

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

Low-Code/No-Code Development Productivity and Test Density, and A

Framework for Functional Test Completion Based on ODC

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 response to the recent needs of DX and the shift to in-house production, low-code no-code development (hereinafter, “LCP/NCP development”) is becoming one of the options for business system development. In such a trend, the demand for LCP/NCP development is increasing year by year, however, some issues have emerged.

For example

- There are many LCP/NCP products available, but no viewpoints, methods, and procedures for selecting the right product have been established.
- LCP/NCP development lacks established testing standards, making system quality inconsistent.
- LCP/NCP development is no correlation between the number of features and the number of defects, making it difficult to determine when functional testing is complete.

In order to solve these issues, the Measurement Council for IT Systems, Japan (hereinafter “MCIS”) established lcncSig, a study group of member volunteers in LCP/NCP development in FY2022. Based on MCIS member organizations' performance data and the knowledge of experts, we have been working to solve problems by estimation using the function point method, quality evaluation, and standardization of the development process. We hope this paper will be of some help for LCP/NCP development.

1. はじめに

近年の DX ブームや内製化シフトのニーズを受けてローコード・ノーコード開発（以降、『LCP/NCP 開発』）が業務システム開発の選択肢のひとつとなりつつある。そのような潮流のなか、LCP/NCP 開発の需要は年々増加傾向だが一方で課題も顕在化してきた。例えば、

- 工数見積り手法が確立されていないため、見積り精度にばらつきが出やすい
- 必要なテスト基準が確立されていないため、システム品質にばらつきが出やすい
- 機能数と不具合数に相関がないため機能テスト完了を判定しづらい

が挙げられる。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」）は 2022 年度にローコード・ノーコード開発における会員有志による研究会「lcncSig」を設立した。会員組織の実績データや有識者の知見をもとに、ファンクションポイント法による見積り、品質評価、開発プロセス標準化など、課題解決に取り組んできた。本発表が LCP/NCP 開発の普及・発展の一助となれば幸いである。

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

〒141-0032 東京都品川区大崎 1-2-2 アートビレッジ大崎セントラルタワー

1-2-2, Osaki Shinagawa-ku, Tokyo, 141-0032 Japan Tel: 050-5305-8428 email: a.nagasaka.5b@future.co.jp

【キーワード：】 MCIS、ファンクションポイント、ローコード、ノーコード、LCP、NCP、simple FP、FP 簡易推定法、テスト密度、不具合密度、ODC 分析、タイプ属性

2. 課題と背景

2.1 見積り精度にバラツキあり

機能の複雑度や品質要件によって LCP/NCP 開発工数は数人日から数人月と大きく変動する。1 機能の開発に数か月を要する事もあるが、『1 機能 X 人日』など、一律見積りを行うと過小見積りとなりスケジュール遅延を招く。要因として以下が挙げられる。

- (1) 選定製品がミスマッチ
開発者に製品を使いこなせるスキル・リテラシがなければ、期待する生産性を発揮する事はできないが、開発組織のケーパビリティに合致した選定となっていない場合がある。
また、製品が保有する機能や特性を見誤ると、仕様変更や拡張コードの追加等、想定外の対応が後に必要になる場合がある。
- (2) 見積りプロセスが未整備
通常システム開発とは異なり、LCP/NCP 開発は市民開発の名のもと EUC (End User Computing) の一環として実施される場合がある。その場合、要件確定後に工数・期間の確定見積り(概算見積りからの再見積り)を実施するプロセスとなっていない場合がある。
- (3) 開発初期段階での FP 算出が困難
開発初期段階は要件自体が曖昧だが、予算や体制確保のためには一定精度で概算を見積る必要がある。従来の FP 算出方法では基本設計レベルの情報が必要となり、開発初期段階での FP 算出が困難な場合がある。
- (4) 信頼できる工数見積りの基準値欠如
開発初期段階で FP が算出できたとしても、 $\text{工数} = \text{FP} \div \text{生産性}$ のため、工数を算出するには生産性の基準値が必要だが、信頼できる基準値が存在しない。

2.2 システム品質にバラツキあり

LCP/NCP 開発では標準コンポーネントを多く利用し実装するため、通常のスクラッチ開発と比べて必要となるテストの量や質を過小に評価し品質劣化しやすい。結果、同等の品質要件のシステムであってもプロジェクトにより品質にバラツキが生じる。要因として以下が挙げられる。

- (1) LCP/NCP に即した開発知見が不十分
通常の開発と異なり、LCP/NCP に即した開発上の留意点があるが、LCP/NCP 開発の経験が発注者/委託者双方に不足しているとやらなくても良いタスクを実施している場合や、逆に怠ってはいけないタスクを怠っている場合がある。
- (2) プロコード開発との差異が不明瞭
開発プロセスにウォーターフォールを採用した場合、通常のプロコード開発と LCP/NCP 開発に大工程レベルでは差異はないが、各工程で実施するタスクや作成物、その作成方法は異なるため、差異が不明瞭なまま開発を進めると品質の低下を招く場合がある。
- (3) 信頼できるテスト基準値の欠如
LCP/NCP 開発の場合、プログラムが初めから部品化され、且つ拡張コードも製品専用のリポジトリに隠蔽されるため正しいステップ数が分からない場合が多い。結果、従来の SLOC ベースのテスト基準(テスト密度)を利用できない場合がある。
- (4) LCP/NCP に即したテスト完了基準が不明瞭
LCP/NCP 開発の場合、機能量と機能テストの不具合数に明確な相関がない事が lencSig の実績調査で判明している。(SQiP2024 にて発表済み)^[1]機能テストの完了条件が曖昧となり、結果、不具合検知が遅れリリース遅延や本番障害に繋がる場合がある。

3. 解決策

前述の課題と要因、及び検討した改善施策を列挙する。今回は多くのプロジェクトにおいてその活用が見込めると『FP ベースの生産性基準確立』『FP ベースのテスト密度確立』『ODC による機能テスト完了基準確立』について後述する。（『LCP に即した工程定義の明確化』『LCP に即した開発知見の文書化』は SQiP2023、『製品選定プロセスの確立』『FP 簡易推定法の確立』は SQiP2024 にて発表済み^[1]）

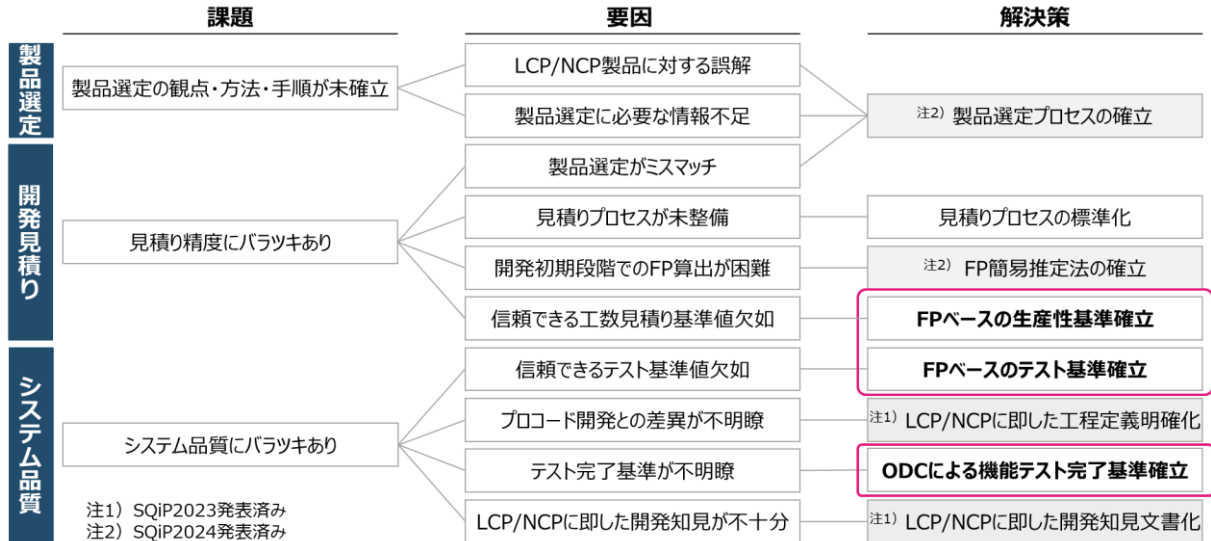


図 1 LCP/NCP 開発における課題・要因・解決策

4. 課題解決に向けた仮説と実施アクション

『FP ベースの生産性基準確立』『FP ベースのテスト基準確立』は LCP/NCP 開発の実績 FP、工数、結合テストケース数を集計・分析する事で、ローコード開発に適合した FP ベースの生産性や機能テスト密度の基準値を確立できるのではないかとこの仮説を策定した。『ODC による機能テスト完了基準確立』は不具合の”量”ではなく”質”に着目し、ODC 分析の手法を活用する事で LCP/NCP 開発に適した機能テストの完了判定が確立できるのではないかとこの仮説を策定した。仮説を検証するため経済調査会の協力も得て、予備調査と本調査を行いデータ分析や検討を実施した。予備調査は東証プライム上場企業の大手企業を中心に利用製品やその用途、課題認識などを得て、本調査は過去からの累積分とあわせて 33 案件の開発実績データを得た。

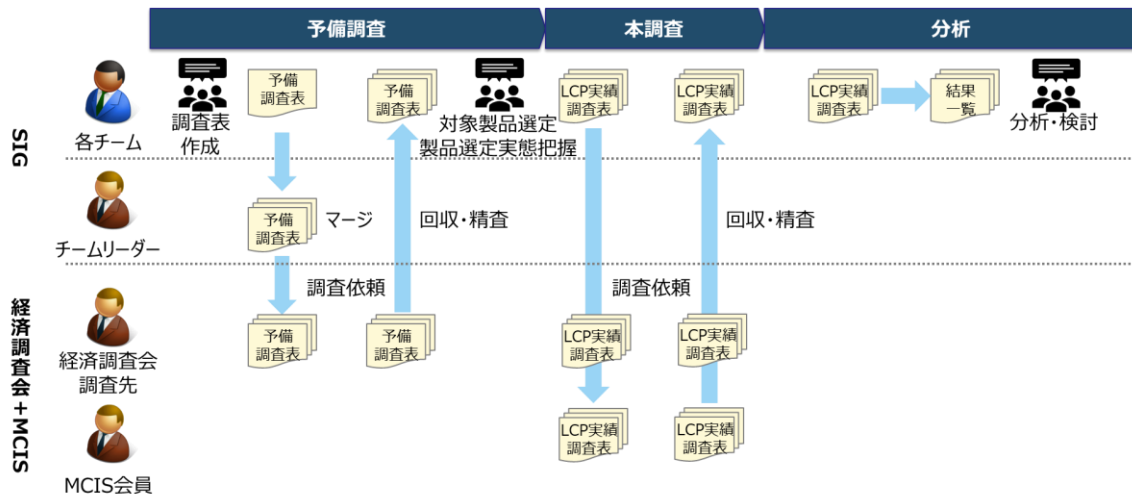


図 2 仮説検証フロー

4.1 FP ベースの生産性基準確立

本調査にて取得した 33 件を対象に基準値の算出を行った。業種は製造業が一番多く、次いで情報通信その他サービスであった。また、サービスレベルは 8h5d、利用ユーザは社内中心であった事から社内システムでの利用が中心と考えられる。

＜本調査依頼時の有効データの定義（一部抜粋）＞

- ・ 予備調査シェア上位の 14 製品利用、且つウォーターフォール型の新規構築または再構築
- ・ 開発総工数が概ね 10 人月以上、且つ稼働後一定期間に業務影響のあるアプリ障害が未発生
- ・ 要件定義完了時点の計画からリリース日が+1 か月未満、工数が+20%未満

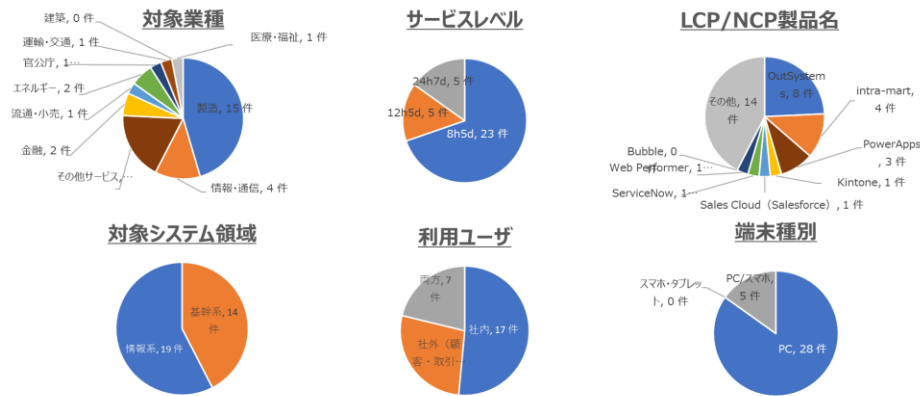


図3 収集した実績データのプロファイリング結果

対象データ 33 件のうち、FP は約 8 割が 1,000FP 未満、工数は約 8 割が 55 人月未満であった。FP は FP 簡易推定法にて算出し、工数は規模と相関のある開発 5 工程（基本設計～システムテスト）を対象として集計した。

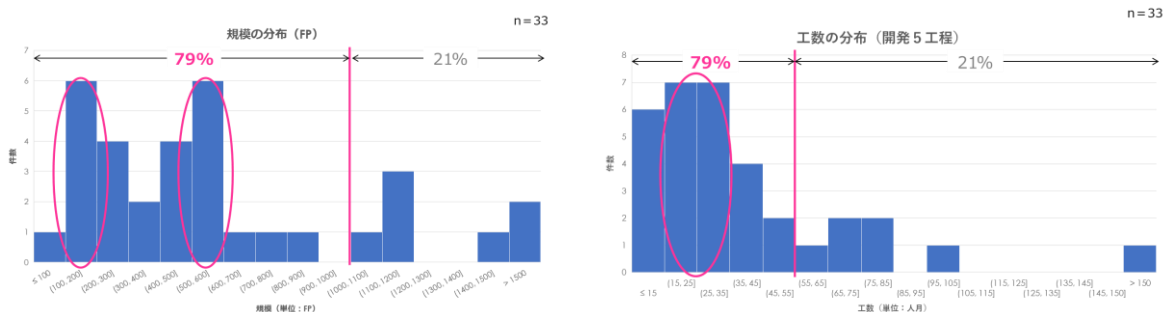


図4 実績データ FP 規模と工数分布

実績データから欠損データ 3 件を除いた 30 件の FP 生産性を層別した結果、700FP 以上になると生産性が急激に高くなる一方、ばらつきも大きくなる事が判明した。また、新規構築と再構築に比較すると、新規構築の生産性がより高く、ばらつきも大きい事が判明した。再構築は標本数が増加したことにより、生産性のばらつきが小さくなったことから、FP 生産性の精度は高まったものとする。

なお、LCP/NCP 開発においては FP 規模と開発工数の相関が一見弱く見えるが、規模を層別する事で相関が明確になり実務で利用できる生産性基準値となる事から、昨年度から層別している。一定規模の案件では部品の再利用が進むため、生産性は高まるものとする。

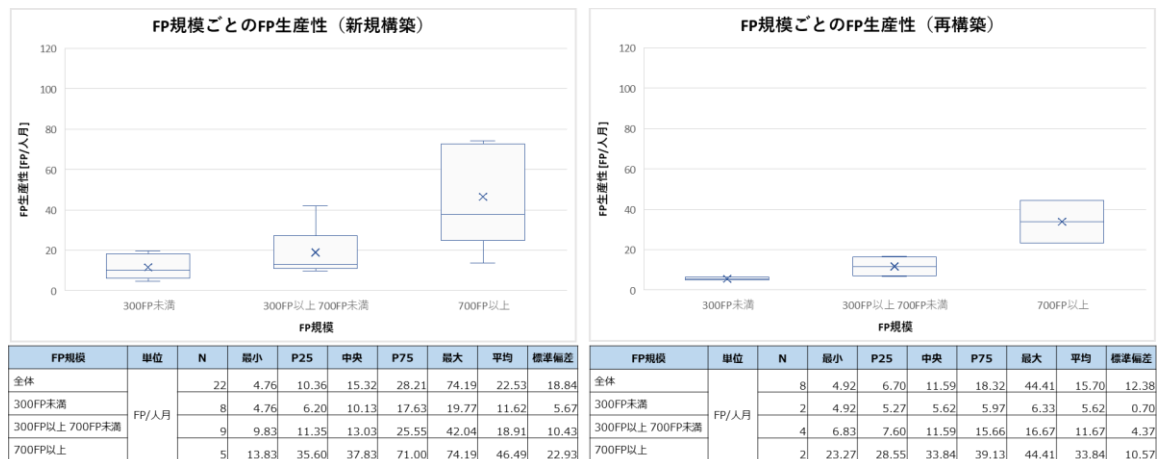


図5 新規構築と再構築のFP生産性

4.2 FPベースのテスト基準確立

取得した33件から未回答等の3件を除外した30件を対象に、結合テストケース密度を分析した。分布を確認すると、大きく3つに分かれており複数の集合が混在していると考えた。今回はテスト密度中の集合（300FP以上）とテスト密度高（300FP未満）の19件を対象とした。FP規模に着目し『300FP』という境界でテスト密度の分布が分かれることを発見したため、分析では規模の層別を実施した。なお、小規模案件では機能間の結合度が高く、必要となるテストケースは多くなるものとする。

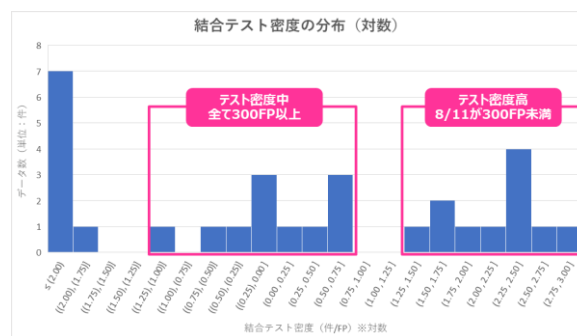


図6 結合テスト密度の分布（対数変換後）

標本数の増加に伴い、再構築のテスト密度のバラツキは大きく改善した。また、開発規模が同等の場合、新規と再構築の間にはテスト密度の有意差は無かった。よって、新規と再構築のいずれの場合も実績収集した案件と同程度の品質を確保するには相応の結合テストケース数が必要と考える。

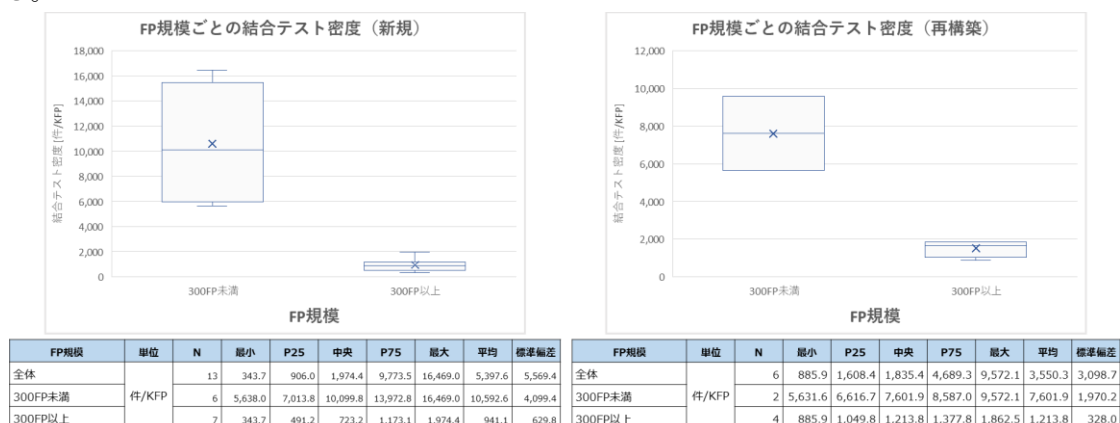


図7 結合テスト密度

4.3 ODC による機能テスト完了基準確立

取得した 33 件から機能テストの不具合に関する有効回答のあった 7 件のデータを ODC 分析^[2]の手法を用いて分析した。実績調査では定量的な不具合数の把握は困難であったため、ODC にて定義されている各タイプ属性（1. 値の選択～8. 開発関連ドキュメント）の不具合割合を高・中・低で回答してもらい、高を 100pt、中を 50pt、低を 1pt としてポイント化したうえで、総計 1,387pt に対する各タイプ属性のポイント割合を比率として算出した。

実回答

	1. 値の選択	2. 条件分岐	3. アルゴリズム	4. タイミング・順序	5. インタフェース	6. 機能性	7. ビルド・パッケージ・結合	8. 開発関連ドキュメント
高	1	1	1	1	1	3	0	0
中	1	4	0	0	4	1	1	0
低	5	2	6	6	2	3	6	7

ポイント化

	1. 値の選択	2. 条件分岐	3. アルゴリズム	4. タイミング・順序	5. インタフェース	6. 機能性	7. ビルド・パッケージ・結合	8. 開発関連ドキュメント
高	100	100	100	100	100	300	0	0
中	50	200	0	0	200	50	50	0
低	5	2	6	6	2	3	6	7
小計	155	302	106	106	302	353	56	7
総計								1387pt

比率算出

	1. 値の選択	2. 条件分岐	3. アルゴリズム	4. タイミング・順序	5. インタフェース	6. 機能性	7. ビルド・パッケージ・結合	8. 開発関連ドキュメント
比率	11%	22%	8%	8%	22%	25%	4%	1%

図 8 機能テスト不具合に対する実績調査結果、ポイント化、比率算出結果

算出した比率を” LCP/NCP の機能テストにおける不具合比率の期待モデル”とし、書籍『ソフトウェア不具合改善手法 ODC 分析』^[2]P. 101 に紹介されていた結合テスト（機能テスト）のタイプ属性割合を” プロコードの機能テストにおける不具合比率の期待モデル”と仮定し比較した。

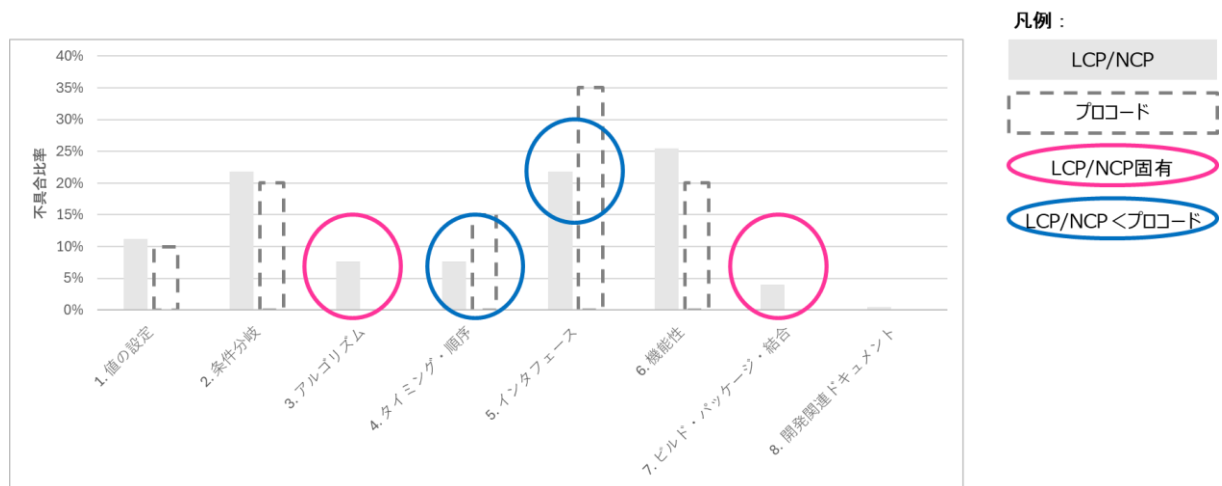


図 9 LCP/NCP とプロコードの機能テスト不具合モデル比較

結果、不具合割合に差異があり、「アルゴリズム」と「ビルド・パッケージ・結合」は LCP/NCP 固有の不具合タイプ、「タイミング・順序」「インタフェース」はプロコードに比べて LCP/NCP は少なくなるものと判断した。「LCP/NCP 固有」は不具合発生を受入れたうえで機能テストにて品質担保すること、「LCP/NCP < プロコード」は、プロコードよりも厳しい基準でテスト結果を評価する事が重要と考える。

不具合タイプ	アルゴリズム	タイミング・順序	インタフェース	ビルド・パッケージ・結合
差異パターン	LCP/NCP固有	LCP/NCP<プロコード	LCP/NCP<プロコード	LCP/NCP固有
差異理由 考察	プロコードでは原則発生しないが、LCP/NCPでは単体テスト工程が原則無いため、組合テストになって初めてアルゴリズムの不具合が見つかることがある	プロコードでは一定程度発生するが、LCP/NCPではタイミングや順序に関しては製品機能でカバーされる面が多い	- プロコードでは一定程度発生するが、LCP/NCPでは製品機能でカバーされる点もある - ただし、拡張コードで実現するケースもありその場合は不具合が一定数発生	プロコードでは原則発生しないが、LCP/NCPでは構成管理・版管理の手法が確立されておらず、環境設定や拡張コードの反映漏れに伴う不具合が一定数発生

図 10 不具合タイプ比率の差異理由考察

前述の結果を踏まえ LCP/NCP における機能テスト結果の評価指標を策定した。下記をベンチマークとして実案件と比較し、テスト完了判定に活用頂ければ幸いである。

*値の設定と機能性は区別が難しい場合は設定値を合算して評価

タイプ	テスト開始時	テスト終了時
1. 値の設定*	10%程度に設定（プロコードと同様）	- 設定した割合の1.2倍を超える場合テスト結果確認（例：条件分岐の場合24%以上） - 特定機能に不具合が偏る場合は機能強化を検討
2. 条件分岐	20%程度に設定（プロコードと同様）	
3. アルゴリズム	10%程度に設定（LCP/NCP固有）	
4. タイミング・順序	8%程度に設定（プロコード15%の半分程度）	- 設定した割合の1.2倍を超える場合テスト結果確認 - 特定機能に不具合が偏る場合は機能強化を検討 プロコードの割合を超えた場合は全機能の品質強化及び結合テスト工程の延伸を検討
5. インタフェース	22%程度に設定（プロコード35%の2/3程度）	
6. 機能性*	25%程度に設定（プロコードと同様）	- 設定した割合の1.2倍を超える場合テスト結果確認 - 特定機能に不具合が偏る場合は機能強化を検討
7. ビルド・パッケージ・結合	5%程度に設定（LCP/NCP固有）	
8. 関連ドキュメント	設定しない、または5%未満で設定	

図 11 LCP/NCP における機能テスト結果の評価指標

なお、タイプ属性のうち7. ビルド・パッケージ・統合、8 関連ドキュメントを除けば、混入工程は原則設計工程と考える。1. 値の設定、5. インタフェース、6. 機能性は基本設計、2. 条件分岐、3. アルゴリズム、4. タイミング・順序は詳細設計が主な混入工程と考えるが、LCP/NCP では原則製造工程がないため、コードインスペクションによる不具合検知が難しい。よって、設計レビューにて仕様を担保しておくことが重要と考える。また、SQiP2023 にて発表した「ローコード開発 Tips 集」^[1]にて述べたとおり、LCP/NCP 開発はデザインパターンがまだ確立されていないため、良い実装・避けるべき実装をチームで素早く共有すべく、処理構造やアルゴリズムを第三者がレビューする事が有効と考える。レビューすべき作成物についても、SQiP2023 にて発表した「ローコード開発作成物フロー」や「LCP 開発プロセス一覧（一部抜粋）」にて一部述べているため、参考にして頂ければ幸いである。^[1]

なお、上記の評価指標を検証するため、今回調査した 33 件とは別に QCD 達成した LCP/NCP 開発実績：2 件にて事後評価した結果、いずれも指標の範囲内であったことから一定程度実用性はあるが、更なる検証が必要と考える。

該当案件の情報提供者からは、

1. 不具合をモデル化&可視化することで、設計の考慮が不足している機能群が直感的に把握できるのは非常に有効
2. タイプ属性は一部判別が難しい箇所があり、特に LCP/NCP 開発では値の設定と機能性の判別に迷いやすいため、標準モデルを作成する場合は判断基準をルール化するなど工夫が必要
3. LCP/NCP 開発の場合、拡張コードの有無やその量によってモデルが変化する可能性があり、実績データをさらに収集して検証すると、今後新たな気づきが得られそうとのフィードバックを得られた。今後の活動の参考にしていきたい。

5. 活動の振り返り

前述の活動を振り返り、自己評価と得られた気づきを述べる。

5.1 FP ベースの生産性基準確立

標本数が 30 件と限りがあるものの、基本統計量を使った工数の見積りや検証が新規、再構築どちらにおいても可能となった。

5.2 FP ベースのテスト基準確立

標本数が 19 件と限りがあるものの、基本統計量を使ったテストケース数の見積りや検証が新規、再構築どちらにおいても可能となった。

5.3 ODC による機能テスト完了基準確立

今後更なる検証が必要なものの、不具合密度に換わる代替指標として LCP/NCP における機能テスト結果の評価指標を提供する事ができた。なお、AI を活用しプログラムを自動生成する開発案件においても、コード量と不具合数が相関しない場合、ODC 分析による不具合の質に着目した工程前進や出荷判定は有効と考える。引き続き注視していきたい。

最後に、これまで述べた各指標を利用する場合のユースケース（例）を示す。紹介した指標を使えば、明日から皆様のプロジェクトマネジメントを、“勘” から“定量的判断”へと変えることが期待できる。実践的な取組みのため、今後の実務で是非活用頂きたい。

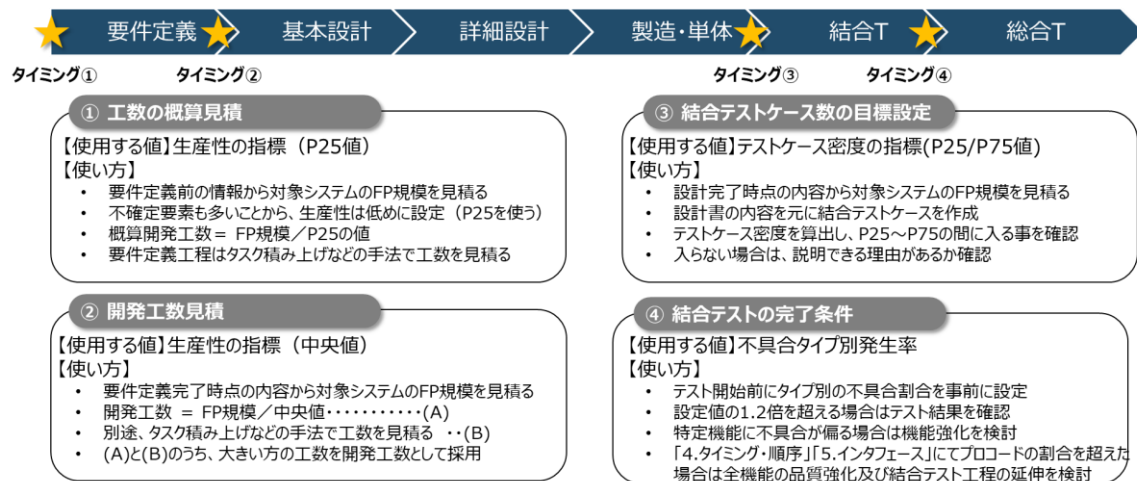


図 12 LCP/NCP における FP 生産性と品質指標のユースケース（例）

6. 今後の展望

lencSig は 2025 年度も活動を継続しており、実績データ収集・分析を継続しつつ、昨今巷でよく話題になる『LPC/NCP は本当にコスト有利なのか』の問いに答えるべく、「プロコード vs LCP/NCP の損益分岐モデル化（導入判断基準の明確化）」の検討など実施している。興味がある方は 6 カ月会費無料の「トライアル会員」となる事で参画可能なためご検討頂ければ幸いです。（詳細は[こちら](#)）

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>
 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>
- [2] 杉崎 眞弘（著）、佐々木 方規（著）、日科技連 ODC 分析研究会（編集）『ソフトウェア不具合改善手法 ODC 分析』