歴史から知るバージョン管理システム

この資料について

- 目的
  - バージョン管理システム（VSC）について分かった気にさせる
  - どうして世の中にVSCが必要なのかを理解する

Gitを学ぶ近道はバージョン管理システムの歴史を知ること！

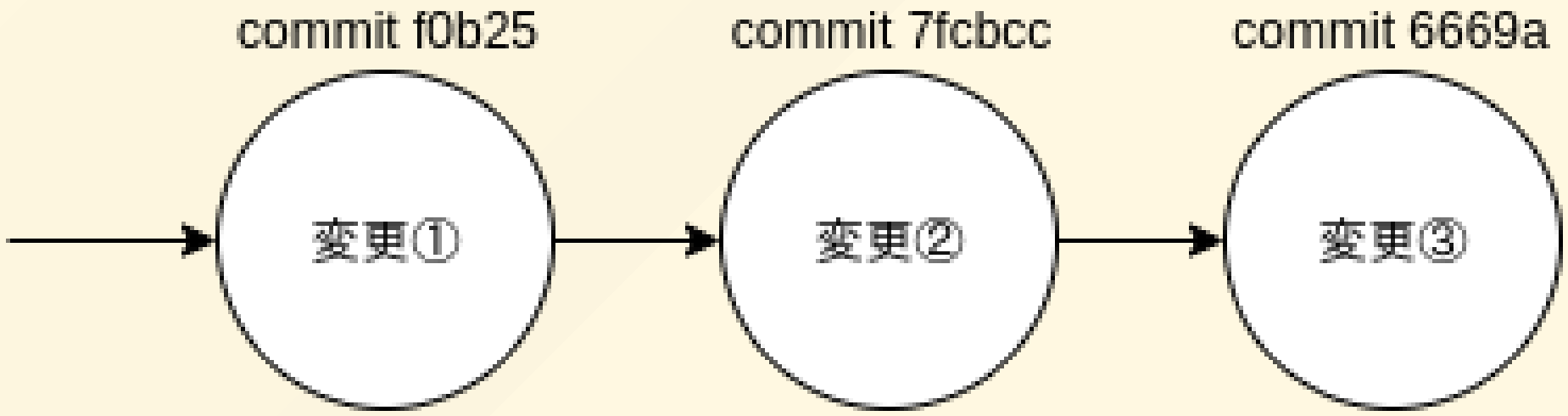
（VSCはVersion Control Systemの略）

バージョン管理システムとは

バージョン管理システム

ファイルの作成や削除、変更を管理するためのシステム。

ファイルの作成日付、変更日、変更点などをシステムで保管して、
いつでも過去の状態に戻したり、過去の変更を見たりできる。



バージョン管理システムがない世界

バージョン管理システムを使わない環境でのファイル管理方法一例

- 仕様書
- 仕様書_最新
- 仕様書_旧
- 仕様書_新

→ 管理するファイルが増え、とにかくやりづらい
→ ファイルごとの変更点が分からない
→ 手動マージなど、**地獄の作業**が待っている

バージョン管理システムがある世界

バージョン管理システムの歴史

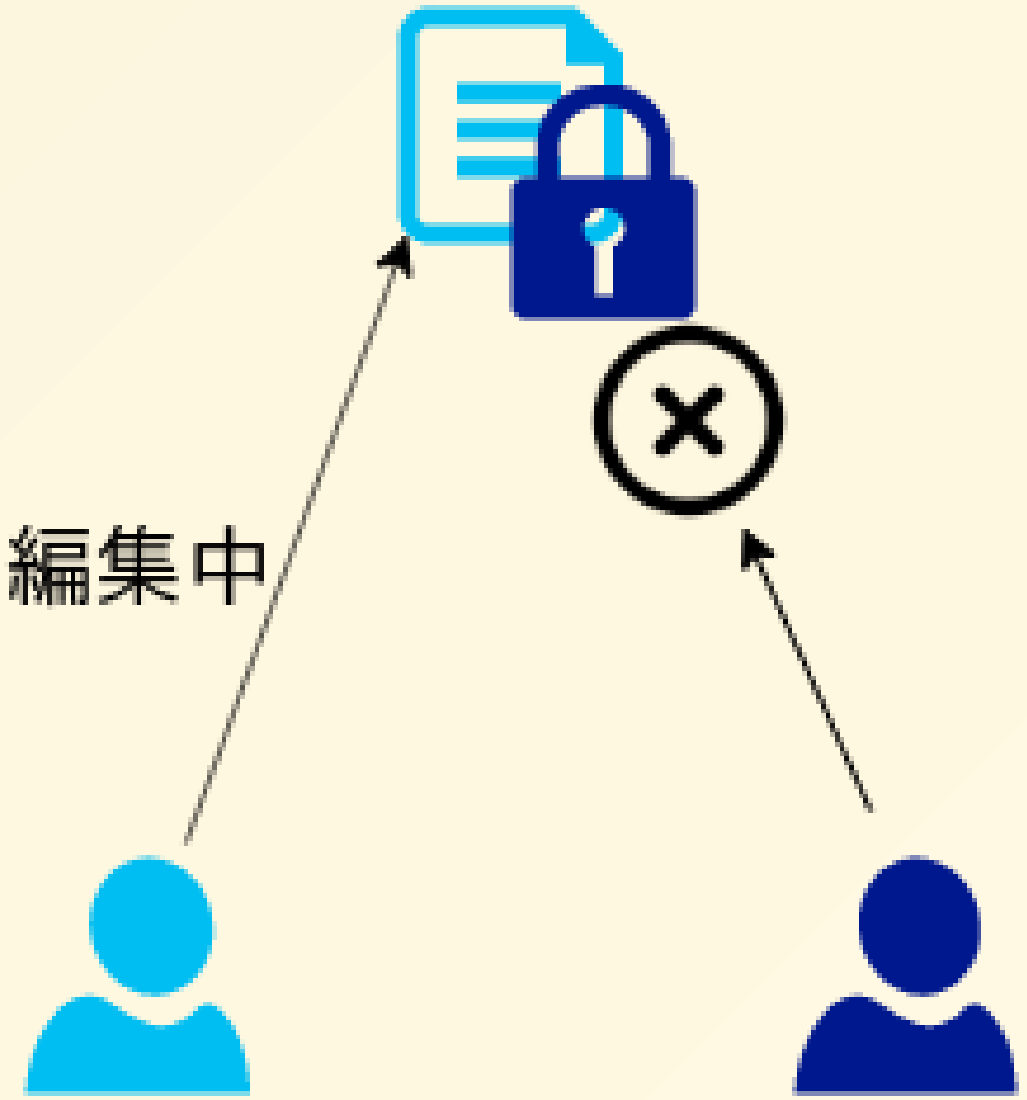
VCSは新しい概念やマシンパワーの成長に合わせて進化してきた。

ここでは代表的なVCSとその機能を紹介する。

- 1982年：RCS（ロックアンドモディファイ）
- 1990年：CVS（エディットアンドマージ）
- 2000年：svn（アトミック・コミット）
- 2005年：git（分散バージョン管理）

1982年：RCS

- 初期のVCS
- ロックアンドモディファイ
- ロックして変更という形で
  ファイルの不整合を防ぐ
- 同時編集者は1人まで



1982年：RCS②

この方式では編集するファイルにロックをかけ、他のユーザーが編集できないようにしてから内容の変更・コミット（反映とロック解除）を行った。

- **コミット**
  - 編集を確定し、システム上のファイルに反映すること
  - RCSでは同時に、ロックの解除も行う

1990年:CVS

- エディットアンドマージ
- 編集するファイルをシステムからローカルにコピーし、このコピーを編集する
- 編集が完了したらシステム上のファイルと合体させる。
- 元ファイルとの差分のみを取り込むマージ機能によって、複数人のユーザが同じファイルを編集できるようになった。

1990年：CVS②

元ファイルとの差分のみを取り込むマージ機能によって、複数人のユーザが同じファイルを編集できるようになった。

- **コミット**
  - 変更差分をシステム上のファイルに反映すること

2000年：svn

- アトミックコミット
- Subversion
- 今でも現役！

2000年：svn②

今までは1ファイルごとにコミットしなければいけなかったが、
この機能によって複数のファイルを1度にコミットすることが可能になった。

スライドの最後でGitとの比較やります

2005年：git

- 分散バージョン管理システム
- 各々がクローンしたもの
  1つ1つがリポジトリとなった

2005年：git②

SubversionとGitの違い