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）

1. はじめに

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，訓練の実施状況や受講者の理解度などに応じてそれらを改訂する場面が少なくない．電子情報系の離職者訓練においては，関連技術の進展や訓練で使用するソフトウェアなどの更新が比較的速いことから，教材の継続的な見直しと更新が求められる．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となる．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る．

訓練教材の作成に用いられる代表的な文書作成環境として，Microsoft Word がある．これは，変更履歴の記録を行う機能を備えており，文書に対する加筆や削除などを記録できる．この機能は，文書の校正や編集過程を確認する上で有用である一方，文書の改訂履歴を体系的に蓄積し，文書のバージョンを管理する仕組みではない．

本稿では，職業訓練の現場における教材の作成・改訂において，ソフトウェア開発の分野で広く用いられるバージョン管理の手法を適用した事例について報告する．実際の運用において明らかとなった課題について整理し，将来の展望について述べる．

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である[1]．最も単純な方法として，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内容に関する覚書を附記して保存する方法が考えられる．しかし，この方法では，ファイルの増加に伴って管理が煩雑となり，どのファイルが最新のものであるかを判別しにくくなるなど，人為的な誤りが生じやすい．例えば，図 1 のように，同一の教材について複数のファイルが存在すると，それぞれの内容や更新の前後関係を把握することが困難となる．

バージョン管理システム（version control system: 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ある．ソフトウェア開発の分野を中心に広く利用されている VCS のひとつとして，Git が挙げられる．Git では，ひとまとまりの変更を記録したものをコミットと呼ぶ．各コミットには，その時点におけるファイルの状態に加えて，変更者，変更日時，変更内容を記述したコミットメッセージなどが記録される．これにより，過去の状態の確認や，ある時点の教材の再現，バージョン間の差分の確認ができるようになる．Git は，分散型の VCS であるため，各作業者がそれぞれ独立したレポジトリ[1)]を持って作業できる．したがって，作業者は，サーバへの接続なしに変更履歴を記録できる．また，必要に応じてサーバへ接続し，レポジトリ間で変更を共有・統合できる．この性質により，複数の指導員がそれぞれの環境で教材を編集する場合にも適用できる．加えて，Git では，コミットから複数のバージョンの流れを分岐（ブランチ）させられる．例えば，標準的な教材を基準として，特定の施設や訓練内容に特化した教材を別のブランチ上で作成できる．さらに，ブランチ間の変更を比較し，必要な変更を別のブランチへ統合（マージ）することもできる．図 2 に教材の派生と変更の統合を想定したブランチの例を示す．Git を用いることで，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る．

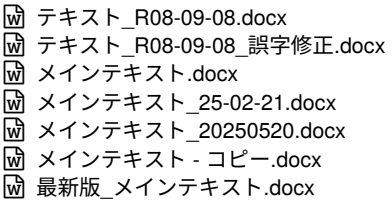


図 1　ファイル名による脆弱なバージョン管理の例

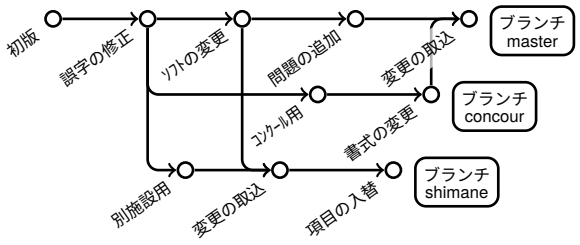


図 2　ブランチによるバージョンの派生の例

3. LaTeX を用いた教材作成

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る．Git によるバージョン管理の利点を活かすには，Git によって容易に変更内容を比較できる形式で教材を管理することが望ましい．Git との親和性の高い文書の記述形式のひとつとして，LaTeX があ

1) レポジトリとは，バージョン管理の対象となるファイル群や変更履歴などのまとまりのことをいう．

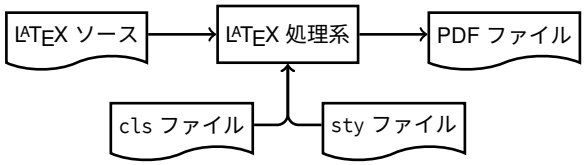


図 3　LaTeX 文書の処理の流れ

る．LaTeX は，TeX を基盤として構築されたフリーの組版システムである[2]．図 3 に LaTeX 文書の処理の流れを示す．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，PDF や DVI，PostScript などの文書を生成する．すなわち，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る．

LaTeX では，文書の内容とその体裁とを分離して記述できる．例えば，教材本文の 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する．文字の大きさや余白，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，LaTeX ソースから分離された `cls` ファイルや `sty` ファイル内に記述される．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適用できるようになる．

4.　環境構築及び運用の実践

LaTeX 及び Git を用いて教材を作成・改訂するための環境を構築した．Microsoft Windows において動作する LaTeX 処理系として TeX Live 2026[3]を用い，Git に関連する操作を行うために Git for Windows[4]を用いた．加えて，Git レポジトリを管理・共有するためのシステムとして，Gitea[5]のセルフホスト運用を行った．Gitea 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た．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などの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た．

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あり，どのファイルが最新版か判りにくく，また，各ファイル間にどのような差があるのかも判りにくい状態であった．2026 年 2 月より，上記の環境を用いて，訓練に用いる教材や補助資料 7 篇，提示用のプログラム 2 篇，受講者へ配布するための文書 2 篇などの作成・改訂を行った．

LaTeX を用いて教材を作成するにあたって，一般的な日本語文書用の体裁を用いることで，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可能となった．反面，教材を作成するために LaTeX 記法特有のコマンドや LaTeX 処理系の操作方法の習得が必要であることは，WYSIWYG（what you see is what you get）な文書作成環境に慣れた指導員にとって，障壁となることがわかった．

Git を用いたバージョン管理を行うことで，些細な修正を含む詳細な変更履歴を追跡可能になった．バージョンごと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が省けた．教材を LaTeX で作成しているため，各コミットでどのような変更があったのかを差分形式で容易に確認可能となった．また，誤りから容易に復旧できるため，抜本的な変更を行うことに対する抵抗感が低減した．加えて，Gitea の機能を活用することで，他の指導員が作成した教材を基に，誤字脱字の修正や内容の充実などといった，教材の保守を容易に行えるようになった．しかし，操作のためのコマンドや手順の習得が負担となり，誤操作の取消しに手間取る点が課題として挙げられた．

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，いずれもシステムを扱うための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る．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た．教材作成においては，LaTeX 文書を直接記述するかわりに，LaTeX をバックエンドに用いる文書プロセッサである LyX[6]を用いた．LyX 文書の構造は，TeX 文書に似たテキスト形式であるため，Git との親和性を損なわない．また，バージョン管理においては，Git の操作をグラフィカルに行える Sourcetree[7]を併用した．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が向上した．

5.　おわりに

本稿では，訓練教材の作成・改訂における LaTeX 及び Git の活用について検討した．本実践は，単一の施設における単一の科のみによって行われた．この取り組みが科や系，施設間においても運用されるようになれば，教材を共有するための仕組みとしても活用でき，ひいては，組織全体で活用できる資産の一つになると考える．

なお，本稿は，LaTeX を用いた文書作成の一環として，[8] で配布しているクラスファイルを用いて作成された．

参考文献

[1] Scott Chacon, Ben Straub, and contributors: “Pro Git”, Version 2.1.450, online <https://github.com/progit/progit2> (2026).
[2] 奥村晴彦，黒木裕介：「[改訂第 9 版] LaTeX 美文書作成入門」，株式会社技術評論社，東京（2023）．
[3] “TeX Live”, TeX Users Group, <https://www.tug.org/texlive/>, 2026/09 参照.
[4] “Git for Windows”, Git for Windows, <https://gitforwindows.org/>, 2026/09 参照.
[5] CommitGo, “Gitea Official Website”, <https://about.gitea.com/>, 2026/09 参照.
[6] “LyX – The Document Processor”, <https://www.lyx.org/>, 2026/09 参照.
[7] Atlassian, “Sourcetree”, <https://www.sourcetreeapp.com/>, 2026/09 参照.
[8] KusaReMKN, “KusaReMKN/uitex”, GitHub, <https://github.com/KusaReMKN/uitex>, 2026/09 参照.