

FP を用いた LCP/NCP とプロコードの損益分岐分析 および改修案件生産性テーラリング手法の提案

A Proposal of FP-Based Break-Even Analysis between LCP/NCP and Pro-Code Development, and a Productivity Tailoring Method for Modification Projects

フューチャーアーキテクト株式会社 クライアントバリューグロースグループ
Future Architect, Inc. Client Value-Growth Group

○長坂 昭彦

○Akihiko Nagasaka

Abstract: In LCP/NCP (low-code/no-code platform) development, decision-making on development-method selection and cost estimation for modification (enhancement) projects remains largely unstructured. This paper proposes two quantitative frameworks. First, a Total-Cost-of-Ownership (TCO) break-even model that compares LCP/NCP and pro-code development, using system scale (function points, FP), number of users, license period, and productivity as parameters, and derives a decision map for development-method selection. Second, a productivity tailoring method for modification projects that adjusts the FP-based productivity baselines established for new development by process-phase ratio and change-impact degree. Both frameworks aim to reduce variability in development-method selection and effort estimation across a system's lifecycle, contributing to more reproducible, quantitative project decision-making.

1. はじめに

近年、ローコード・ノーコード開発（以下、『LCP/NCP 開発』）の普及に伴い、プロコード開発との適用方式の判断が複雑化している。従来は初期開発コストを中心とした比較が主であったが、ライセンス費用や保守費用を含めたトータルコスト（TCO）での評価が必要となっている。また、開発の主流が新規構築から改修へと移行する中で、改修案件における生産性基準が未整備であり、見積りは個別判断に依存し精度のばらつきが生じている。

これらの課題を解決するため、IT システム可視化協議会（以下、「MCIS」）のローコード・ノーコード開発研究会（以下、「lcncSig」）では、会員組織の実績データや有識者の知見をもとに検討を進めてきた。本発表では、①TCO に基づく開発方式選択（LCP/NCP とプロコード）、②改修案件に適用可能な生産性テーラリング手法、の 2 点を提示し、実務における意思決定および見積りを定量的かつ再現可能な形で実施するための枠組みを示す。これにより、属人的な見積りや開発方式選択に起因する判断ばらつきを抑え、プロジェクト計画段階における品質リスクの低減に寄与することを狙いとする。

2. 課題と背景

2.1 開発方式選択における課題（LCP/NCP とプロコード）

LCP/NCP 開発とプロコード開発ではコスト構造が異なり、初期構築費用と継続利用費用（保守費、ライセンス費用）の性質が大きく異なる。LCP/NCP は開發生産性が高い場合が多く初期開発費を抑えられる可能性がある一方、ライセンス課金により利用者数・利用期間に応じて費用が蓄積するため、長期的にはコスト優位性が逆転する場合がある。従来はライセンス費用を含めない初期コスト比較が主流であり、TCO 観点での定量的な判断基準が存在しなかった。

2.2 改修案件における見積りの課題

lcncSig では FY22～FY24 の 3 年間の活動により、新規構築・再構築案件における FP ベースの生

産性基準値を一定程度確立した [1] [2] [3]。一方、近年は開発の主流が新規構築から改修へ移行しつつあるが、改修案件を対象とした基準値はまだ確立できていない。改修案件は、要件定義が影響調査中心、開発対象が追加・変更機能、テスト対象が原則全機能となるなど、新規構築とは工程ごとの見積り対象 FP の性質が異なるため、既存の基準値をそのまま適用することができない。

3. 本発表のアプローチ

前述の課題を解決するため、本発表では以下の2つの手法を提案する。

(1) FP を用いた LCP/NCP とプロコードの損益分岐モデル：ユーザ数・利用期間・開発生産性・システム規模 (FP) を主要パラメーターとした TC0 モデルを構築し、開発方式選択の判断基準を定量化する。

(2) 改修案件生産性テーラリング手法：新規開発で確立した生産性基準値を、工程別工数割合と変更影響度に基づき補正し、改修案件の工数算出に適用する。

いずれも実務における属人的な判断のばらつきを抑え、意思決定プロセスの再現性を高めることを目的とする。

4. 損益分岐モデルの構築と検証

4.1 モデルの評価対象と前提

損益分岐モデルの評価軸は、システムのライフサイクル (例：3～5 年) における累積コストとする。評価対象は初期構築費用および継続利用費用 (保守費用、ライセンス費用) とし、システム化によるコストメリットが期待できる 300～1,000FP を主な対象規模とした (300FP 未満は本モデルの主たる適用対象外とした)。損益分岐点は、LCP/NCP の累積コストとプロコードの累積コストが等しくなる時点として定義し、事業サイクルや資産償却期間等から定めた使用期間における累積コストの比較により、開発方式の優劣を評価する。なお、LCP/NCP ライセンス費用の課金方式はサブスク型 (ライセンス数×単価×利用期間) を前提とし、サーバ課金型は対象外とした。またインフラ費用や保守・運用の責任分界点等の要因も評価対象外としている。

4.2 コスト構造の整理

システム総コスト (TC0) を対象に、LCP/NCP、プロコードそれぞれのコスト内訳を図 1 の通り整理した。両方式とも開発費用・保守費用の構造を持つが、LCP/NCP はこれに加えてライセンス費用 (ライセンス数×ライセンス単価×利用期間) が発生する点が最大の相違点である。



図1 TC0のコスト構造比較 (LCP/NCP とプロコード)

4.3 パラメーターと TC0 算出式

整理したコスト構造に基づき、下記のパラメーターを用いて TC0 を算出する式を定義した (表 1、図 2)。LCP/NCP の FP 生産性は lcncSig の新規開発実績値 [3]、プロコードの FP 生産性は IPA が公開する生産性データ [4] 等を参考に設定した。なお、LCP/NCP の FP 単価は市場に公開された指標が存在しないため、プロコードの FP 単価 (G) と両方式の FP 生産性比 (F÷E) を用

いて間接的に算出する構成とした。

表 1 主要パラメーター

記号	名称	初期値提案	単位	備考
A	FP(システム総規模)	×	FP	
B	初期開発期間(LCP/NCP)	×	カ月	LCP/NCP で開発した場合の開発期間。(初期開発のライセンス費用算出用)
C	運用期間	×	年	
D	LCP/NCP ライセンス数	×	本	
E	FP 生産性(LCP/NCP)	△	FP/人月	lencSig の新規開発実績値より、総規模に応じた初期値を設定。
F	FP 生産性(プロコード)	○	FP/人月	IPA の新規開発生産性を活用し、1000FP 未満の生産性を設定。
G	FP 単価(プロコード)	○	k¥	lencSig 保有の実績データの N が一番多い 10~50 人月の単価
H	ライセンス単価	○	k¥	1 ライセンスあたりの月額
I	LCP/NCP 保守費率	○	%	LCP/NCP の初期開発費に対する割合。年額で設定。
J	プロコード保守費率	○	%	プロコードの初期開発費に対する割合。年額で設定。

初期値提案の凡例：

- × … 初期値として使用できるデータがありません。ご自身で設定が必要です。
- △ … lencSig の実績値を使用できます。規模に応じて使用する値の見直しが必要です。
- … IPA や市場の相場など、一般的なものを使用できます。

$$\begin{aligned}
 \text{TCO(LCP/NCP)} &= \boxed{A \times (G \times F \div E) + D \times H \times B} + \boxed{A \times (G \times F \div E) \times I \times C} + \boxed{D \times H \times C \times 12} \\
 \text{TCO(プロコード)} &= \boxed{A \times G} + \boxed{A \times G \times J \times C} + \boxed{\text{ライセンス費用}}
 \end{aligned}$$

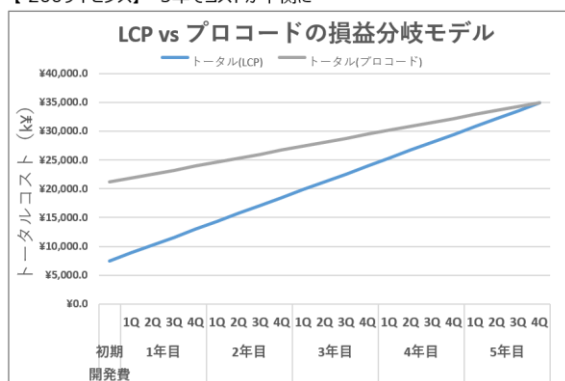
初期開発コスト 保守費 ライセンス費用

図 2 TCO 算出式

4.4 検証結果

規模 800FP、LCP/NCP の初期開発期間 8 か月、ライセンス単価 2,000 円/ユーザ・月、運用期間 5 年の条件で、ライセンス数の違いによるコスト差をシミュレーションした(図 3)。算出式の共通項を除くと、本シミュレーション条件では、コストへの影響が最も大きいのはライセンス料(ユーザ数×利用期間)であることが確認された。例えば 200 ユーザ規模では約 5 年で累積コストが均衡し、300 ユーザ規模では約 3 年でプロコードが有利に転じることを確認した。これにより、システム規模・ユーザ数・利用期間の組み合わせによって開発方式の優劣が変化することを明確化した。

【200ライセンス】5年でコストが均衡に



【300ライセンス】3年目でコストが逆転 (= 3年未満ではLCP/NCP有利)

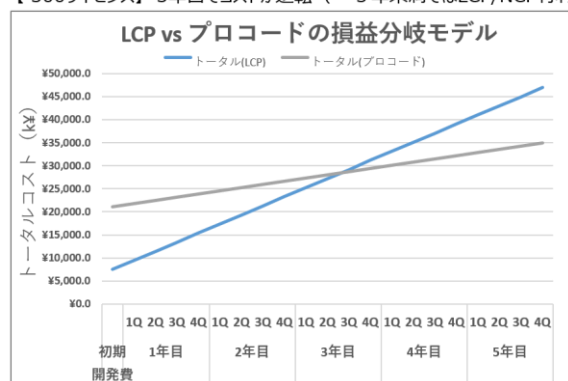


図 3 損益分岐シミュレーション結果 (200/300 ライセンスの比較)

4.5 適用判断マップ

検証結果を踏まえ、システム規模 (FP) とライセンス数を軸とした開発方式選択の判断マップを作成した(図 4)。FP 規模が小さい、利用者数が少ない場合はモデルの算出結果から LCP/NCP が優位となり、逆に FP 規模・利用者数が多い場合はプロコードが優位となる傾向が確認された。なお、早期立ち上げ・改修頻度・業務要件の複雑性については、本 TCO モデルの直接的な評価変

数ではなく、lcncSig の実務知見に基づく定性的な判断要素として付加している。また、300FP 未満かつ利用者数 20 人以下の領域は、本モデルの主たる適用対象外とし、Excel や SaaS 等も含めた代替手段を優先的に検討する領域とした。

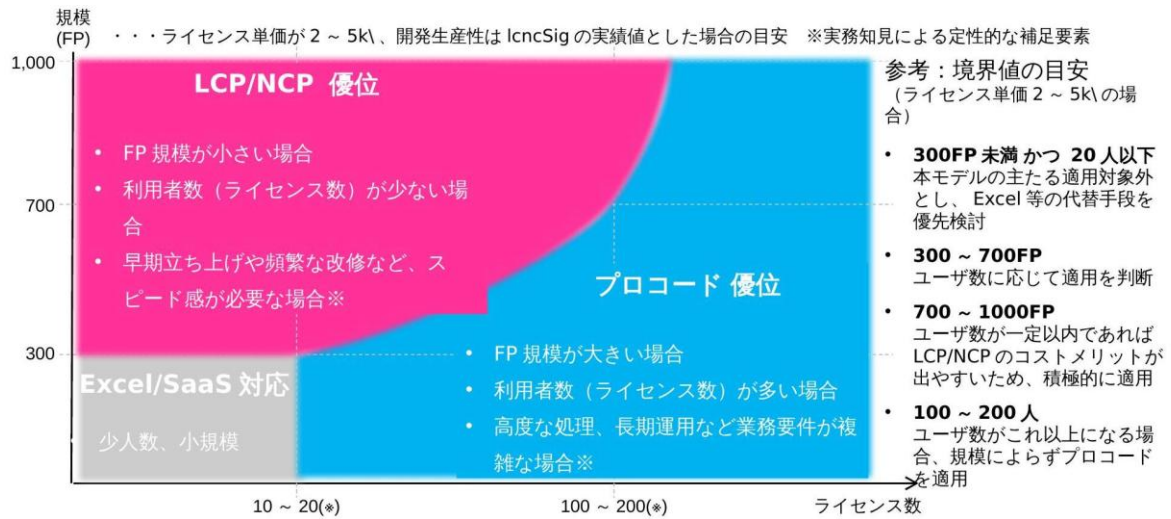


図4 損益分岐モデルによる開発方式選択の適用判断マップ

5. 改修案件生産性テラリング手法

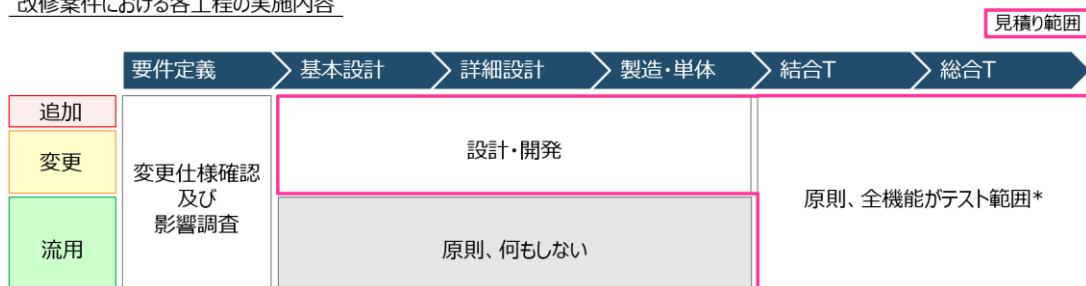
5.1 アプローチ

改修案件の実績データは事例が少なく、新規・再構築のように実績収集による基準値確立には数年を要すると予想される。そこで本発表では、既に確立している新規構築の生産性基準値 [3] をテラリングして改修案件に適用する手法を検討した。改修案件は工程ごとに見積り対象となる FP 範囲が異なるため、開発工程とテスト工程を分離して工数を算出する点が特徴である。

5.2 改修案件の基本的な考え方

改修案件では、要件定義は変更仕様確認および影響調査が中心となり、基本設計から製造・単体までの開発対象は追加・変更機能に限定される (流用機能は原則対応不要)。一方、結合テストおよび総合テストでは、流用機能を含む原則全機能がテスト範囲となる (図 5)。このため、開発工程とテスト工程で見積り対象 FP を切り替える必要がある。

改修案件における各工程の実施内容



*流用機能はどこまでテストすべきかは後工程で論点となるが、要件定義終了時点ではテスト計画は未作成な事が多く、テスト工数を精緻化する事は難しいため、ここでは全量テストを前提とした工数見積りとする

図5 改修案件における各工程の実施内容と見積り範囲

5.3 開発工程別工数割合

実績データのうち工程別工数割合が判明している 26 件 (新規開発 18 件、再構築 8 件) を分析した結果、新規開発は基本設計～製造・単体 (前 3 工程) の割合が高く、再構築は基本設計・詳細設計の割合が高い傾向が確認された。本手法は新規開発の生産性基準値をベースにテラリングするため、新規開発の工数割合を採用し、前 3 工程・後 2 工程 (結合テスト・総合テスト) の工数割合の中央値をそれぞれ 71.5%、28.5% として採用した (図 6)。

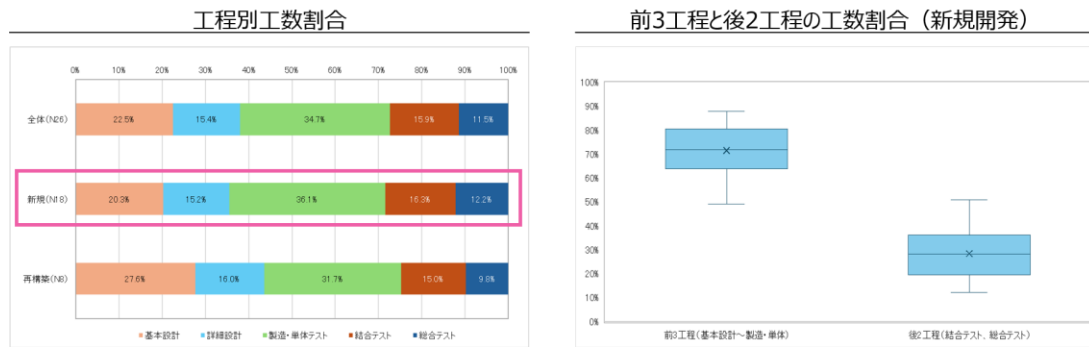


図6 開発工程別工数割合（新規開発）

5.4 基本設計から実装・単体までのテーラリング

対象工程を基本設計～製造・単体に限定することで生産性は上昇するため、5 工程の基準生産性を工程割合（0.715）で除算して補正した（表2）。

表2 開発工程（基本設計～製造・単体）テーラリング後の生産性基準値

FP 規模	生産性中央値（5 工程）	生産性中央値（開発 3 工程）
300FP 未満	10.13 fp/人月	14.17 fp/人月
300FP 以上 700FP 未満	13.03 fp/人月	18.22 fp/人月
700FP 以上	37.83 fp/人月	52.91 fp/人月
全体	15.32 fp/人月	21.43 fp/人月

5.5 変更影響度を加味した開発工数補正

変更開発が多い場合は、新規（追加）と変更を FP 割合で工数按分し、変更開発工数はさらに影響度合いに応じて補正する。影響度は、新規追加機能を 100%とした場合の相対工数係数として定義する。画面・帳票の UI/UX 変更のみの場合は影響度 20%、入力検査や四則演算の変更を含む場合は影響度 30%、テーブルのカラム追加・変更やデータアクセスの変更を含む場合は影響度 40%と仮定し、基本設計から実装・単体までの工数に影響度を積算する（表3）。複数分類に該当する場合は、最も影響度の高い分類を適用する。この影響度は lcncSig 参加企業の実務経験に基づく暫定値である。

表3 変更影響度に応じた工数補正の例

変更機能の分類	FP 割合による工数按分	影響度	補正後の工数
UI/UX の変更のみ	4 人月	20%	0.8 人月
入力検査／四則演算を含む	6 人月	30%	1.8 人月
テーブル／データアクセスを含む	6 人月	40%	2.4 人月

5.6 結合テストから総合テストまでのテーラリング

同様に、結合テスト・総合テストの対象工程割合（0.285）で基準生産性を補正し、テスト工数を算出する（表4）。テスト対象は原則全機能であるため、テスト FP には流用機能を含む全 FP を用いる。

表4 テスト工程（結合テスト～総合テスト）テーラリング後の生産性基準値

FP 規模	生産性中央値（5 工程）	生産性中央値（テスト 2 工程）
300FP 未満	10.13 fp/人月	35.54 fp/人月
300FP 以上 700FP 未満	13.03 fp/人月	45.72 fp/人月
700FP 以上	37.83 fp/人月	132.74 fp/人月
全体	15.32 fp/人月	53.75 fp/人月

5.7 全体工数算出と検証

開発工数（追加・変更）とテスト工数（全機能対象）を合算して全体工数を算出する。算出した全体工数は WBS 見積等、他の見積り手法で算出した工数と比較し妥当性を確認する補助指標として活用する。

5.8 改修案件の工数算出例

改修案件生産性テーラリング手法の適用例として、全体規模 1,000FP（追加 200FP、変更 300FP、流用 500FP）を対象に工数算出の流れを示す（図 7、工数は原則として小数第一位を四捨五入した整数人月で表記するが、変更影響度による補正結果は小数第一位まで表記する）。開発工程は開発対象 FP（追加+変更=500FP、300FP 以上 700FP 未満）、テスト工程は全機能 FP（1,000FP、700FP 以上）と、工程ごとに異なる規模区分を参照する点に注意されたい。開発 3 工程の生産性中央値 18.22fp/人月を 500FP に適用し開発工数 27 人月を算出、FP 割合で追加 11 人月・変更 16 人月に按分する。変更開発分は UI/UX のみ（4 人月・影響度 20%）、入力検査・四則演算含む（6 人月・30%）、テーブル・データアクセス含む（6 人月・40%）に分類して積算し、16 人月を 5.0 人月に補正する。テスト工数は全機能 1,000FP にテスト 2 工程の生産性中央値 132.74fp/人月を適用し 8 人月と算出する。以上より、追加 11 人月+変更（補正後）5.0 人月+テスト 8 人月=全体工数 24 人月となる。算出結果は WBS 見積等と比較し、差異が許容範囲内かを確認する補助指標として活用する。

生産性中央値（5工程）

300FP未満

10.13_{fp/人月}

300FP以上700FP未満

13.03_{fp/人月}

700FP以上

37.83_{fp/人月}

全体

15.32_{fp/人月}

生産性中央値（開発3工程）

300FP未満

14.17_{fp/人月}

300FP以上700FP未満

18.22_{fp/人月}

700FP以上

52.91_{fp/人月}

全体

21.43_{fp/人月}

工数算出（例）

全FP

200

300

500

開発FP

500_{fp}

開発工数

27_{人月}

凡例：

追加

変更

流用

工数：追加/変更案分

開発工数

27_{人月}

追加開発

11_{人月}

変更開発

16_{人月}

変更開発：影響度合いに応じて機能分類

UI/UXのみ

4_{人月}

入力検査/四則演算含

6_{人月}

テーブル/データアクセス含

6_{人月}

変更開発（補正前）

16_{人月}

工数算出（例）

影響度はIcncSig参加企業の実務経験をもとにした暫定値

影響度小：20%

0.8_{人月}

影響度中：30%

1.8_{人月}

影響度大：40%

2.4_{人月}

変更開発は既存機能を利用するため、追加と比べて開発工数が小さくなる

変更開発（補正後）

5.0_{人月}

生産性中央値（開発5工程）		生産性中央値（テスト2工程）		工数算出（例）			
300FP未満	10.13 _{fp/人月}	300FP未満	35.54 _{fp/人月}	全FP	200	300	500
300FP以上700FP未満	13.03 _{fp/人月}	300FP以上700FP未満	45.72 _{fp/人月}	テストFP	1,000		
700FP以上	37.83 _{fp/人月}	700FP以上	132.74 _{fp/人月}	テスト工数	8 _{人月}		
全体	15.32 _{fp/人月}	全体	53.75 _{fp/人月}	凡例：	追加	変更	流用
開発工数モデル見積り（例）				2点見積りによる妥当性確認（イメージ）			

追加開発	11人月
変更開発	5人月
テスト	8人月
全体	24人月

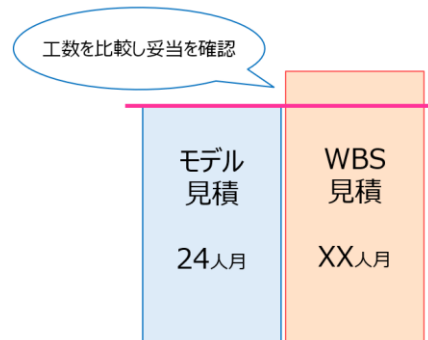


図7 改修案件の工数算出例

5.9 改修案件見積りの適用手順

以上の手順を整理すると、STEP1～STEP5の5段階として示すことができる（図8）。なお、本手法はLCP/NCPに限らず、プロコードの改修やパッケージのアドオン改修等、他の開発方式にも適用可能である（ただし、基準生産性および工程別工数割合は、対象となる開発方式の実績に応じて個別に設定する必要がある）。

STEP1 対象FPを分類	対象機能を追加、変更、流用に分類し、後続の見積り対象FPをそれぞれ確定する
STEP2 追加開発工数算出	新規開発における基準生産性を工程別割合で補正（$\div 0.715$）し、追加開発部分の工数を算出する
STEP3 変更影響補正	変更開発部分はUI変更（20%）、入力検査等（30%）、DB変更等（40%）の割合で積算し、影響度合いに応じて工数を調整する
STEP4 テスト工数算出	新規開発における基準生産性を工程別割合で補正（$\div 0.285$）し、全体機能に対するテスト工数を算出する
STEP5 全体工数算出	- 開発工数とテスト工数を加算し全体工数を算出する - 算出した全体工数は他の見積り手法で算出した工数と比較し、妥当性を確認して確定する

図8 改修案件見積りの適用手順（STEP1～STEP5）

6. 活動の振り返り

6.1 損益分岐モデルの構築と検証

本モデルにより、LCP/NCP とプロコードの適用判断において、システム規模（FP）、ユーザ数、利用期間が主要な決定要因であることを定量的に示すことができた。特に、本シミュレーション条件ではライセンス費用（ユーザ数×利用期間）がコストへの影響として最も大きいことが確認され、従来の初期コスト中心の判断から、TCO に基づく合理的な判断への転換の一助となる。一方、LCP/NCP の生産性については lcncSig の実績値を初期値として使用しているため、規模や業務特性に応じた見直しが今後も必要である。本発表内容については MCIS 会員からも、「既存システムのリプレース検討時に損益分岐シミュレーターを活用したい」「システムライフサイクル全体でのコストを考えるきっかけになる」といった声が寄せられ、実務での活用可能性について前向きな反応が得られている。開発方式の選定に悩む現場に対する具体的な検討の出発点としても有効であることが確認できた。

6.2 改修案件生産性テラリング手法

新規開発で確立した生産性基準値をテラリングすることで、限られた時間・情報の中でも改修案件における工数算出の枠組みを構築できた。新規開発案件 18 件の分析に基づく工程別工数割合（71.5%/28.5%）を基準値として採用しており、統計的な標本数としては限りがあるものの、追加・変更・流用を区別した工数算出手法として一定の実用性が期待できる。なお、変更影響度（20%/30%/40%）は実務経験に基づく暫定値である。予実比較による精度検証も試みたが、改修案件の実績データが十分に集積されなかったため実施には至らず、今後の課題として残っている。また、MCIS 会員からは、「kintone での改修案件の見積りが難しかったが、アプローチ方法を知ることができてよかった」など工数見積りの考え方自体は有用との評価を得た一方で、「見積り手法が複雑で実務で使いこなすのが少し難しい」との意見も寄せられており、実務適用性を高めるための手法の簡素化が今後の検討課題である。

7. 今後の展望

lcncSig は 2022 年度から 2025 年度までの 4 年間にわたり活動を行ってきたが、一定の成果を上げたため、2025 年度をもって研究会としての活動を終了した。本発表で紹介した損益分岐モデルおよび改修案件生産性テラリング手法を含む活動成果は、MCIS の Web サイトに対象となる成果物名を公開している（<https://www.mcis-jp.org/ローコード-ノーコード開発sig>）。また、各年度の詳細な成果物（実績データの分析結果、各種ガイド、シミュレーター等）は MCIS 会員専用サイトにて公開しているため、興味のある方は同協議会へお問い合わせ頂きたい。なお、改修案件生産性テラリング手法の予実比較検証については、lcncSig としての実績データ収集の枠組みが終了したため実施には至らなかったが、著者個人の実務適用の中で継続して検証していきたい。また、テラリング手法についても、実務適用性を高める簡素化を含め、引き続き改善を図っていきたい。

参考文献

- [1] ソフトウェア品質シンポジウム 2023 「ローコード開発プロセスについて」
<https://www.juse.jp/sqip/library/download/index.cgi/B1-1.pdf?id=S2023-IBa4>
- [2] ソフトウェア品質シンポジウム 2024 「ローコード・ノーコードの製品選定の勘所と生産性・品質基準について」 <https://www.juse.jp/sqip/library/download/index.cgi/A1-4.pdf?id=S2024-M06o>
- [3] ソフトウェア品質シンポジウム 2025 「ローコード・ノーコードの開発生産性とテスト密度及び ODC タイプ属性による機能テストの完了判定について」
- [4] 独立行政法人情報処理推進機構（IPA） 「ソフトウェア開発分析データ集」（入手可能な最新版を参照）