

GitHub を活用した 教科書作成 方法論 の提案

Cline

2025 年 2 月 5 日

1 第 1 章：はじめに

2 序論：オープン教科書作成の新たな地平

教育の現場では、知識の創造と共有のあり方が常に進化を続けています。デジタル技術の発展、特にオープンソースプラットフォームの普及は、教材作成、公開、そして改善のプロセスに革命的な変化をもたらす可能性を秘めています。本論文では、GitHub を中核とした、**真にオープン**で**協調的**な教科書作成モデルを提案し、教育機関における教材開発の新たな地平を拓くことを目指します。

従来の教科書作成は、専門家主導によるトップダウン型のアプローチが主流でした。完成された教材は、改訂が困難で、学習者のニーズや変化する知識体系への迅速な対応が難しいという課題がありました。また、教材作成のプロセスが不透明で、**誰が**、**どのように** 教材を作成し、**どのように** 改善しているのかが不明瞭であるという問題も指摘されてきました。

これに対し、本論文で提案する GitHub を活用した教科書作成モデルは、以下の 原則（げんそく）に基づいています。

- **オープン性**: 教材のソースコードを公開し、**誰でも** 閲覧、fork、修正提案が可能な状態にします。
- **協調性**: GitHub のプルリクエストや Issue 機能を活用し、**誰でも** 教材の改善に**参加**できる仕組みを構築します。
- **透明性**: 教材の変更履歴を公開し、**誰が**、**いつ**、**どのように** 教材を修正したのかを明確にします。
- **持続可能性**: コミュニティによる継続的な改善サイクルを組み込み、教材の鮮度と質を長期的に維持します。
- **アクセシビリティ**: GitHub Pages を活用し、**誰でも**、**どこからでも** 教材にアクセスできる環境を提供します。

本論文では、これらの `принципы` を具現化する具体的な方法として、GitHub、GitHub Pages、GitHub Issues の活用法を詳細に解説します。また、教育機関における導入事例や、モデルの有効性と課題についても議論します。

本研究が、教育関係者、教材開発者、そして学習者自身にとって、よりオープンで協調的な教材作成のあり方を考えるきっかけとなり、教育の質の向上に貢献できれば幸いです。

次章では、GitHub を教材作成基盤として活用する具体的な方法について解説します。

3 第2章：GitHub を用いた 教科書 作成

4 第2章：GitHub を基盤とした教材作成

本章では、オープン教科書作成の中核となる、GitHub を基盤とした教材作成について解説します。GitHub は、単なるバージョン管理ツールではなく、**オープン**で**協調的**な教材作成を実現するための強力なプラットフォームです。

4.1 GitHub を教材作成に活用する 原則 (げんそく)

GitHub を教材作成に活用する 原則 (げんそく) として、以下の点が挙げられます。

- **オープン性**: 教材リポジトリを **公開** し、**誰でも** 閲覧可能にします。
- **可視性**: 教材の作成プロセス、変更履歴を **公開** し、透明性を確保します。
- **協調性**: Issue や Pull Request を活用し、**誰でも** 教材改善に**貢献**できる **オープン** な **参加モデル** を構築します。
- **継続的改善**: コミュニティからのフィードバックを **迅速** に 反映し、教材を継続的に **改善** する サイクルを確立します。

4.2 GitHub の主要機能と教材作成

GitHub は、教材作成に役立つ **多様な** 機能を備えています。主要な機能と、教材作成における活用例を以下に示します。

- **リポジトリ**: 教材の **ソースコード**、画像、その他の関連ファイルを **保存** します。- 例: 教材 **リポジトリ** を作成し、章ごとに Markdown ファイルを **分割** して **保存** します。
- **バージョン管理 (Git)**: 教材の **変更履歴** を **追跡** し、過去のバージョンへのロールバックを可能にします。- 例: 章の修正、図の差し替え、演習問題の追加など、教材への変更を commit 単位で **記録** します。
- **Issue**: 教材の **課題**、改善要望、質問などを **記録** し、議論するための **フォーラム** を提供します。- 例: 学習者 から 教材 の 誤植 報告 や 内容 に関する 質問 を Issue で 受け付けます。
- **Pull Request**: 教材への変更提案を行い、レビューを依頼する **仕組み** を提供します。- 例: 学習者が教材の修正案を Pull Request として提案し、**著者** がレビューしてマージします。
- **GitHub Pages**: リポジトリ内のファイルを **ウェブサイト** として **公開** する **仕組み** を提供します。- 例: リポジトリの Markdown ファイルを HTML に **変換** し、**教科書ウェブサイト** を **自動化** します。

4.3 教材作成のワークフロー

GitHub を活用した教材作成の **一般的な** ワークフローを以下に示します。

1. **リポジトリ作成:** 教材 リポジトリ を作成 します (公開リポジトリ を 推奨)。
2. **ソースコード作成:** Markdown や $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ などで 教材 の ソースコード を 作成 します。
3. **コミットとプッシュ:** 作成した ソースコード を リポジトリ に commit し、push します。
4. **GitHub Pages 設定:** GitHub Pages を 有効化 し、教材 **ウェブサイト** を **公開** します。
5. **Issue/Pull Request ベースの改善:** 学習者 や 他の教育関係者 から Issue や Pull Request を 通じて フィードバック や 改善提案 を 収集 し、教材 に 反映 します。

4.4 Markdown と $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ の選択

教材 の 内容 や 目的 に 応じて、Markdown と $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ を **適切に** 選択 することが 重要 です。

- **Markdown:** 軽量 で 記述が容易 であり、**ウェブコンテンツ** に 適しています。- **ウェブ教科書**、**講義資料**、**軽量の技術ドキュメント** などに 適しています。
- **$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$:** 数式 や 複雑なフォーマット に 優れており、**プロフェッショナルな印刷物** を 作成 できます。- **印刷教科書**、**学術論文**、**フォーマットにこだわりたい教材** などに 適しています。

4.5 まとめ

本章では、GitHub を **基盤** とした 教材 作成 について 解説 しました。GitHub の **多様な 機能** を 活用 することで、**オープン**、**可視的**、**協調的**、**継続的改善** が **可能** な **教材作成モデル** を 実現 できます。

次章では、GitHub Pages を 用いた 教材 の 公開 方法 について、詳細 に 解説 します。

5 第3章：GitHub Pages を用いた教科書公開

前章では、GitHub を基盤とした教材作成について解説しました。本章では、GitHub の **提供する 静的ウェブサイトホスティング** サービスである GitHub Pages を活用し、作成した教材をウェブ上に公開する方法について解説します。

5.1 GitHub Pages とは：教材公開の最適な選択肢

GitHub Pages は、GitHub リポジトリ 内の ファイル (HTML, CSS, JavaScript, 画像 など) を静的ウェブサイトとして公開できるサービスです。**教科書** や **講義資料** など、静的 **コンテンツ** で構成される教材の公開に最適です。

5.2 GitHub Pages の利点：教材公開を容易に

GitHub Pages を教材公開に利用するメリットは、主に以下の点が挙げられます。

- **簡単公開:** リポジトリ に教材ファイルを push するだけで、ウェブサイトを公開できます。
- **無料ホスティング:** GitHub Pages は無償で利用できるため、費用を抑えて **ウェブサイト** を運用できます。
- **カスタムドメイン:** カスタムドメインを設定することで、ウェブサイトのアドレスを独自のものにでき

ます。

- **HTTPS 対応:** HTTPS に標準で対応しており、ウェブサイトのセキュリティを確保できます。
- **バージョン管理との連携:** GitHub リポジトリ と連携しているため、教材の更新に合わせて **ウェブサイト** も自動的に更新されます。

5.3 GitHub Pages の タイプ：プロジェクトページとユーザーページ

GitHub Pages には、主に以下の 2 つのタイプがあります。

- **プロジェクトページ:** リポジトリ ごとに公開されるウェブサイトです。- 主に、ソフトウェアプロジェクトのドキュメンテーションやデモサイトなどに利用されます。- ウェブサイトの URL は、`https://<ユーザ名>.github.io/<リポジトリ名>` となります。
- **ユーザーページ:** ユーザーアカウントごとに公開されるウェブサイトです。- 主に、個人のポートフォリオやブログなどに利用されます。- ウェブサイトの URL は、`https://<ユーザ名>.github.io` となります。- ユーザーページは、**リポジトリ 名が <ユーザ名>.github.io である リポジトリ の main ブランチから公開されます。**

教材公開には、プロジェクトページ、ユーザーページのどちらでも利用できますが、**教科書** など複数の教材を公開する場合は、ユーザーページを利用すると便利です。

5.4 GitHub Pages 公開手順：プロジェクトページの場合

ここでは、プロジェクトページを例に、GitHub Pages で教材を公開する手順を説明します。

1. **リポジトリ作成:** 教材を **保存** する **リポジトリ** を作成します。
2. **ファイル作成:** リポジトリ に HTML ファイル や Markdown ファイルなど、ウェブサイトとして公開したいファイルを作成します。- Markdown ファイルは、GitHub Pages によって自動的に HTML に変換されます。
3. **GitHub Pages 有効化:** リポジトリ の Settings ページから、GitHub Pages を有効化します。- Source ブランチを選択し (**通常 main ブランチ**)、Save ボタンをクリックします。
4. **ウェブサイト公開:** 数分後、`https://<ユーザ名>.github.io/<リポジトリ名>` に **ウェブサイト** が公開されます。

5.5 mkdocs を用いた 教科書 作成：より効率的な 教科書 作成

教科書 のような **多ページ** な **ウェブサイト** を作成する場合、mkdocs のような静的サイト ジェネレーターを利用すると便利です。mkdocs は、Markdown 形式で **教科書** の **コンテンツ** を記述し、**美しいウェブサイト** を **容易に** 生成できるツールです。

mkdocs を利用した **教科書** 作成の手順は、以下のようになります。

1. **mkdocs インストール:** `pip install mkdocs` コマンドで mkdocs をインストールします。
2. **mkdocs プロジェクト作成:** `mkdocs create <プロジェクト名>` コマンドで mkdocs プロジェクトを

作成します。

3. **設定ファイル修正**: `mkdocs.yml` ファイルを修正し、**教科書** のタイトルやテーマなどを設定します。
4. **コンテンツ記述**: `docs` ディレクトリ内に Markdown ファイルで **教科書** の **コンテンツ** を記述します。
5. **ウェブサイト生成**: `mkdocs build` コマンドで **ウェブサイト** を生成します。
6. **リポジトリにデプロイ**: 生成された **ウェブサイト** (`site` ディレクトリ内) を **リポジトリ** に push します。
7. **GitHub Pages 有効化**: 前述の手順で GitHub Pages を有効化します。

`mkdocs` を利用することで、**教科書** のような構造化された **ウェブサイト** を **効率的** に作成し、GitHub Pages で公開することができます。

5.6 まとめ：GitHub Pages で教科書を世界に公開

本章では、GitHub Pages を用いた教材公開の方法について解説しました。GitHub Pages を活用することで、教材を **容易** にウェブ上に公開し、**アクセシビリティ** を向上させることができます。

次章では、GitHub Issues を用いた教材の改善方法について解説します。

6 第4章：GitHub Issues を用いた教科書改善

教科書 は **教育者** や **学習者** からのフィードバックを継続的に取り入れ、改善し続けることが重要です。GitHub Issues は、**リポジトリ** に関連するタスクやバグ報告、機能要望などを管理するためのツールであり、教材の改善プロセスに活用できます。本章では、GitHub Issues を活用した教材改善について、以下の点について解説します。

6.1 GitHub Issues とは：教材改善に最適なツール

GitHub Issues は、GitHub **リポジトリ** 内で発生した課題や議論を管理するための機能です。教材の改善に関するフィードバックや課題を **リポジトリ** で管理することで、**教育者** や **学習者** との連携をスムーズにし、教材の **質** 向上に繋げることができます。

6.2 GitHub Issues のメリット：教材改善を効率的に

GitHub Issues を教材改善に利用するメリットは、主に以下の点が挙げられます。

- **フィードバックの一元管理**: 教材に関するフィードバックを Issues に集約することで、**教育者** や **学習者** からの意見を **効率的** に管理できます。
- **課題の可視化**: 改善課題を Issues として登録することで、**現在** 取り組むべき課題や今後対応すべき課題を可視化できます。
- **進捗管理**: 各 **Issue** の進捗状況をラベルやマイルストーンで管理することで、改善プロセスの進捗を把握しやすくなります。
- **コミュニケーションの促進**: Issues 上で **教育者** や **学習者** とコミュニケーションを取りながら、課題解決に取り組むことができます。

- **履歴の記録:** Issues での議論や決定事項は履歴として記録されるため、後から経緯を振り返ることができます。

6.3 GitHub Issues を用いた改善プロセス：教科書改善のサイクル

GitHub Issues を用いた教材改善のプロセスは、主に以下のステップで構成されます。

1. **フィードバック収集:** 教育者 や 学習者 から教材に関するフィードバックを収集します。- GitHub Issues, アンケート, 質問箱, 口頭でのフィードバックなど、 **多様な 方法で フィードバック** を収集します。
2. **Issue 起票:** 収集したフィードバックをもとに、改善課題を Issue として起票します。- 1 つの課題に対して 1 つの Issue を起票するのが基本です。- Issue タイトルは課題の内容を端的に表現し、詳細記述欄に課題の背景や具体的な内容を記述します。- 必要に応じてラベル (例: **バグ**, **改善要望**) や担当者、マイルストーンを設定します。
3. **Issue 対応:** 起票された Issue に対して、 **教育者** や開発チームが対応を行います。- Issue a b t o p や関係者と Issues 上で議論しながら、課題の解決策を検討します。- 必要に応じて **リポジトリ** のコードを修正し、変更内容を Issue にコメントとして記録します。
4. **Issue クローズ:** 課題が解決したら、Issue をクローズします。- 解決 **概要** や対応 **概要** などを Issue にコメントとして残しておく、後から振り返る際に役立ちます。

この **サイクル** を繰り返すことで、 **教科書** は **恒常的** に改善され、 **教育者** や **学習者** にとってより価値の高い教材となっていくでしょう。

6.4 Issue 起票の ポイント：効果的な Issue 管理のために

Issue を起票する際には、以下の点に注意すると、Issue が管理しやすくなります。

- **1 Issue 1 課題:** 1 つの Issue に対して 1 つの課題を対応させます。複数の課題をまとめて 1 つの Issue にすると、課題が複雑になり、管理が難しくなります。
- **具体的に記述:** Issue タイトルや詳細記述欄には、課題の内容を具体的に記述します。抽象的な記述では、課題の内容が理解しにくく、対応に時間がかかります。
- **ラベル活用:** Issue ラベルを適切に活用することで、Issue の分類や優先順位付けがしやすくなります。例えば、 **バグ**, **改善要望**, **緊急**, **高優先度** などのラベルを用意しておく便利です。
- **担当者設定:** Issue に担当者を設定することで、責任の所在が明確になり、対応漏れを防ぐことができます。
- **マイルストーン設定:** 中長期的な改善計画がある場合は、マイルストーンを設定することで、Issue の整理や進捗管理がしやすくなります。

6.5 Issue テンプレートの活用：容易な Issue 起票

GitHub には、Issue テンプレートを作成する機能があります。Issue テンプレートを活用することで、Issue 起票時に必要な項目を定型化し、Issue の 質 を均質化することができます。例えば、**バグ報告**、**機能要望**、**質問** などの Issue テンプレートを用意しておくと、**教育者** や **学習者** が **容易** に Issue を起票できるようになります。

6.6 まとめ：GitHub Issues で 教科書 を 向上 し続ける

本章では、GitHub Issues を用いた教材改善の方法について解説しました。GitHub Issues を活用することで、**教育者** や **学習者** からのフィードバックを **容易** に収集し、**教科書** の改善に繋げることができます。

次章では、ここまでの内容を踏まえ、**教科書** 作成の事例を紹介します。

7 第 5 章：結論

本稿では、GitHub を教材作成に活用する方法について、以下の 3 つの側面から考察しました。

- **GitHub リポジトリを用いた教材のソースコード管理**: - ソースコードをリポジトリで管理することで、バージョン管理や共同編集が容易になり、教材作成の効率と品質を向上させることができます。
- **GitHub Pages を用いた教材のウェブ公開**: - GitHub Pages を活用することで、教材を容易にウェブで公開し、誰でもアクセスできるようにすることができます。
- **GitHub Issues を用いた教材の改善**: - GitHub Issues を活用することで、読者からのフィードバックを容易に収集し、教材の継続的な改善を実現することができます。

これらの考察を踏まえ、GitHub は教材作成手法として非常に有効であると結論付けられます。GitHub を活用することで、教材著者はソースコード管理、ウェブ公開、フィードバック収集、改善といった教材作成に必要な全てのプロセスを一元的に行うことができ、教材作成の効率と品質を大幅に向上させることが期待できます。

また、GitHub はオープンソース教材の作成にも非常に適しています。ソースコードを公開し、誰でも貢献できるようにすることで、コミュニティ参加を促進し、教材の品質向上と充実を図ることができます。オープンソース教材は、継続的に改善し続ける教材の典型例と言えるでしょう。

もちろん、GitHub は万能のツールではありません。教材の内容や目的によっては、他のツールや手法の方が適切な場合もあるでしょう。しかし、本稿で述べたようなメリットを考慮すれば、GitHub は教材作成手法の有力な選択肢の 1 つとも言えるでしょう。

今後は、GitHub を教材作成手法として普及させるための活動が重要になってくると考えられます。教材著者向けのセミナーやワークショップを開催したり、GitHub 教材作成手法のガイドラインを作成したりすることで、より多くの教材著者が GitHub を活用し、品質の高い教材が生み出されることが期待されます。

謝辞

本稿を執筆するにあたり、貴重な助言を賜りました専門家の方々に深く感謝申し上げます。

付録

本稿で紹介した教材作成手法を実践するための導入手順について、次章で詳しく解説します。

8 第6章：導入手順

9 第7章：導入手順

本章では、教材をGitHubで作成・公開・改善するための具体的な導入手順を解説します。本稿で紹介した手法を教材作成に導入する際の参考にしてください。

9.1 リポジトリ 作成

まず、教材のソースコードを管理するためのリポジトリを作成します。GitHubで新しいリポジトリを作成し、リポジトリ名を決定します。リポジトリ名は、教材のタイトルや内容を反映した、分かりやすい名前にしましょう。パブリックリポジトリとして作成し、ソースコードを公開に公開します。

9.2 ソースコード 作成

リポジトリを作成したら、教材のソースコードを作成します。ソースコードは、MarkdownやLaTeXなどのマークアップ言語で記述し、テキストエディタやIDEで作成します。教材の構成を検討し、章や節に分割してソースコードを作成しましょう。図や表、コード例などを教材に含める場合は、適切なファイルもリポジトリに追加します。

9.3 GitHub Pages 設定

教材をウェブで公開するために、GitHub Pagesを設定します。リポジトリのSettingsページで、GitHub Pagesを有効にし、sourceをmainブランチまたはgh-pagesブランチに設定します。必要に応じてカスタムドメインを設定し、ウェブサイトのURLを決定します。GitHub Pagesの設定が完了すると、ウェブサイトが公開され、教材にアクセスできるようになります。

9.4 教材 公開

GitHub Pagesでウェブサイトが公開されたら、教材を公開します。ウェブサイトのURLを周囲に告知し、教材の公開をアナウンスしましょう。SNSやブログなどで教材を紹介したり、教材リポジトリのREADMEにウェブサイトへのリンクを掲載したりすることも効果的です。

9.5 フィードバック 収集 と 改善

教材を公開後は、読者からのフィードバックを収集し、教材の改善を行います。GitHub Issuesを活用し、フィードバックを一元的に管理し、課題の可視化や優先順位付けを行いましょう。フィードバックをもとにソースコードを修正し、リポジトリにcommitします。修正内容は自動的にウェ

ブサイト に 反映 され、教材 が 更新 されます。この サイクル を 繰り返す こと で、教材 を 継続的 に 改善 させていきましょう。

9.6 まとめ：GitHub 教材作成 手法 を 実践 しよう

本章では、GitHub 教材作成 手法 の 導入 手順 を 解説 しました。今回 解説 した 手順 を 参考に、ぜひ GitHub 教材作成 手法 を 実践 してみてください。GitHub を 活用 すること で、教材作成 が より 効率的 に、より 高品質 になる はず です。そして、より 多く の 人々 が教材 を 容易 に 作成 し、公開 し、改善 できるようになる こと を 願っています。