職業訓練教材の作成・改訂における LaTeX 及び Git の活用事例

松永 龍真　福原 卓巳　岩島 早苗

島根職業能力開発促進センター

2026-11-28

職業訓練教材の作成・改訂における
LaTeX 及び Git の活用事例

松永 龍真　福原 卓巳　岩島 早苗

島根職業能力開発促進センター

2026-11-28

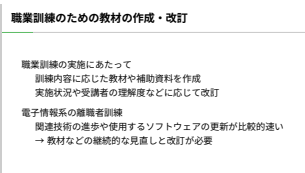
職業訓練教材の作成・改訂における LaTeX 及び Git の活用事例と題して、島根職業能力開発促進センターの松永龍真が発表します。

背景・目的

職業訓練のための教材の作成・改訂

職業訓練の実施にあたって
- 訓練内容に応じた教材や補助資料を作成
- 実施状況や受講者の理解度などに応じて改訂

電子情報系の離職者訓練
- 関連技術の進歩や使用するソフトウェアの更新が比較的速い
- → 教材などの継続的な見直しと改訂が必要

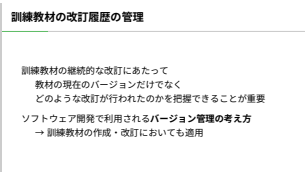


職業訓練を実施するにあたって、訓練内容に応じたテキスト教材や補助資料を作成し、訓練の実施状況に応じてそれらを改訂する場面は少なくありません。特に、電子情報系の離職者訓練においては、関連する技術の進歩や訓練に使用するソフトウェアなどの更新が比較的速く、教材などの継続的な見直しや改訂が必要となります。

訓練教材の改訂履歴の管理

訓練教材の継続的な改訂にあたって
　教材の現在のバージョンだけでなく
　どのような改訂が行われたのかを把握できることが重要

ソフトウェア開発で利用される**バージョン管理の考え方**
　→ 訓練教材の作成・改訂においても適用

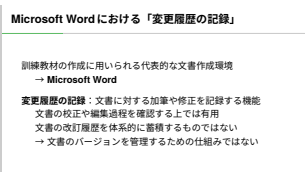


教材を継続的に改訂する場合、現在のバージョンだけでなく、過去にどのような改訂が行われたのかを把握できることが重要です。そのため、ソフトウェア開発などで利用されるバージョン管理の考え方を訓練教材にも適用することは有用であると考えられます。

Microsoft Word における「変更履歴の記録」

訓練教材の作成に用いられる代表的な文書作成環境
→ **Microsoft Word**

変更履歴の記録：文書に対する加筆や修正を記録する機能
文書の校正や編集過程を確認する上では有用
文書の改訂履歴を体系的に蓄積するものではない
→ 文書のバージョンを管理するための仕組みではない



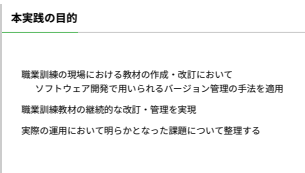
訓練教材の作成に用いられる環境として、Microsoft Word があります。これには、変更履歴の記録を行うための機能が備わっており、文書に対する加筆や修正を記録できます。この機能は、文書の校正や編集過程を確認する上では有用である一方、文書の改訂履歴を体系的に蓄積するものではありません。したがって、文書のバージョンを管理するための仕組みとしてこの機能を使うことは難しいと考えられます。

本実践の目的

職業訓練の現場における教材の作成・改訂において
　ソフトウェア開発で用いられるバージョン管理の手法を適用

職業訓練教材の継続的な改訂・管理を実現

実際の運用において明らかとなった課題について整理する



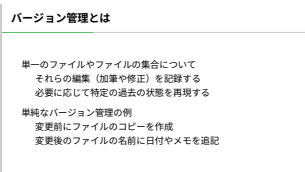
これらを踏まえた、今回の実践の目的です。職業訓練の現場における、教材の作成・改訂において、ソフトウェア開発で用いられるバージョン管理の手法を適用します。これにより、訓練教材の継続的な改訂・管理を実現します。実際の訓練現場における約半年間の運用において明らかになった課題について整理し、今後の展望について述べます。

Gitを用いたバージョン管理

バージョン管理とは

単一のファイルやファイルの集合について
- それらの編集（加筆や修正）を記録する
- 必要に応じて特定の過去の状態を再現する

単純なバージョン管理の例
- 変更前にファイルのコピーを作成
- 変更後のファイルの名前に日付やメモを追記



バージョン管理とは、単一のファイルや複数のファイルの集合に対する変更を記録し、必要に応じて過去の状態を再現できるようにすることを指します。これを実現する最も単純な例として、変更前のファイルのコピーを作成し、ファイル名などに変更日時や変更に関する覚書を添えておく方法が考えられます。

ファイル名によるバージョン管理の例

- テキスト_R08-09-08.docx
- テキスト_R08-09-08_誤字修正.docx
- メインテキスト.docx
- メインテキスト_25-02-21.docx
- メインテキスト_20250520.docx
- メインテキスト - コピー.docx
- 最新版_メインテキスト.docx

ファイル名によるバージョン管理の例
テキスト_R08-09-08.docx
テキスト_R08-09-08_誤字修正.docx
メインテキスト.docx
メインテキスト_25-02-21.docx
メインテキスト_20250520.docx
メインテキスト - コピー.docx
最新版_メインテキスト.docx

ファイル名によるバージョン管理の例を示します。この方法は単純である一方、運用する上ではいくつかの問題があります。まず、コピーされたファイルが増加するにしたがって、これらのファイルの管理が煩雑になっていきます。加えて、どのファイルが最新であるのかを判別しづらくなり、それぞれの内容の差や更新の前後関係を把握しづらくなります。

Git：ソフトウェア開発分野のバージョン管理システム

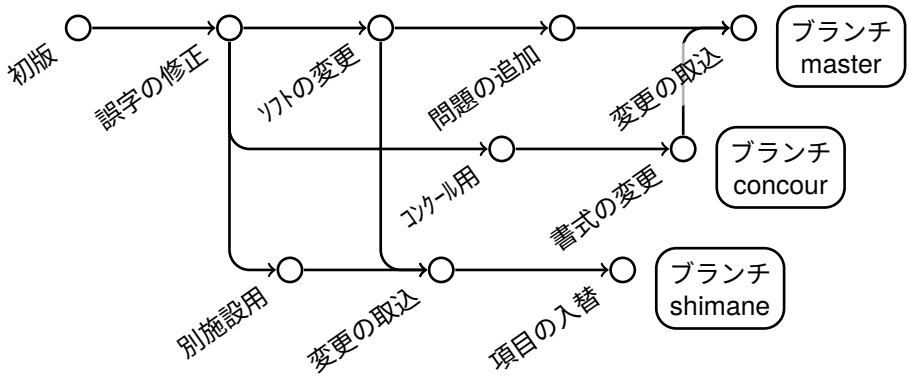
バージョン管理システム（version control system: VCS）
　変更履歴を体系的に記録・管理するシステム

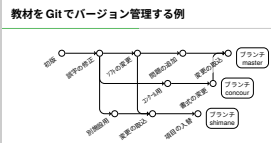
Git：ソフトウェア開発の分野を中心に広く利用される VCS
　ひとまとまりの変更を**コミット**として記録
　変更者や日時、内容などを記述したメッセージを添える
過去の状態の確認や再現、バージョン間の比較を実現可能

Git：ソフトウェア開発分野のバージョン管理システム
バージョン管理システム（version control system: VCS）
　変更履歴を体系的に記録・管理するシステム
Git：ソフトウェア開発の分野を中心に広く利用される VCS
　ひとまとまりの変更を**コミット**として記録
　変更者や日時、内容などを記述したメッセージを添える
過去の状態の確認や再現、バージョン間の比較を実現可能

バージョン管理システム（VCS）は、このような変更履歴を体系的に記録・管理するためのシステムです。ソフトウェア開発の分野を中心に広く利用される VCS のひとつに、Git があります。Git では、ひとまとまりの変更点をコミットと呼ばれる単位で管理します。コミットには、変更点のほかに、誰が変更を加えたのか、いつ変更を加えたのか、その内容や背景は何であるのかを表すメッセージを添えられます。これにより、過去の状態の確認や特定のバージョンの再現、バージョン間の差分の比較をより容易に実現できるようになります。

教材を Git でバージョン管理する例





スライドの図には、教材開発におけるバージョンの履歴の例を示しています。Git において、ファイルの変更履歴は、コミットの連なりとして表現されます。また、コミットから複数のバージョンの流れを分岐したり、統合したりできます。これをブランチやマージといいます。教材のバージョン管理に Git を導入し、適切な管理を行うことで、教材を単なるファイルとして扱うのではなく、その改訂履歴及び派生関係を含めて管理できるようになります。

LaTeX を用いた教材作成

Git と相性の良いファイル形式

Git はテキストファイルの変更内容を差分表示することに長ける
　変更内容の比較を容易に行える形式で教材を管理
　→ Git の利点を活かせる

Microsoft Word のファイル形式（ISO/IEC 29500）はバイナリ形式
　そのままでは Git で管理するには不向き

Git と相性の良い教材の記述形式を検討

Git と相性の良いファイル形式

Git はテキストファイルの変更内容を差分表示することに長ける
　変更内容の比較を容易に行える形式で教材を管理
　→ Git の利点を活かせる

Microsoft Word のファイル形式（ISO/IEC 29500）はバイナリ形式
　そのままでは Git で管理するには不向き

Git と相性の良い教材の記述形式を検討

Git は、テキストファイルにおける変更内容を差分として扱うことに長けます。Git によるバージョン管理の利点を最大限活かすには、Git によって容易に変更内容を比較できる形式での教材管理が望まれます。ここで、Microsoft Word のファイル形式は、バイナリ形式であるため、そのまま Git で管理するには不向きであると考えられます。これらを踏まえ、Git と相性の良い教材の記述形式を検討しました。

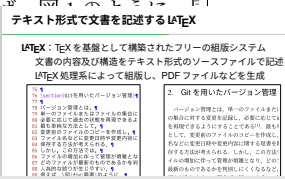
テキスト形式で文書を記述する LaTeX

LaTeX：TeX を基盤として構築されたフリーの組版システム
文書の内容及び構造をテキスト形式のソースファイルで記述
LaTeX 処理系によって組版し、PDF ファイルなどを生成

```
75 ¶
76 \section{Gitを用いたバージョン管理}¶
77 ¶
78 バージョン管理とは，¶
79 単一のファイルまたはファイルの集合に
80 必要に応じて過去の状態を再現できるよ
81 最も単純な方法として，¶
82 変更前のファイルのコピーを作成し，¶
83 ファイル名などに変更日時や変更内容に
84 保存する方法が考えられる．¶
85 しかし，この方法では，¶
86 ファイルの増加に伴って管理が煩雑とな
87 どのファイルが最新のものであるかを判
88 人為的な誤りが生じやすい．¶
89 例えば，\図{fig:最悪}のように，¶
```

2. Git を用いたバージョン管理

バージョン管理とは，単一のファイルまたは
の集合に対する変更を記録し，必要に応じて過
を再現できるようにすることである[1]．最も単
として，変更前のファイルのコピーを作成し，
名などに変更日時や変更内容に関する覚書を附
存する方法が考えられる．しかし，この方法で
イルの増加に伴って管理が煩雑となり，どのフ
最新のものであるかを判別しにくくなるなど，



Git との親和性の高い方式のひとつとして、組版システムである LaTeX を用いるものがあります。LaTeX 文書では、スライド左側のように文書の内容及び構造をテキスト形式のソースファイルとして記述し、これを LaTeX 処理系によって組版することでスライド右側のような PDF ファイル形式などの文書を生成します。教材の作成者は教材を直接編集するのではなく、LaTeX ソースを編集し、それをコンパイルすることで成果物である教材を得ます。

文書の構造化で体裁を統一できる

LaTeX では文書の内容と体裁とを分離して記述できる

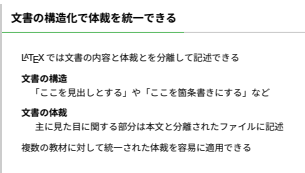
文書の構造

「ここを見出しとする」や「ここを箇条書きにする」など

文書の体裁

主に見た目に関する部分は本文と分離されたファイルに記述

複数の教材に対して統一された体裁を容易に適用できる



LaTeX で教材を作成することの利点は、Git との相性に留まりません。LaTeX では、文書の内容と体裁とを分離して記述できます。例えば、LaTeX ソースでは、文書の内容に加えて「ここを見出しとする」や「ここを箇条書きにする」といった文書の構造を記述します。文字の大きさやページの余白、見出しの書式などは、文書クラスやマクロとして定義され、本文を記述するファイルから分離されます。これにより、共通の文書クラスなどを用いることで、複数の教材に対して統一された体裁を容易に適用できるようになります。

運用の実践

運用の実践

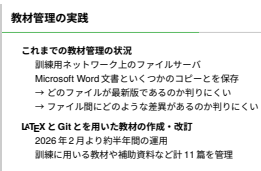
教材管理の実践

これまでの教材管理の状況

訓練用ネットワーク上のファイルサーバ
Microsoft Word 文書といくつかのコピーとを保存
→ どのファイルが最新版であるのか判りにくい
→ ファイル間にどのような差異があるのか判りにくい

LaTeX と Git とを用いた教材の作成・改訂

2026 年 2 月より約半年間の運用
訓練に用いる教材や補助資料など計 11 篇を管理



これまでの教材管理は、訓練用ネットワーク上のファイル共有サーバに Microsoft Word 文書とそのコピーとがいくつか保存されているだけでした。そのため、どのファイルが最新版であるのか判りにくく、さらに、ファイル間にどのような差異があるのか判りにくい状態でした。これを受け、LaTeX 及び Git を用いて教材の作成・改訂を行うための環境を構築しました。2026 年 2 月から約半年間、このシステムを運用し、訓練に用いる教材や補助資料、提示用のプログラムを含む計 11 篇の管理を行いました。

環境構築

Microsoft Windows 上で動作する環境を構築

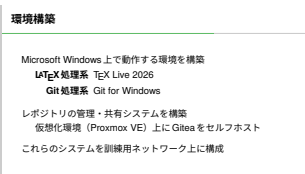
LaTeX 処理系 TeX Live 2026

Git 処理系 Git for Windows

レポジトリの管理・共有システムを構築

仮想化環境（Proxmox VE）上に Gitea をセルフホスト

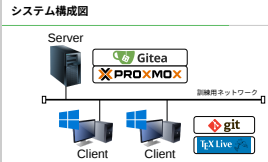
これらのシステムを訓練用ネットワーク上に構成



構築した環境についてお話します。Microsoft Windows 上において動作する LaTeX 処理系として TeX Live 2026 を用い、Git に関連する操作を行うために Git for Windows を用いました。加えて、Git のレポジトリを管理・共有するためのシステムとして Gitea のセルフホスト運用を行いました。これは、仮想化プラットフォーム環境である Proxmox VE 上に構築され、訓練用ネットワークに接続されました。

システム構成図





システム構成図をスライドに示しています。セルフホスト可能なシステムを用いることで、訓練用ネットワークのような閉じたネットワーク上でも複数人の作業者による教材の作成・改訂を可能とする環境を実現しました。

LaTeX の導入による利点と反省点

一般的な日本語文書の体裁を提供する jlreq クラスを使用
　教材の作成者によらず統一感のある体裁を実現
　文書の構造の整理をしやすくなった
　目次や図番号の管理負荷を低減できた

LaTeX 記法のコマンドや LaTeX 処理系の操作方法の習得が必要
　WYSIWYG な文書作成環境に慣れた指導員にとって障壁

LaTeX の導入による利点と反省点
一般的な日本語文書の体裁を提供する jlreq クラスを使用
教材の作成者によらず統一感のある体裁を実現
文書の構造の整理をしやすくなった
目次や図番号の管理負荷を低減できた
LaTeX 記法のコマンドや LaTeX 処理系の操作方法の習得が必要
WYSIWYG な文書作成環境に慣れた指導員にとって障壁

LaTeX の導入による利点と反省点とを整理します。LaTeX を用いて文書を作成するにあたって、一般的な日本語文書の体裁を提供する文書クラスを共通して使用しました。これにより、教材の作成者によらず統一感のある体裁の文書を安定して作成できるようになりました。また、文書の内容と体裁とを分離して記述することにより、教材の構造の整理をしやすくなるという副次的な利点もありました。加えて、目次や図番号の管理負荷を低減できたという意見もありました。反面、LaTeX 記法特有のコマンドや処理系の操作方法の習得が必要であることは、見たままを得られる WYSIWYG な環境に慣れた指導員にとって障壁となることがわかりました。

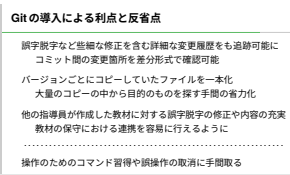
Gitの導入による利点と反省点

誤字脱字など些細な修正を含む詳細な変更履歴をも追跡可能に
　コミット間の変更箇所を差分形式で確認可能

バージョンごとにコピーしていたファイルを一本化
　大量のコピーの中から目的のものを探す手間の省力化

他の指導員が作成した教材に対する誤字脱字の修正や内容の充実
　教材の保守における連携を容易に行えるように

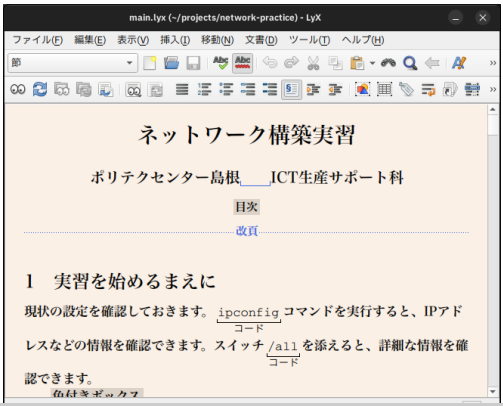
操作のためのコマンド習得や誤操作の取消に手間取る

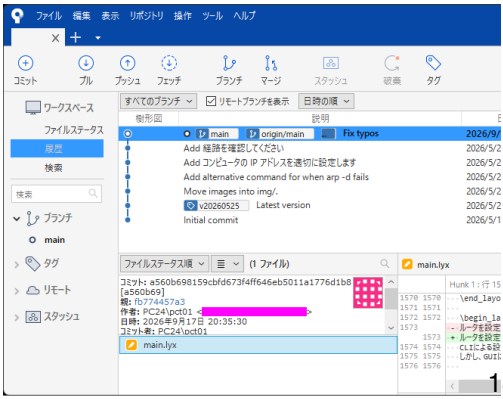


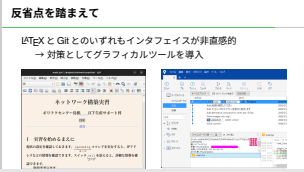
続いて、Git の導入による利点を反省点とを整理します。Git のバージョン管理機能により、些細な変更履歴をも追跡可能になり、また、変更内容の把握が容易になりました。バージョン毎にコピーしていたファイルを一本化でき、大量のコピーの中から目的のものを探すための手間を省けました。加えて、Gitea の機能を活用することにより、他の指導員が作成した教材に対する誤字脱字の修正や内容の充実など、教材の保守を容易に行えるようになりました。反面、Git の操作のためのコマンドの習得が必要であり、さらにコマンド操作で誤操作を取り消す際に手間取る点が課題として挙げられました。

反省点を踏まえて

LaTeX と Git とのいずれもインタフェイスが非直感的
→ 対策としてグラフィカルツールを導入







LaTeX を用いた教材の作成や Git を用いたバージョン管理に関する問題は、いずれもインタフェイスが非直感的であることに由来すると考えられます。これらの問題を解消するために、グラフィカルツールの導入を試みました。教材作成においては、LaTeX 文書を直接記述する代わりに LyX を用いました。これは、LaTeX をバックエンドに用いる文書プロセッサです。LyX のファイル構造は TeX 文書に似たテキスト形式であるため、Git との親和性を損わないまま教材開発の負担感を軽減できます。また、Git の操作をグラフィカルに行える Sourcetree の併用を試みました。これらを導入することで、教材の作成・改訂に係るシステムの使用感を向上できました。

まとめ

職業訓練教材の作成・改訂における LaTeX 及び Git の活用

職業訓練教材の作成・改訂に LaTeX 及び Git を導入する実践
- 教材をバージョン管理することは有用
- 非直感的なインタフェイスを克服するための工夫が必要

Git によるバージョン管理を科や系、施設をまたいで運用
- 教材を共有するための仕組みとしても活用可能
- → 改訂履歴を含め、組織全体で活用できる資産へ

職業訓練教材の作成・改訂における LaTeX 及び Git の活用
職業訓練教材の作成・改訂に LaTeX 及び Git を導入する実践
教材をバージョン管理することは有用
非直感的なインタフェイスを克服するための工夫が必要
Git によるバージョン管理を科や系、施設をまたいで運用
教材を共有するための仕組みとしても活用可能
→ 改訂履歴を含め、組織全体で活用できる資産へ

職業訓練教材の作成・改訂に LaTeX 及び Git を導入する実践を行いました。この実践により、教材を体系的にバージョン管理することは有用であることの実感が得られました。ただし、これを広く適用するには、非直感的なインタフェイスを克服するための工夫が必要であると考えられます。今回の実践は、一施設の中の単一の科によって行われたに過ぎません。今後の展望として、このようなバージョン管理を科や系、施設をまたいで運用できるようになると、教材を共有するための仕組みとしても活用可能であると考えられ、ひいては、改訂履歴を含めて組織全体で活用できる資産として運用できるのではないかと考えます。

おわり

このスライドについて

Written in November 2026.

Permanent ID of this document: `04c51c83a244dd56`.

Copyright © 2026 松永龍真, 福原卓巳, and 岩島早苗.

特記なき場合、プログラムやソースコード、その他のコンテンツは BSD 2-Clause License の下で利用可能です。なお、各画像の著作権は、それぞれの画像の著作者に帰属します。

このスライドについて

Written in November 2026.

Permanent ID of this document: 04c51c83a244dd56.

Copyright © 2026 松永龍真, 福原卓巳, and 岩島早苗.

特記なき場合、プログラムやソースコード、その他のコンテンツは BSD 2-Clause License の下で利用可能です。なお、各画像の著作権は、それぞれの画像の著作者に帰属します。