同じ対応となることもある．

ユニケージ開発手法では，成果物を導入した後の事業が成功しているかを重要視する．成果物の不適合や不満への改修には都度対応し，運用と並行して行う．ユニケージ開発手法は，アプリケーションとデータが密接に依存していない疎結合のアーキテクチャとなっているため [5]， 一から設計を行わなくても必要な部分にのみ改修を施すだけで対応が完了する．

表 6 運用保守体制の特徴

比較対象	比較のポイント	データ構造化開発手法の特徴	ユニケージ開発手法の特徴
運用保守	システムの評価	設計仕様通りに動作しているかをみる 要件定義で合意したシステム機能を実現できているかを評価する	事業が成功しているかを見る
	不適合や不満への改修	後からの変更や改善するためには設計からやり直す必要がある。 契約不適合責任(旧瑕疵担保責金)がある場合は改修等に応じるが、要件定義での合意内容がその基準になる 要件変更や改善要望への対応は新規開発と同じ対応となることもある	疎結合のアーキテクチャなので必要な部分のみに改修を施し, 柔軟に新しい機能を足すことができる。
	システムの改修	新規開発として対応する	完成後に手放すのでなく、要望に応じてその都度機能を追加する
	総括して評価する期間	開発全体的に時間がかかる 要件に基づき工数を算出しスケジュールを見積るプロセスをとる 手順を確実に順番に進めるため時間がかかるプロセスになることが多い	臨機応変に対応しながら進める ユーザの要望が発散し収集がつかない場合はかえって時間がかかってしまう

7. ユニケージ開発手法の適応事例

7.1. 適応事例

実際にユニケージ開発手法を導入して開発を行ったシステムの適応事例とその後の影響について説明する. ユニバーサルジョイント部品を製造する協和工業株式会社では 2018 年から 2021 年までの間にユニケージ開発手法を含んだビジネスアーキテクチャを適応し, 業務システム開発と業務効率の改善に取り組んだ. [6] 同社は人手不足やリードタイムの長期化といった課題を抱えており, 従来型の製造業向けのパッケージソフトウェアを導入したが課題の解決には至らなかった. USP 研究所と共に 3 年間に渡り企業に伴走して何回も作り直しを行った結果, 各種センサーと連携した製造ラインの自動化や見える化が行われ生産性と効率性が向上した. 従来のパッケージソフトウェアに合わせて行っていた業務が, システム側が業務に合わせて改良をし続けるようになった. 生産計画・進捗・実績等が目で見えるようになり, 今まで多くの時間を割いていた伝票や在庫を調べる作業時間も無くなった. システム構築の内製化や IT 人材の育成など企業全体の DX 化にも貢献し, 情報処理推進機構(IPA)による DX 認定制度の認定事業者と経済産業省の「DX セレクション 2024」の DX 優良事例[8]に選定された.

7.2. 技術的なベンチマーク

ユニケージ開発手法の大規模データに対する処理スピードの速さと, 変化を反映させるための試行錯誤や検証のしやすさを示すために, テキストファイルとコマンドを基本としたデータ管理と RDBMS の場合で処理速度を比較する. 4 コアプロセッサ搭載のコンピュータで表 7 に示した構造のテーブル (TRANS: 全 5 フィールド 4, 000 万行, USERS: 全 9 フィールド 10 万行) を出力テーブル(全 11 フィールド) の結果を得る際の処理時間を MySQL, PostgreSQL, テキストファイル+コマンドの 3 つの技術の間で比較する. テキストファイル+コマンドの場合は同じ行列数のテキストファイル(TRANS, USERS) に対して加工処理を行うコマンドをパイプ(|)で接続し, 処理結果が出力されるまでの時間を計測する.

表 8 に実行したクエリ文やコマンドとその結果を示す. MySQL, PostgreSQL と比較しテキストファイルとコマンドの場合は結果が出るまでの平均処理時間が圧倒的に早い. 大規模データに対して単なる並び替えや結合, 集計といった処理が高速に行えるだけでなく, エンジニアが実際の大規模データを用いてデータの設計や開発方針変更に伴う処理の組み換えを行う際の迅速な検証や試行錯誤を行うことが可能である. 仕様変更や機能追加の要望に対して顧客の目の前で処理時間の検証や, デモンストレーションをすることもできる.

表 7 比較に用いたテーブル構造

	フィールド										
	1	2	3	4	5	6	7	8	9	10	11
TRANS 約2GB 40,000,000 行	Transaction date	Account number	Transaction amount	Branch ID	Branch name						
	YYYYMMDD	VARCHAR 10	BIGINT	VARCHAR 10	VARCHAR 32						
USERS 約15.3GB 100,000 行	Account number	User name	Industry ID	Industry name	Average salary index	Salary	Phone number	Email address	Address		
	VARCHAR 10	VARCHAR 64	VARCHAR 10	VARCHAR 90	FLOAT	BIGINT	VARCHAR 10	VARCHAR 64	VARCHAR 32		
出力テーブル	Account number	Transaction month	Total transaction amount	User name	Industry ID	Industry name	Average salary index	Salary	Phone number	Email address	Address
	VARCHAR 10	YYYYMM	BIGINT	VARCHAR 64	VARCHAR 10	VARCHAR 90	FLOAT	BIGINT	VARCHAR 10	VARCHAR 64	VARCHAR 32

表 8 実行した処理と出力までの時間

	実行したクエリ(コマンド)	処理時間 (秒)
MySQL	SELECT T.account, T.txn_month, T.sum_amount, U.name, U.industry_id, U.industry_name, U.salary_index, U.salary, U.phone, U.email, U.address FROM users U INNER JOIN (SELECT account, date_format(txn_date, '%Y%m') AS txn_month, sum(amount) as sum_amount FROM transactions GROUP BY txn_month, account ORDER BY account, txn_month) T ON U.account = T.account INTO OUTFILE '/var/lib/mysql-files/OUT.MYSQL' FIELDS TERMINATED BY ' '	325
PostreSQL	COPY (SELECT T.account, T.txn_month, T.sum_amount, U.name, U.industry_id, U.industry_name, U.salary_index, U.salary, U.phone, U.email, U.address FROM users U INNER JOIN (SELECT account, to_char(txn_date,'YYYYMM') AS txn_month, sum(amount) as sum_amount FROM transactions GROUP BY txn_month, account ORDER BY account, txn_month) T ON U.account = T.account) TO '/tmp/OUT.PSQL' WITH CSV DELIMITER ' ';	133
テキストファイル+コマンド (シングルコア)	self 1.1.6 2 3 TRANS | # TRANS の第1、2、3フィールドを抽出 msort key=1n@2n | # 第1ソートキーを第1フィールド、第2ソートキーを第2フィールドにし昇順ソート sm2 1 2 3 3 | # フィールドの集計 msort key=2n | # 第2フィールドで昇順ソート self 2 1 3 | # 第2, 1, 3フィールドを抽出 join1x key=1 - USERS > OUT.UNI # 第1フィールドをキーにしてUSERSファイルと結合し OUT.UNIに出力	18.17
テキストファイル+コマンド (4コア)		13.17

8. まとめ

「ユニケージ開発手法」について，データ構造化手法との比較を行いながら特徴を説明した．ユニケージ開発手法が，短い時間で解決策を検証でき，事業の変化にも柔軟な対応力を持ち開発手法と技術アーキテクチャが一体であることを示した．ユニケージ開発手法は分業をせずに，確実に動くシステムをシンプルに安く早く作ることができる．従来のデータ構造化手法の方式では機転を利かすことが困難であった成果物の改修や，その後の事業そのものの改善にも繋がる手法であることを示した．今後の展望について，これまでにユニケージ開発手法を導入しシステム開発を行ったユースケースに対して定量化と評価を行う必要がある．

謝辞

本論文の作成にあたり，適応事例の紹介にてご協力頂いた協和工業株式会社様には厚く御礼申し上げます．

参考文献

[1] Thomas Delaet， The power of pace in technology，https://mck.co/49JCqPC ， 2024
[2] 狩野 英司， 行政機関におけるアジャイル型開発の導入に関する調査研究， 行政&情報システム 2020 年 10 月号， pp48-53， 2020
[3] 山本 佳和， 山本 修一郎， 製造業に必要なアジャイル開発についての考察， 第 27 回知識流通ネットワーク研究会， SIG-KSN-027-04， 2020
[4] 當仲寛哲， S. ブヤンジャルガル， 鈴木明夫， 山本修一郎， データ駆動型ユニケージアーキテクチャの提案， 知識流通ネットワーク研究会， SIG-KSN-031-04， 2022
[5] Nobuaki Tounaka，Shuichiro Yamamoto，Buyanjargal Shirnen，Data Driven Process-Based Unicage Architecture，KES2023， 2023
[6] Nobuaki Tounaka， Shuichiro Yamamoto，Buyanjargal Shirnen， Unicage Architecture Development Method， CENTRIS 2023， 2023
[7] 當仲寛哲，山崎裕詞，熊谷章，熊野憲辰，木ノ下勝郎，ユニケージ原論，ユニバーサル・シェル・プログラミング研究所
[8] 協和工業，経済産業省「DX セレクション 2024」優良事例に選定， https://www.kyowa-uj.com/news/notice/848/ ， 2024