

複雑化する資源評価業務との付き合い方

木下順二・真鍋明弘・市野川桃子

水産機構資源研

Nov. 20, 2021

背景と着眼

現状 改正漁業法の下で資源評価対象魚種が拡大する一方、人的リソースは限られる。

変化 このような状況下で、資源評価の透明性・再現性・客観性の要求は高まっており、担当者の負担はますます増えている。

問い この状況に適応するためには働き方（考え方）を変化させていく必要があるのではないか。

そこで、現状の資源評価業務の流れを整理し、ソフトウェア開発手法の適用によって効率化や手順の改善が可能かを検討する。

※ なお、このスライド資料で示した内容の詳細は

Hayashi A, Ichinokawa M, Kinoshita J, Manabe A (2021)
Optimizing stock assessment workflows by applying software
development methodology. Fisheries Research, 244, 106108.

<https://doi.org/10.1016/J.FISHRES.2021.106108>

に掲載されています。

現状の資源評価業務の”例”

資源評価業務は データ整理から始まり、数値計算、ドキュメント作成までの流れ作業 であり、主担当者は全ての工程を多能工的にこなしている。その上で、

1. ドキュメント作成の際に、図表等の検索・選択・取得をマウス操作などの手作業で行う必要がある
 - 検索の手間と選び取りミスの危険がある（この作業は他者からは見えない）。
 - 多量の単純作業—数値や図表のコピーアンドペーストなど—が存在する。
 - 大量の情報・ファイルの管理を行う必要がある。
2. 複数のシナリオを検討する必要がある
 - シナリオ数（試行錯誤回数）が増えるほど単純作業とファイル管理のコストが増す。
3. ドキュメント（ドラフト）を生成して初めて他人からのフィードバックがある
 - スペルミスなどへの細かな指摘から
数値の転記ミスなど注意を要する重要な指摘、さらに、
どのシナリオを選ぶかなどの大きな指摘までがこの段階でなされる。
 - 単純作業に労力を取られて、本質的な問題にたどり着くまでに時間がかかる可能性がある。

状況改善の”案”

作業の最終工程であるドキュメント作成を流れの起点にする という逆転の発想の下で、可能な限りヒューマンエラーを減らす方法をソフトウェア開発手法から学ぶ。

1. ドキュメントに情報へのリンク（パス）を書き込み、図表や数値が自動引用、自動更新される仕組みを使う

□ ツール例: **R, Rmarkdown**



- パスを書き込むことで、どの情報を選び取ったかが明確になる（透明性の確保）
- データの転記ミス等が減る（再現性の向上）
- 図表や数値はその都度計算するため、図表保存などを行う必要が無い（整理性の向上）
- 一度パスを書いてしまえば、次回からはコピペなどの単純作業が不要（省力化）

! R と Rmarkdown のコーディング（プログラミング）技術の習得が必要

状況改善の”案”

2. 多数のシナリオを管理しやすい仕組みを使う

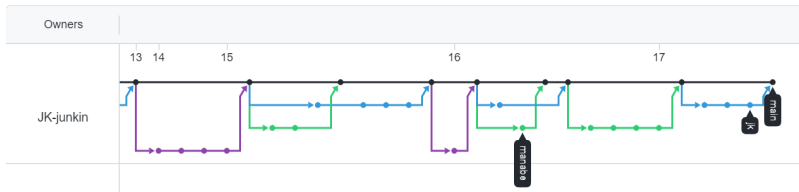
□ ツール例: **Git** (バージョン管理システム)



- 更新履歴の管理を行うシステム
- 更新履歴の管理に止まらず（多人数での）並列開発を念頭に作られたツール
- ファイルの派生を同時に追従し、管理できる（ブランチング）
- ブランチ（派生）は必要に応じて統合できる

！このツールの仕組みの理解および取扱技術の習得が必要

- ブランチングでの共同開発履歴の例（本スライド資料）



状況改善の”案”

3. フィードバックを得やすい仕組みを使う

□ ツール例: **GitHub**、Amazon S3 (クラウドサービス)



- 例えば、数値が異常に大きいなどの経験知をテストプログラムとして組み込み、それが検知されたときはシステムが停止する（問題箇所の発見）
 - システムに人間の知恵を付加するという「自動化」の考え方
 - こうした経験知は暗黙知である場合も多いため、テストプログラムとして明文しておくことで透明性の確保にもつながる
- クラウドサービス上で最新のドキュメントが閲覧できる
 - これはプログラムが全てのテストを無事通過したことを意味するので、「テストが充実していれば」ドキュメントの信頼性が向上する
 - ウェブブラウザを介して閲覧できるので他者がレビューしやすい
 - レビューしているドキュメントのバージョンが、レビュワー間で異なるというすれ違いを低減できる
- クラウドサービス上にコードを格納するため、複数人で同時に開発が可能
 - 他の人の作業完了を待つ必要が無い

！ソフトウェア（パッケージ）として開発する必要

！パッケージを共同開発する文化が必要

！クラウドサービスの取扱技術の習得と仕組みの理解（レビュワーも）

CI ドキュメントシステムの流れ（例）

1. Excel ブックに各種データを整理
→ これをデータベースとする
 2. Excel ブックのデータを R に読込
 3. R および Rmarkdown を用いて
ドキュメントを作成
(数値計算、出力様式の指定)
 4. 作成したドキュメントを **GitHub**
にアップロード
 5. クラウドサービス上で文書が自動
生成される (PDF, Word, HTML)
 6. 生成された文書が関係者に自動的
に公開される
 7. **GitHub** 上でレビューを受ける
 8. ドキュメントファイルを改訂
- 以下、繰り返し。

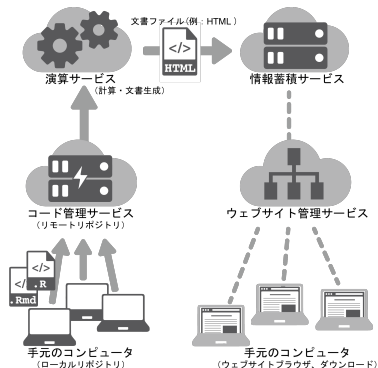


Figure 1: 図. CI ドキュメントシステムの構造 雲形の背景画像はクラウドサービスを表す。Hayashi et al. (2021) の Fig.1 を一部改変して示した。

結果はどうなる

1. ドキュメント生成の自動化

- **再現性の向上** 使用したデータ、解析方法、暗黙知の明文化
- **透明性の確保** ドキュメントのバージョン管理（派生も含めて）
- **信頼性の向上** 転記・更新漏れなどの低減、異常数値等の自動検出
- **省力化** コピペなどの単純作業の低減、年次更新への素早い対応

2. 仕事履歴の見える化（主に GitHub 上で）

- 誰がどのような仕事をしたか
- 誰が何を決めたか
- 議論の内容の文字化

3. 着実な向上

- R パッケージとして開発されていれば、他の担当者の有用なコードを導入できる。
- これにより、組織全体として重複作業の削減につながる。
- さらに、余裕ができることで、より本質的な問題に割く時間を確保しやすくなる。

課題・留意点

- ドキュメントに必要な情報が明確でないと、却って混乱を生み出す。
→ 予め関係者間でドキュメント要件を合意しておくことが重要
- 新しいツール（R, Rmarkdown, Git, GitHub）の学習コスト
→ 自分だけでなく、例えば後任の人にも学習をお願いすることになる。
- 多魚種に共通のパッケージを作るには大規模で相当な労力が必要
→ 各魚種に特有の暗黙知がある（はず）
→ データ整理の仕方も様々
- Microsoft Office 系のソフト（Word や Excel 等）でもマクロやクエリを使えばできるのかもしれない。

結び | 「複雑化する資源評価業務との付き合い方」とは何か

1. 継続的な品質向上を目指した、少人数での大量生産方法の模索

- ドキュメント生成の自動化によってコピペミスや数値の取り扱いミスが減少する
- 自動化によって研究の再現性が向上し、研究の頑健性があがる
- 単純作業の減少と、研究の頑健性向上は次の研究へ進む余裕を生み出す

2. 属人化から標準化へ

- 属人化された知識をメモのみならずコードとして保存
- 引継ぎのたびに失われる知識を保全する
- 各魚種ごとに標準化されたフローや知識を残し、誰でも資源評価を出来るようにする

3. バージョン管理されたコードベースの仕事

- ファイルの管理や整理コストを削減し、ファイル間の連携性が容易になる
- コードベースのやりとりや GitHub 上などでの共同作業により、問題解決性が向上する
- シナリオ分岐が容易になり、より高度な解析への足がかりとなる

参考 | リーン生産方式

- 本スライド資料で示した方法はソフトウェア開発の現場で用いられているリーン生産方式（ムダの少ない生産方式）を参考にしています。

そのコンセプトは、

1. プロジェクトの **価値** を明確にする
2. **価値の流れ** を明確にする
3. 工程間の作業速度を可能な限り均すことで **流れを止めない**
4. **後工程から前工程に** タスクを取りに行く
5. 流転するユーザーの要求を完全に満たすよう改善を **続ける**

の 5 項目から成ります（Hayashi et al. 2021）。

- リーン生産方式の元となったのは、トヨタ自動車が生産現場で昭和 20 年代初頭から実践されているトヨタ生産方式に由来するようです。
トヨタ生産方式は、製品が自動車なので究極の安全品質を確保しながらも、原価低減による利益増大を達成するためには徹底したムダの排除しかない、という信念の下、30 年近い年月をかけて形になった（今なお改善され続けている）生産方式です。
- ソフトウェア開発や自動車生産など、資源評価業務の現場とはかけ離れた存在に感じますが、**品質を維持・向上しながらユーザーの要求の変化に対応し続ける** という特徴は資源評価業務にも共通する部分があると思います。

補足 | 資源評価とリーン生産方式

資源評価業務でそれを実現するためには、（当たり前のことかもしれませんが）

1. 作業手順を改善し続ける

- ムダの認識と撲滅
- ムダの分類（例）
 - 1.1 コピーなどの単純作業の増加
 - 1.2（単純作業の）未完の作業
 - 1.3 遅れや待ち
 - 1.4 欠陥
 - 1.5 タスクの切り替えにかかる時間
 - 1.6 余分な機能
 - 1.7 引継ぎにかかる労力（引き継がれない暗黙知）
- ムダ（問題点）を発見するために5回の「なぜ」を常に自らに問う

2. 手順改善では対応できない部分を設備改善（使う道具）で補う

- 新しい道具に習熟しようとする気概を持つ

3. 働き方を変えるには長い時間と多くの労力が必要であることを認める

- 個人的な技術向上から組織を上げての変革まで

などの継続的な行動と、強い意思が必要だと考えています。

補足 | 本発表資料の作成について

- 本発表の資料（スライド）は **Rmarkdown, Git, GitHub** を用いて複数人で同時作成しました。発表資料作成に使用したコードは GitHub の公開レポジトリに格納されています。

レポジトリ URL : https://github.com/JK-junkin/gakkai_FO202111

The screenshot shows the GitHub interface for the repository `JK-junkin / gakkai_FO202111`. The repository is public and has 3 watchers, 1 star, and 0 forks. The navigation bar includes links for Code, Issues (6), Pull requests (2), Actions, Projects, Wiki, and Security. The main content area shows a list of files and a pull request. The files list includes `figures`, `.gitignore`, `README.md`, `example.Rmd`, `outline.md`, `slide.Rmd`, and `style.css`, each with a corresponding commit message and time. The pull request section shows a merge pull request #31 from JK-junkin/manabe, created 16 hours ago. The right sidebar contains links for About, README, Releases, and Packages.

JK-junkin / gakkai_FO202111 Public

Unwatch 3 Unstar 1 Fork 0

<> Code Issues 6 Pull requests 2 Actions Projects Wiki Security ...

main Go to file Add file Code About

JK-junkin Merge pull request #31 from JK-junkin/manabe 16 hours ago 39

figures	Add network graph #26	yesterday
.gitignore	Add ignore list file	4 days ago
README.md	Add README.md	23 days ago
example.Rmd	Add example.Rmd	20 hours ago
outline.md	Add what is new stock assessment wor...	3 days ago
slide.Rmd	Change according with suggestion	yesterday
style.css	:lipsticks: Add css file and small change	2 days ago

No description, website, or topics provided.

Readme

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

参考文献・サイト等

- Hayashi A, Ichinokawa M, Kinoshita J, Manabe A (2021) Optimizing stock assessment workflows by applying software development methodology. Fisheries Research, 244, 106108.
<https://doi.org/10.1016/J.FISHRES.2021.106108>
- 大野耐一 (1978) 『トヨタ生産方式—脱規模の経営を目指して—』ダイヤモンド社. pp.232.
- 高橋康介 (2018) 『再現可能性のすゝめ—RStudio によるデータ解析とレポート作成—』 Wonderful R vol.3, 石田基広監修. 共立出版. pp.161.
- Jennifer Davis, Ryn Daniels 著, 吉羽龍太郎 監訳, 長尾高弘 訳 (2018) 『Effective DevOps—4 本柱による持続可能な組織文化の育て方—』オライリー・ジャパン. pp.341.
- 市谷聡啓, 新井剛 (2018) 『カイゼン・ジャーニー—たった 1 人からはじめて、「越境」するチームをつくるまで—』翔泳社. pp.279.