

重い開発プロセスを改善した事例紹介～ECRS の原則を用いた引き算～

1. 重い開発プロセスへの問題意識

年に一度ぐらいは「開発プロセスが重すぎるので、もっと軽くしたい」というご相談を受けます。詳細を伺うと監査やアセスメントで指摘された弱みを改善するうちに、成果物や作業が過剰になっているケースが多いようです。ガイドラインや手順書が増えるにつれ、メンテナンスの時間が増え、更新する際に変更モレが発生することに問題意識を持たれている方が一定数います。過去にはプロセスを軽くしようと何度も検討してきたものの、まだ対策を打てていないという声もありました。

2. 重い開発プロセスになった経緯

プロセスがいつの間にか重くなってしまう背景には、ピンポイントの改善の積み重ねがあります。例えば、ガイドラインや手順書に必要な定義や説明が不足していると、その都度該当箇所に加筆します。Excel の帳票やチケットの項目に情報が足りない場合に、既存の項目に一部重複していることに気付かず付け足されることもあります。適切な追加箇所が見つからず、関係が薄い文書に無理やり記述を追加されたケースもよく見かけます。その結果、メンテナンス性の低下とともに、読み手にとってどこに何が書かれているのか分かりにくくなり、帳票やチケットの記入項目の意図を理解できない状況が生じます。

3. ECRS の原則を用いた開発プロセスの引き算

では、どのように重いプロセスを軽くすればよいでしょうか。全体を現状分析して抜本的な見直しを行う方法も考えられますが、ここでは製造現場で用いられる ECRS の原則^{※1}を活用したアプローチを紹介します。プロセス改善のワークショップでは具体的なタスクや成果物、記載項目を以下の観点（表 1）で確認します。

表 1 ECRS の原則

ECRSの原則	効率化の視点
Eliminate（廃止）	なくせないか？
Combine（結合）	一緒にできないか？
Rearrange（交換）	順序の変更はできないか？
Simplify（簡素化）	単純にできないか？

出所 日本経営工学会編（2002）「生産管理用語辞典」をもとに作成

具体的には改善が必要と考えるプロセスの活動や一連のタスク、帳票、あるいは帳票の中身などを ECRS の視点で確認します。「なくせないか?」、「一緒にできないか?」、「順序の変更はできないか?」、「単純にできないか?」と順番に関係者とディスカッションをしながら改善の可能性を探します。ホワイトボードで活動やタスク、帳票などを書き出しながら確認しますが、プロセスマイニング※2 を活用して可視化した開発フローで確認することも可能です。

4. 過去の改善事例

過去の改善事例（表 2）から、経験的に効果の大きい対策から取り組むことを推奨しています。ただし、効果の大きさと取り組みの労力は比例します。例えば「Eliminate（廃止）」は最も効果が見込める反面、①廃止によるリスクの確認、②廃止について関係者との合意形成、③廃止後にツールを導入する場合は予算の確保など、多くのステークホルダーを巻き込む大規模な改善活動となります。ここで ECRS だけを考えていると落とし穴があります。なぜならば帳票や手順がシンプルになった結果、開発プロセスのリスクが増大したり、できていたことができなくなっては本末転倒です。ポイントとなる活動や成果物あるいはその内容を残しつつ、不要な部分をそぎ落とすことが重要です。

表 2 過去の改善事例

効率化の視点	過去の事例	効果
なくせないか?	・ 構成品目の入手による確認の廃止とツールの導入 ・ エクセル上のID番号によるトレーサビリティの廃止とツールの導入 ・ オフィス文書によるプロセス定義の廃止とツール導入 ・ Excel帳票とチケットへの二重登録の廃止	高い ↑ ↓ 低い
一緒にできないか?	・ 工程終了レビューと品質ゲートレビューの一本化 ・ 技術レビューと成果物監査の同時実施 ・ 成果物およびプロセス監査と重複していた品質ゲートのチェック内容の統合 ・ システムとソフトウェアの設計成果物の管理方法を統一 ・ ガイドラインと手順書間の重複していた内容の統合	
順序の変更はできないか?	・ 実機テストをSILSで検証後に実施 ・ 一括レビューを分割してレビューを実施 ・ レビュー依頼前に設計業務のインプットとレビュー目的を事前説明 ・ プロジェクト計画書レビュー時の利害関係者との合意をキックオフ会議時に変更	
単純にできないか?	・ 要求仕様の構文のテンプレート化 ・ ガイドラインや手順書の説明をポイントを押さえて簡略化 ・ 同じ内容を確認する承認ルートの省略	

5. 開発プロセスの付加価値を残す引き算

今回は ECRS 原則を用いて、各タスクや成果物の必要性和付加価値を見極めながら引き算するアプロー



チをご紹介しました。定義や基準、手順に「足し算」だけでなく、「引き算」の視点を取り入れることで開発プロセスの最適化が図れます。実際には Automotive SPICE や機能安全、サイバーセキュリティで重要視される活動や成果物とその内容を把握していなければ、引き算の判断が難しい場合もあります。弊社では、Automotive SPICE や機能安全、サイバーセキュリティに関わる開発プロセスの軽量化をお客様と弊社メンバーと一緒に検討するハンズオンのワークショップも実施しています。今回の話に関係しそうなところでは BI（Business Intelligence）ツールによる開発プロセスの改善効果の見える化、プロセスマイニングによる開発プロセスの可視化と改善等のワークショップも併せて提供しています。ご興味のある方は関連する弊社ホームページをご覧ください。

※1 ※2 ECRS（イクルス）： Eliminate（廃止）、Combine（結合）、Rearrange（交換）、Simplify（簡素化）の頭文字をとった略称

※2 プロセスマイニング：プロジェクト管理ツールやチケット管理ツールのデータから開発プロセスを可視化して分析することにより業務フローの問題点やボトルネックを見つけること

【リンク】

BI 環境構築事例（過去のメルマガ）：<https://biz3.co.jp/download/column/9201>

プロセスマイニング：<https://biz3.co.jp/trend/proccessmining>

開発プロセスを軽量化（過去のメルマガ）：<https://biz3.co.jp/download/column/3984>

2025/9/4 [西門 克郎](#)