

ノーコード・ローコード開発を成功に 導くための「ノー&ローコード虎の巻 」の提案

Suggestion of "No & Low Code Toranomaki" for the success of no-code/low-code

第38年度（2022年度）ソフトウェア品質管理研究会
研究コース1 ソフトウェアプロセス評価・改善

「No & Low-codeチーム」

リーダー：内田 泰介（株式会社日立ソリューションズ・クリエイト）

研究員：池田 幸子（株式会社日立システムズ）

古里 透（株式会社アズビル）

主 査：山田 淳（株式会社東芝）

副 主 査：田中 桂三（オムロン株式会社）

アドバイザー：中森 博晃（パナソニック スマートファクトリーソリューションズ株式会社）

目次

- **研究選定の理由**
- **ローコード、ノーコード開発の定義と現状の動向仮説の立案**
- **ローコード、ノーコード開発の先入観**
- **ローコード、ノーコード開発の実態**
- **ローコード、ノーコード開発の虎の巻**
- **虎の巻の評価方法**
- **問題事例と対応する虎の巻の紹介①～③**
- **おわりに**

研究選定の理由

我々研究コース1（Bチーム）のメンバーは
全員品質保証部門に所属しています。

最近ローコード・ノーコード開発が流行っていますが、
これに対する品質保証のプロセスが良くわかっていませんで
した。

開発現場にヒアリングしてみましたが、期待と実態のギャッ
プがあり、品質保証もままならない状況と感じました。

そこで我々はローコード・ノーコード開発でも適切に品質保
証ができる留意点をまとめた「ロー&ノーコード虎の巻」を
作成しました。

これを設計者に捧げることで、品質保証し、QAもそれに基づ
いた品質保証活動をしていってほしいと思っています。

ローコード、ノーコード開発の定義と現状の動向

ノーコード・ローコード開発とは？

可能な限りソースコードを書かずパラメータの設定や GUI 操作により Web サイトやアプリケーションを開発すること。

最近、事例が増えてきており注目されています。



ローコード、ノーコード開発の先入観

ローコード・ノーコード開発に対して
以下の先入観を持たれている

経営者の過剰な期待

(短期開発、低コスト、高品質)

開発者の期待

(高度な設計技術不要、テスト不要)

保守担当の期待

(開発対象減に伴う作業負荷の軽減)

・・・本当なのか？



ローコード、ノーコード開発の実態

**ローコード・ノーコード開発に対して
先入観と異なる意見もある**

過剰な期待となっていないかを考えたため

**開発現場や有識者へヒアリングを行い
実態の把握をした**

・・・留意点を纏めるのはどうだ？



ローコード、ノーコード開発の虎の巻

留意点をまとめたもの

⇒ 虎の巻は

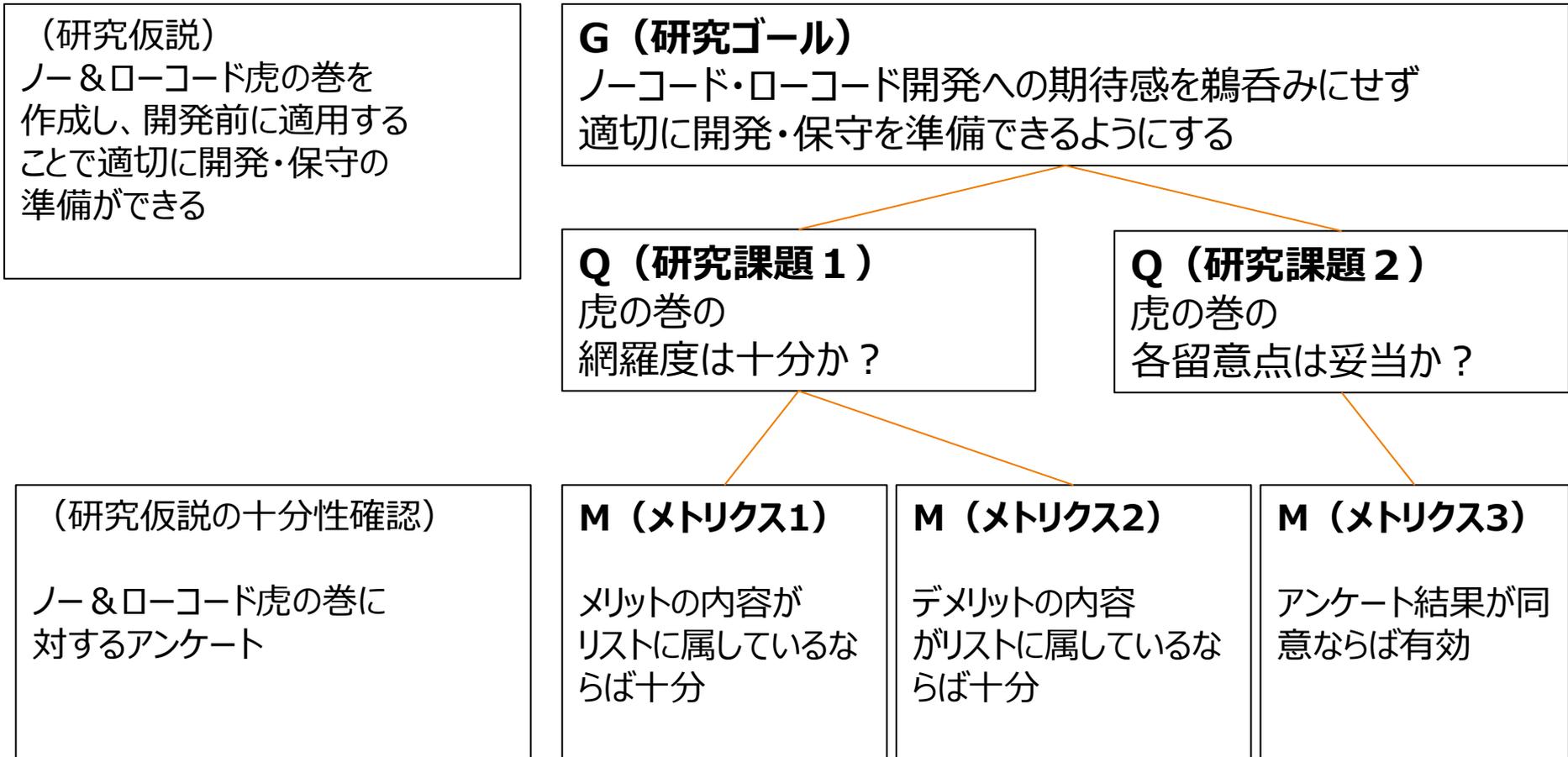
こんな効果を期待しています

- ・ 過剰な期待感の抑制
- ・ 留意点をチェックし対策をすることで
ローコード・ノーコード開発の
メリットを生かした
開発保守作業が実現できる



虎の巻の評価方法

虎の巻の評価はGQM(Goal Question Metrics)手法を利用



虎の巻の評価結果

評価結果、

網羅性 . . . 概ね問題なし

妥当性 . . . 概ね問題なし

| ノーコード・ローコード開発の留意点 | 1 | | 2 | | 3 | | 4 | | 5 | | 点数 |
|---|-----|----|----|----|----|----|----|----|----|-----|----|
| | 10点 | 5点 | 1点 | 0点 | 0点 | 0点 | 0点 | 0点 | 0点 | | |
| ノーコード開発であっても、実現したい業務フローやユースケースを作成する必要がある。 | 27 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 9.4 | |
| 従来の開発と同等に製品に合わせたビジネスモデルや業務フローを提案できる人がいること。 | 23 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8.7 | |
| 利用する製品のSLAがどの範囲となっているか認識しておく必要がある。 | 22 | 7 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 8.2 | |
| データモデルや構造設計を明確に理解している必要がある。 (例：同じ意味で異なる名称のデータがあると、複雑になる) | 17 | 11 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 7.3 | |
| 担当者によって製品理解度はバラツキが出てしまうため、それなりの設計力のあるメンバを配置する必要がある。 | 20 | 5 | 3 | 0 | 0 | 3 | 0 | 0 | 0 | 7.4 | |
| パッケージに標準的に備わっている機能のみで要件は満たせるか否かを確認する必要がある。 (満たせていない場合は、それなりの要員を配置する必要がある) | 23 | 6 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 8.4 | |
| すべてのテストが自動で実施されるものではなく、単体、結合テストは実施する必要がある。 | 22 | 3 | 2 | 0 | 0 | 4 | 0 | 0 | 0 | 7.6 | |
| ブラックボックスのテストまでしかできない為、障害発生する恐れがある。よって保守に備えて、ログ出力や調査要求などの契約が必要。 | 20 | 8 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 7.7 | |
| 互換性がなくなった場合のリスク対策が必要。 | 23 | 6 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 8.4 | |
| 製品のVersionUp時に発生する影響の見極めが必要。 | 23 | 6 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 8.4 | |
| 販売実績のない製品や販売終了後のサポート有無が不明な場合、製品が使えなくなるリスクを踏まえた上で採用している。もしくはそのようなソフトは選定の対象外としている。 | 21 | 6 | 2 | 0 | 0 | 1 | 0 | 1 | 0 | 7.8 | |
| 複数人で開発する場合は構成管理を実施する必要がある。 | 22 | 8 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 8.4 | |
| 海外の製品を利用する場合は、使用許諾を十分理解し、許諾を遵守する必要がある。 | 22 | 6 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 8.1 | |
| 海外の個人情報については、各国の法令を遵守し、現地のサーバーで管理する。 | 20 | 5 | 2 | 0 | 0 | 3 | 0 | 0 | 0 | 7.3 | |
| 有料プランを利用する際は、逆にコストが高くなる場合がある。また、自由度がない分、逆に改造の手間がかかりコストが高くなる可能性がある。 | 20 | 8 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 7.7 | |
| 高度あるいは特別な要求・品質確保に100%対応できるものではなく、知識・期間の範囲内で、要求対応・品質確保を行うことを前提にしていることを、ステークホルダーと合意できている。 | 21 | 9 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 8.3 | |

出来上がった虎の巻は以下のもの

ノー＆ローコード虎の巻

| # | SWEBOKの知識領域 | ノーコード・ローコード開発における期待感 | ノーコード・ローコード開発の失敗事例 | ノーコード・ローコード開発の留意点 |
|---|-------------|--|--|---|
| 1 | ソフトウェアの要求 | 期待感A 業務用の製品であるため、ある程度のことは意識せずとも何でもできる | ■開発事例からのフィードバック 帳票添付機能が使える製品であったが、添付件数は1帳票のみで複数帳票は添付することができず、顧客の要望とあわななかった | (会社独自のローカルルールをやめるなど) ローコード・ノーコード開発に合わせた、業務フローの見直しも必要 |
| 2 | ソフトウェアの要求 | | | 留意点B 従来の開発と同等に製品に合わせたビジネスモデルや業務フローを提案できる人がいること |
| 3 | ソフトウェアの要求 | | ■他研究コースからフィードバック 顧客とのSLAによって、製品側の問題であってもこちら側の問題になることがあった | 留意点C 利用する製品のSLAがどの範囲となっているか認識しておく必要がある |
| 4 | ソフトウェア設計 | 期待感B 設計不要 開発初心者でもOK | ■他研究コースからフィードバック お試して、ローコードで開発を実施したが、データモデルを設計せずに進めたため、汎用性のないものが出来上がってしまった ■他研究コースからフィードバック データ構造適正化ができておらず、影響分析のワードが定まらないことや、認識齟齬によるトラブルになる事例もあった。 | ある程度の設計知識は必要であり、開発初心者だけでは難しい |

具体的な内容を何点か紹介します

問題事例と対応する虎の巻の紹介①

問題のあった事例①

先入観：どんな仕様でも簡単に実現できる



実態：顧客の要求は複数帳票の添付、
開発ツールの仕様制約で単一帳票のみしか添付
できないことが開発途中で判明（PJ中断）

虎の巻：要求定義段階で実現したい業務フローや
ユースケースを設計し、重要な要求の見落としを
防ぐべし！



場合によっては業務フロー自体を見直すべし

問題事例と対応する虎の巻の紹介②

問題のあった事例②

先入観：保守不要であり、ログの提供なども不要と判断

実態：ローコード・ノーコードで作成した部分に障害が発生し、解析の為にログが必要になり、改めて自分達でログを取得する追加作業が生じた

虎の巻：ローコード・ノーコード開発でも障害が発生するものと考え、
契約せよ！
(ログ出力を成果物に含める)



問題事例と対応する虎の巻の紹介③

問題のあった事例③

先入観：開発が不要なので当然保守も不要だろう！

実態：システムのGUIはローコード・ノーコード開発、ロジックは別環境（Java）で開発していたローコード・ノーコード製品のVerUpにより、内部データの持ち方が変わり、Java開発部分は全て作り直す必要となった。

虎の巻：ローコード・ノーコードの保守作業を侮るなかれ。互換性がなくなる場合のリスク対策をとるべし！
(別環境で開発時のI/F設計を計画化するなど)



ローコード、ノーコード開発への展望

**ここまで否定的な意見を多く紹介しましたが、
ノーコード・ローコード開発を否定している訳ではなく、
正しく利用されることで期待通りのを願っている**

**虎の巻を使うことで
開発者の負担軽減、
顧客要求のタイムリーな対応など
良いことはいっぱいある**

**自分の会社の開発メンバーにも共有します。
皆さんも利用してみてください。**

謝辞

本論文の執筆にあたり、
ご助言・ご協力をいただきました皆様に
深謝いたします。

- 研究コース2(レビューチーム)
および 細川さん
- 研究コース5(AIチーム)

- 研究コース1(プロセス改善チーム)
- 指導員



ご清聴ありがとうございました