

ローコード開発におけるデータ項目設計に関する考察

市原 和幸† 藤原 一樹† 北嶋 淳† 落合 昌樹† 宮本 由美†

株式会社 日立製作所‡

1. はじめに

開発効率の向上を目的として、ローコードツールの活用が注目されている。制約はあるが、プログラム作成の一部を自動化できるため、スクラッチ開発と比較して開発効率だけでなく、品質確保の観点でもメリットがある。開発プロセスに関しては、顧客ニーズの迅速かつ柔軟な実現のため、アジャイル型開発が適している場合がある。短いサイクルで部分的な機能開発を繰り返すことで、顧客と認識共有できるからである。ローコードツールの活用によって、コーディング品質を担保したプログラムを早期に顧客へ提供できるため、対話を重視するアジャイル型開発にローコードツールの活用は有効である一方、それぞれの特徴を鑑みた品質の検討が必要である。

2. 考察の対象

ローコードツールの活用により、コーディングによる不良の発生を大きく削減できる。つまり、品質不良の作り込みは設計に依存するため、設計の品質確保が重要となる。設計で発生する品質不良として、要件定義に起因する不良、設計で作り込む不良に分類する。今回は設計で作り込む不良に重点を置き、設計の共通化について考える。ここで述べる設計の共通化とは、複数のチームや作業員で設計書など成果物の記述が統制されることを指す。通常、ソフトウェア開発の設計では、基準に則って成果物を作成することで、複数のチームや作業員を統制して、設計の共通化を図る。顧客の要件を短いサイクルで繰り返し開発するアジャイル型開発では、成果物の変更回数や変更規模も多量と想定できるため、アジリティだけでなく、設計品質の観点も重要である。本論文ではローコードツールを活用したアジャイル型開発において、複数チームで設計するケースを対象に、想定されるリ

スクから設計を共通化する施策の検討と評価をする。

3. 想定されるリスク

システムを複数チームで設計する場合、共通的な設計情報の管理が重要となる。システムにおける共通的な設計情報には、画面レイアウトやデータベースのカラムとなるデータ項目などが挙げられる。データ項目には、サブシステム固有の情報だけでなく、サブシステム間で共通したデータ項目も存在する。複数チームでの開発は、データ項目の表記ゆれとして、シノニム（異名同義語）、ホモニム（同名異義語）が発生しやすい。シノニムやホモニムは、品質低下のリスクを招く原因となる。データ項目の統制に関する先行研究としては、要件定義工程における要件の一貫性検証の提案がある[1]。アジャイル型開発では、データ項目の修正が開発中に発生するため、品質低下のリスクも大きくなると予想できる。そのため、システムとしてデータ項目を管理し、設計の共通化を図る必要がある。

そこで、我々はデータ中心アプローチ(DOA)の活用を検討した。DOAは、保守性と効率性の向上を目的として、業務で扱うデータを中心に、設計の共通化を図ることができるため、エンタープライズ系システムの開発手法として活用されている[2]。ローコードツールによるアジャイル型開発においても、設計品質や成果物管理の重要性は同じであるため、DOAによるデータ項目の設計と管理について施策を検討する。

4. 施策

想定されるリスクに対する施策をDOAに基づいて検討する。

4.1. 設計プロセスの整備

複数チームでデータ辞書を共通化した設計プロセス運用を検討する。運用手順を以下に記す。各チームの作業員を業務担当者とし、業務担当者とは別に、データ辞書を管理する担当者を置く。データ辞書管理者はデータ辞書の運用が軌道に乗るまで専任の作業が望ましい。データ項目に関する設計の作業フローを図1に示す。業

Consideration on the Design of Data Items in a Low-Code Development Process

† Kazuyuki Ichihara, Kazuki Fujiwara, Jun Kitajima, Masaki Ochiai, Yumi Miyamoto

‡ Hitachi, Ltd

務担当者は作業対象の設計書に着手したら、まず、対象の設計書に必要なデータ項目が、データ辞書に存在するかを確認する。データ項目がすでに存在する場合は、設計書の作成・修正を実施する。データ項目が存在しない場合は、データ辞書管理者にデータ項目の追加・変更を申請する。データ辞書管理者は業務担当者からの申請内容を確認し、申請内容に問題なければ、データ項目を追加・変更する。データ辞書管理者はデータ項目の追加、もしくは申請内容の不備を申請元の業務担当者に連絡する。業務担当者は申請結果に不備の指摘がないかを確認し、不備があった場合は設計書を修正し、データ項目の追加を再度申請する。不備がなければ、設計書の作成・修正をする。

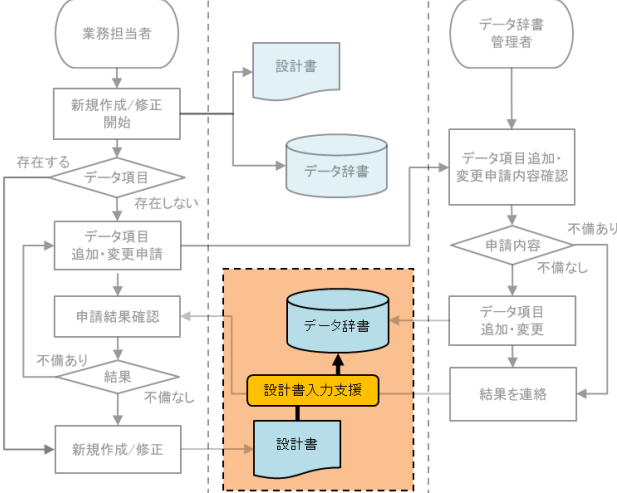


図 2 設計書の入力支援

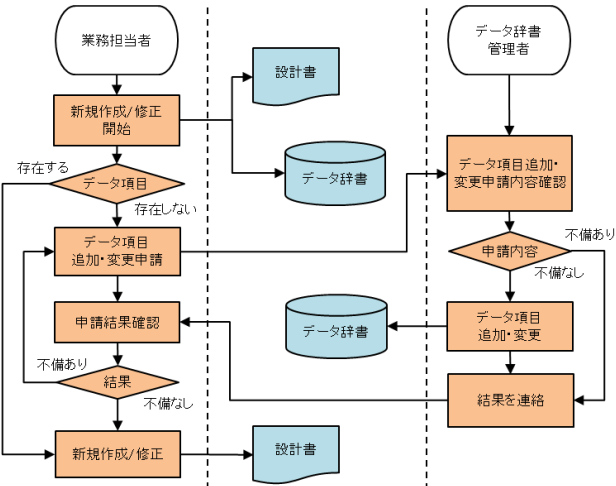


図 1 データ辞書の運用

データ辞書管理者のみがデータ辞書を編集する設計プロセスとしたことで、データ項目の表記ゆれを防止できる。また、設計書の修正回数や規模に比例して、データ項目の表記ゆれのリスクも高くなるため、適用効果は高くなると予想する。

4.2. 設計書の入力誤り防止

データ項目に関する設計プロセスの整備により、複数チームで設計の統制ができる一方、業務担当者の記述誤りは防止できない。そのため、データ辞書を参照した設計書入力支援ツールの適用を検討する。設計書入力支援を追加したプロセスを図 2 に示す。必要なデータ項目をツール上で選択して、設計書に自動入力することで、業務担当者の手入力による記述誤りを防止できる。

5. 評価

ローコードツールを活用した A 案件の基本設計における摘出不良を対象に、本施策で想定される適用効果を評価する。不良原因および対策内容を確認して、施策適用により防止可能な不良件数を推測する。結果は表 1 に示すとおり、本施策の適用で約 7.3%の不良を防止できると考える。

表 1 A 案件 施策適用の想定効果

No.	不良原因	不良件数	防止件数
1	設計誤り	54	0
2	設計不十分・不統一、基準不足	278	25
3	記述ミス	149	10
-	合計	481	35

6. まとめ

ローコードツールを活用したアジャイル型開発で、DOA によるデータ項目設計の共通化とその想定効果について考察した。一方で、アジャイル型開発の利点であるアジリティの考察まではできていない。今後は、生産性の評価も実施し、品質と効率性のバランスについて検討を深めていく。

参考文献

[1] 位野木万里, 近藤公久, 『要求仕様の一貫性検証支援ツールの提案と適用評価』, SEC journal Vol. 13 No. 1, 2017.
[2] 高橋まゆみ, 南波則孝, 『C/S 情報系システムへの DOA 適用』, 情報処理学会研究報告ソフトウェア工学 (SE), 1997.