

医薬品開発におけるプログラムバリデーションによる統計解析の品質管理

益田隆史

株式会社ACRONET 開発本部 生物統計部 東京グループ

Quality Control of Statistical Analysis by Program Validation in Drug Development

Takashi Masuda

Tokyo Group, Biostatistics Department, Drug Development Division
ACRONET Corporation

要旨：

プログラムバリデーションによる医薬品開発における統計解析の品質管理を試みた。その性質について現在主流であるダブルプログラミングとの比較を行い、手法としての可能性について考察した。

キーワード：バリデーション、ダブルプログラミング、医薬品開発、プログラムレビュー、テスト

ダブルプログラミングによる統計解析の品質管理

- 2000年SASユーザー会
 - 菅波秀規・益田隆史
 - ダブルプログラミングを系統的に考えてみた
- その後、現在に至るまで
 - ダブルプログラミングが事実上、業界標準になっている
 - 一方、プログラムバリデーションについてはさほど触れられず
 - Computerized System ValidationやCFR21 Part11の話題もあるが、統計解析については業界全体としては未対応

ダブルプログラミングと プログラムバリデーション(狭義)(1)

- ダブルプログラミング
 - 独立2系統で結果を出力し、結果を照合
 - 一致していれば結果を信用する
 - プログラムの正しさは問わない(問えない)
 - 目の前に存在するデータに対してのみ結果の品質を確保
 - 新規のデータについては品質を確保できない
 - 使い捨てのプログラムに向いている
 - プログラムは正しくなくても、結果が正しければ許される(プログラムの誤りに気付く機会が無い)

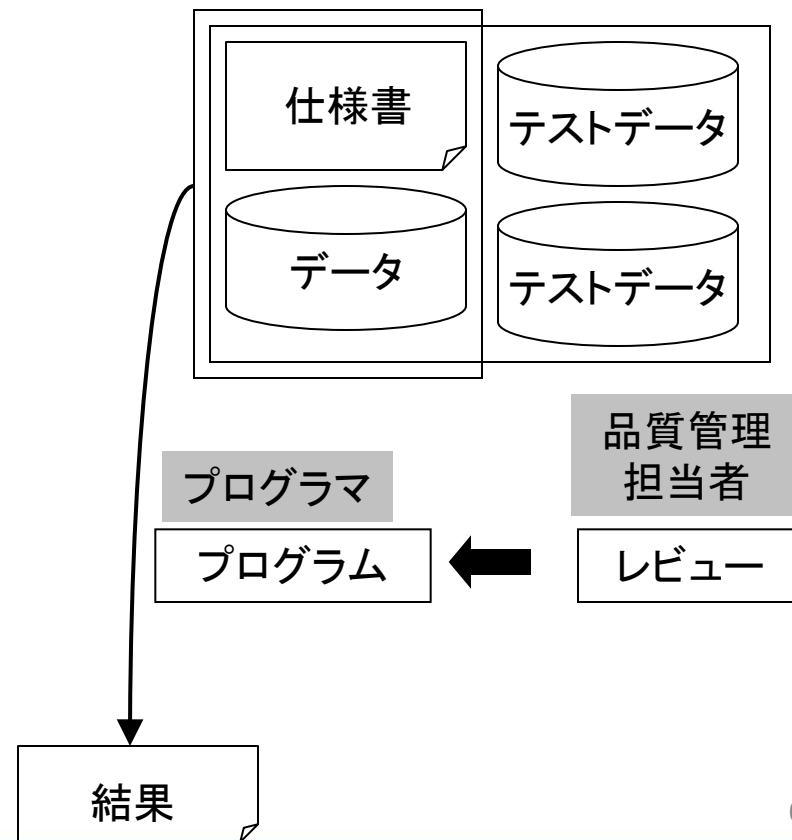
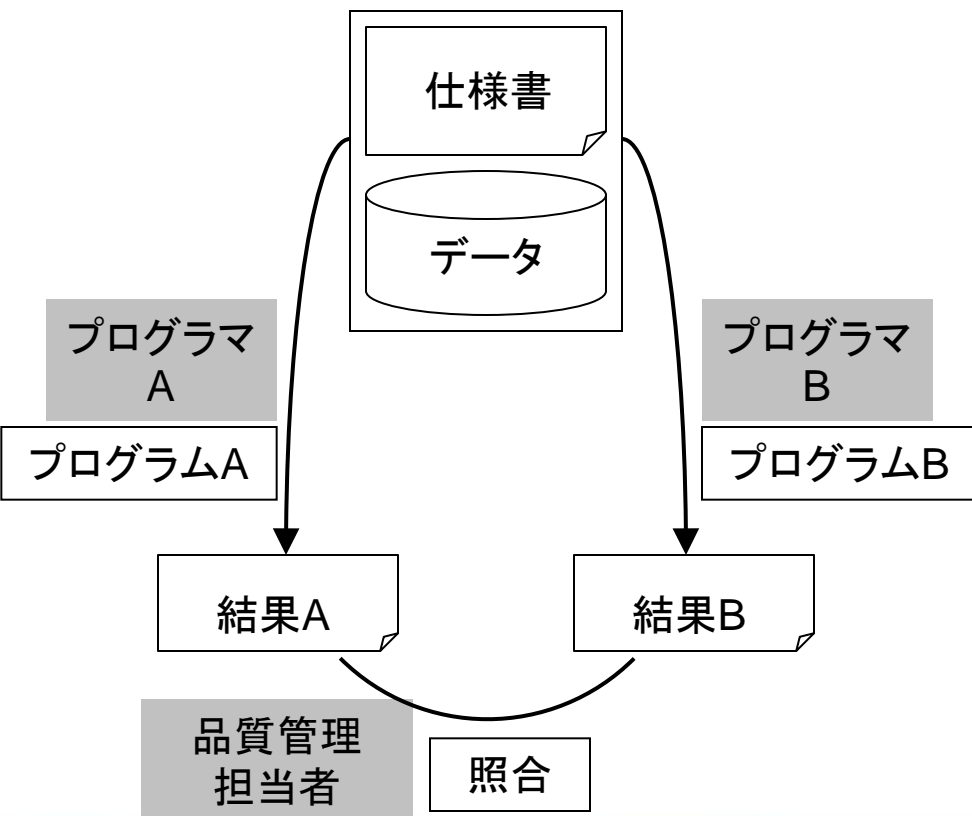
ダブルプログラミングと プログラムバリデーション(狭義)(2)

- ・ プログラムバリデーション(狭義)
 - ・ プログラムの動作を保証する
 - ・ プログラムレビュー
 - ・ テストデータによるテスト
 - ・ 一般に、ソフトウェア業界ではこちらが主流
 - ・ 使い回すプログラムでは必須
 - ・ 想定されるどのようなデータでも正しく動作する必要があるため

ダブルプログラミングと プログラムバリデーション(狭義)(3)

ダブルプログラミング

プログラムバリデーション(狭義)



プログラムバリデーション(狭義)による 品質管理を試みた

- 対象
 - 製造販売後の探索解析
 - 5パターン、計22帳票
 - 解析用データセット作成プログラム・集計プログラム

方法の詳細

- 一系統のプログラマがプログラムを作成
- 別の品質管理担当者がプログラムのバリデーション
 - バリデーション計画の立案
 - バリデーション計画書
 - バリデーションの実施
 - プログラムのレビュー
 - テストデータによるテスト
 - 結果の記録
 - バリデーション報告書
- その後、もう一系統のプログラムも作成し、ダブルプログラミングにより性能評価

バリデーション計画

- 事前の計画が重要と考えた
 - 想定されるデータに対して動作を確認
 - 想定されるデータを念頭に、プログラムのレビューおよびテストデータによるテストを行う
- 計画時にいかに適切な想定を行うか
 - ここで想定した範囲内で正しく動作
 - 実際には、100%完全な想定は出来ない
 - いかに100%に近付けるか
 - 医薬品開発では、この点が困難であり、大きな課題となる
- 事前に計画し、記録しておく事が重要

プログラムレビュー

- プログラムを読む
 - 想定されるデータを正しく処理しているか
 - 今回は1名の品質管理担当者が目視
 - 複数の品質管理担当者(例えばプロジェクトチーム)によるレビュー会の方が好ましい
 - レビュー手法はシステム分野で様々なものが開発されている
 - 他人のプログラムを読むのは労力が大きい
 - ある程度の標準化により改善
 - プログラマが、他人が読む事を前提にプログラムを書けば改善

テストデータによる動作確認

- バリデーション計画で想定したテストデータを作成、動作を確認
 - 結果の確認だけでなく、場合によってはステップごとに動作を追跡
- 単体テスト・結合テスト
 - プログラムがマクロによりモジュール化されている場合
- バリデーション計画の完成度が品質に大きく影響

記録の作成

- バリデーション報告
 - バリデーション計画と対になる
 - バリデーション計画の各項目に対し、結果を記録

検出された問題

- ダブルプログラミングでは検出困難な問題
 - 丸めの問題: round関数を用いずに、format関数で丸め
 - 浮動小数点の誤差により、稀に意図せぬ結果
- ダブルプログラミングでは検出できない場合がある問題
 - 条件判定の誤り(不等号の条件での等号の記述漏れ・不等号の条件で欠測対応漏れ)
- ダブルプログラミングで容易に検出される問題
 - 条件判定の真偽が逆転
 - 仕様の誤解
- その他、今回検討した手法に依存しない問題
 - 古い版の仕様書を参照
 - 出力テンプレート(DDEで流し込むExcelファイル)の誤り

結果と考察

- ダブルプログラミングでは発見されにくい問題が検出された
 - 前述
 - バリデーション後、ダブルプログラミングで追跡を行ったが、新たに発見された問題は無し
- プログラムの頑健性が向上
 - データ変更でも、特に問題は発生せず
 - 使い回しをするプログラムには非常に有効
 - プログラムの使い回しは増える傾向
 - CDISCの普及によるデータの標準化
 - 中間解析の増加によるプログラムの実行回数増加
 - 製造販売後調査の機会の増加、作表数の増加

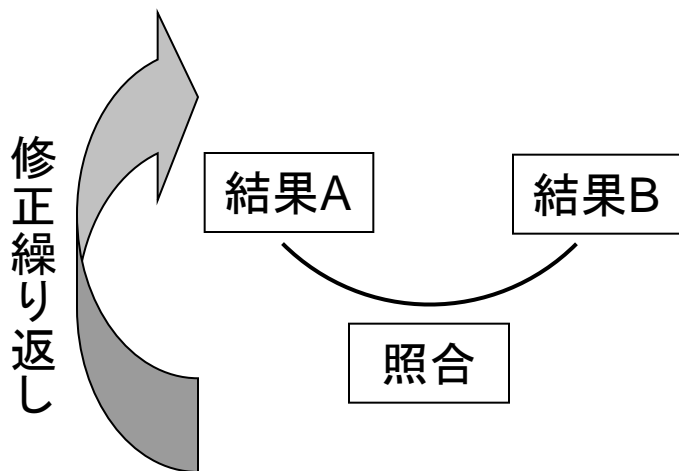
結果と考察

- 想定されるデータの見積もりが重要
 - そもそも、想定出来るのか
 - プログラムバリデーションの最大の障壁
- 今回の方法では、品質管理担当者のスキルが成果物の品質に直結
 - 今回の品質管理担当者のスキル:10年以上の経験、プロジェクトリーダー
 - いかに品質管理手法の標準化を行うかが鍵
 - 品質管理担当者個人ではなく、チームとしての品質管理
 - 例えばプロジェクトチームでのレビュー会により、チェックする「目」の数を増やす
 - 経験豊富な品質管理担当者が検出力が高いが、フレッシュな視点も貴重

結果と考察

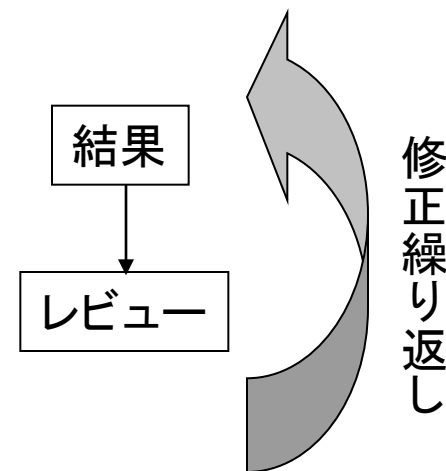
- ・ 余裕を持ったスケジュールが必要
 - ・ プログラムが完成してからプログラムレビュー、問題あればそこから修正

ダブルプログラミング



2系統の修正を平行して行える

プログラムバリデーション(狭義)



修正とレビューは平行して行えない

結果と考察

- ダブルプログラミング以上の人時を要した
 - ダブルプログラミングの片サイドの作業の1.5倍程度
 - 1回使い捨てと考えた場合
 - 使いまわしのプログラムではむしろリソースの削減に繋がる可能性が高い
 - 同一の仕様を使い回す機会が多い、メーカーにコストメリットが見込まれる
- 他人のプログラムを読み、意見を交わす
 - スキルアップに繋がる可能性

結論

- プログラムバリデーション(狭義)による品質管理を行った
- ダブルプログラミングとは大きく異なる特性を持つことが明らかとなった
- 使い回しをするプログラム・データが想定できるプログラムには非常に有効と考えられる
- ダブルプログラミングとの適切な使い分けにより、品質の向上と業務の効率化の可能性がある