Using Git for Lesson Plans for Unit Level Training

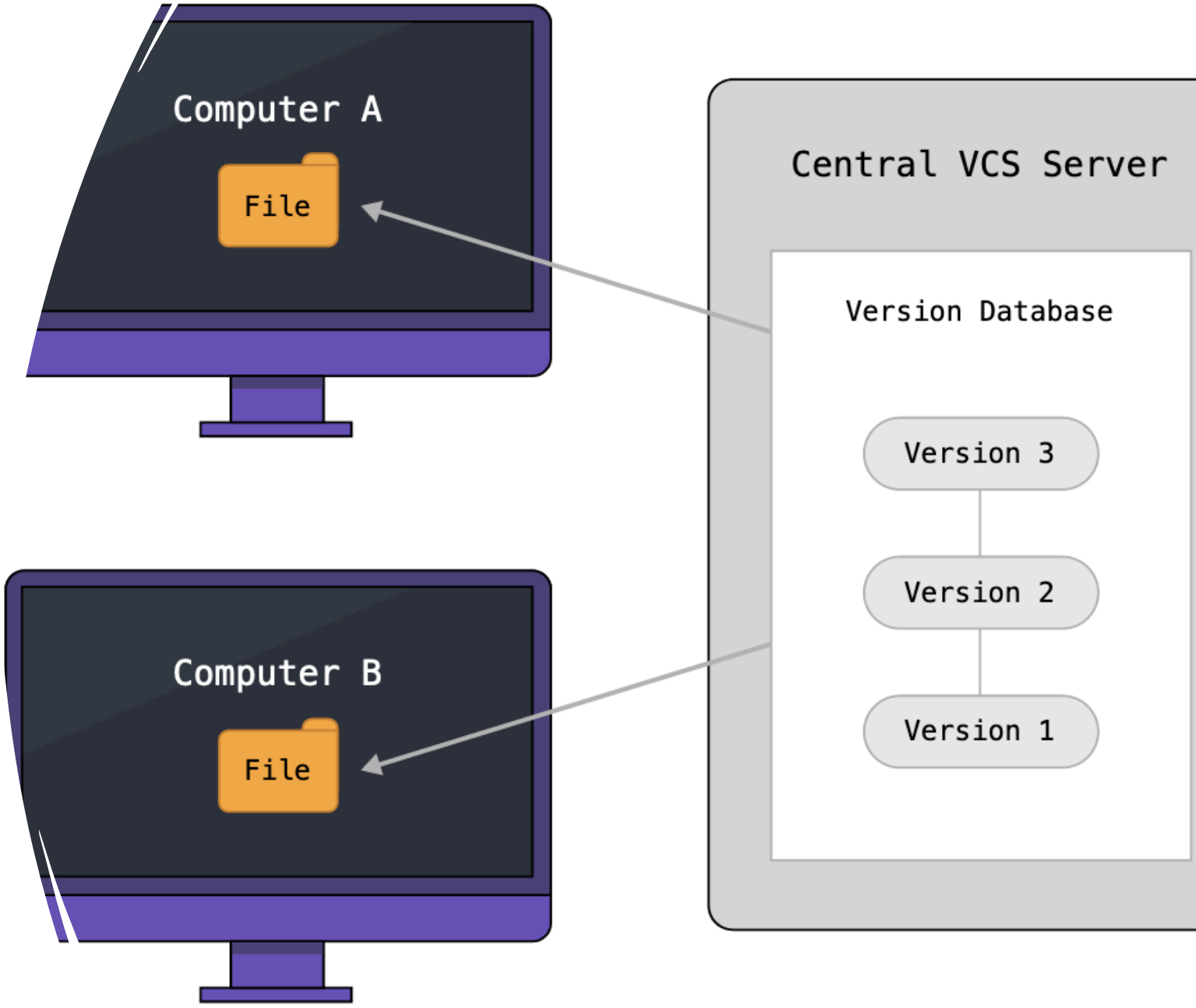
The Problems (Broadly)

- Inadequate Knowledge Management: Detachment and Individual Training
- Slides instead of Detailed Lesson Plans
- Lack of Feedback between Developer, Instructors, and Students
- Inadequate Version Control
- No Synchronized Training Content Updates

See Last slide for more

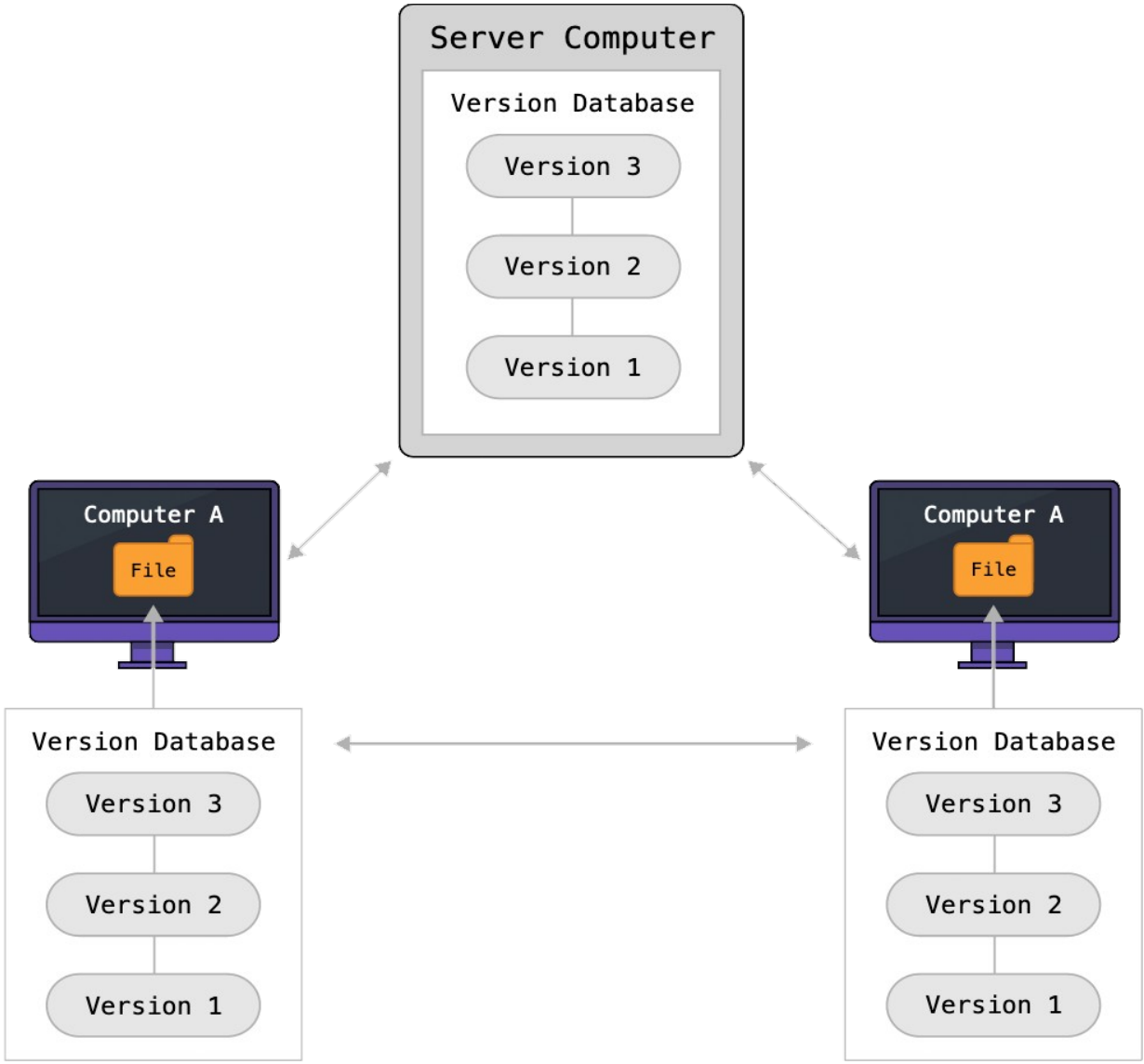
Knowledge Management Now

- TRADOC centralized system for approval and distribution of training (very exclusive) proponent level
- Limited access to developed training resources by units
- Limited interactions between
  - Instructors
  - Students
  - Operational Force



Better Knowledge Management

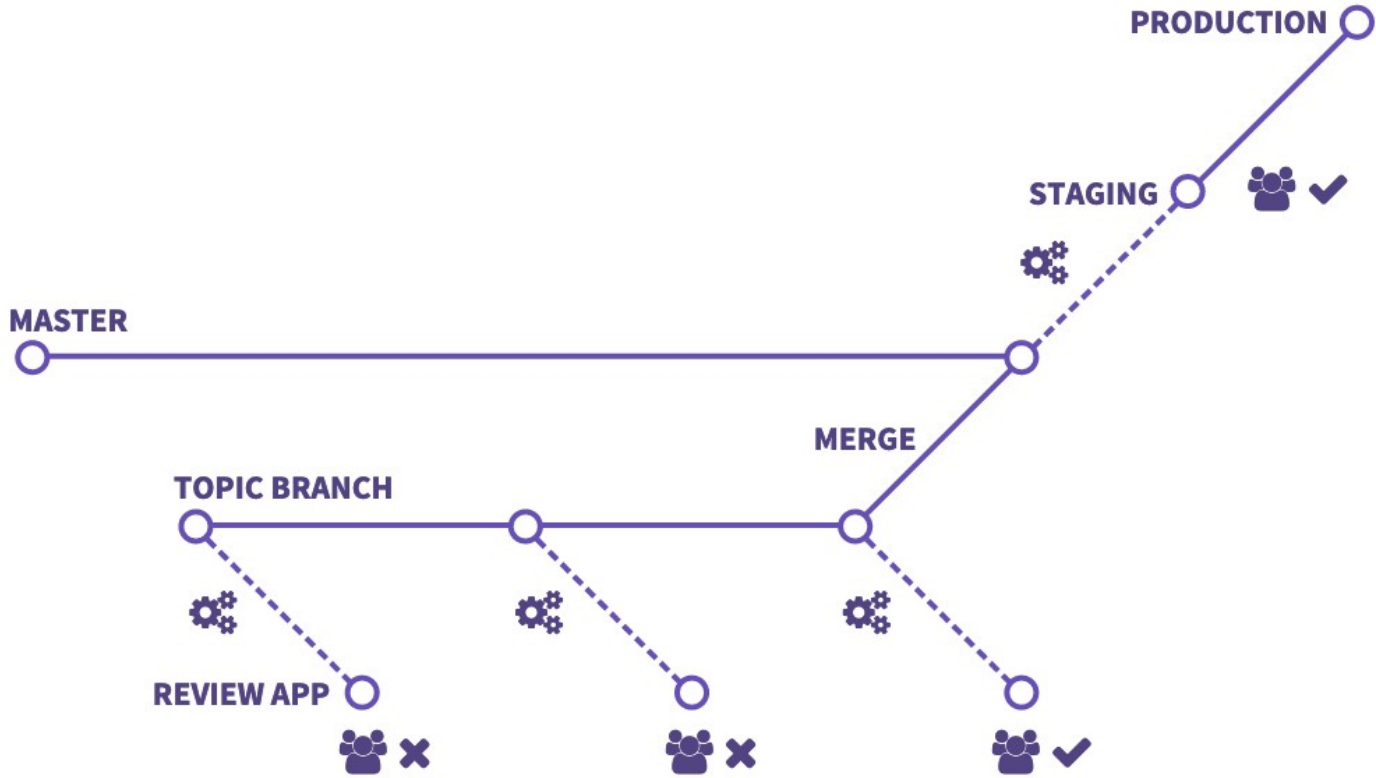
- Office365 real-time sync of training slides and instructor content
- Good
  - Rapid Updates
  - Some Permission Controls
- Bad
  - Version Control
  - Expanding on Training
  - Team Management
  - No Ground Level Feedback
  - Ground Level Updates
  - Word Formatted – Hard to Version Control



Git – Even Better

Git is an open-source, distributed version control system used for DevSecOps and Agile project management.

- A low-cost solution for providing a library of training to NCOs, unit trainers, and developers
  - PlatformOne (NIPR)
  - Licensed Gitlab (MME)
  - Locally Hosted (Group)
- Provides direct feedback from unit trainer and students for quick fixes and updates. (via issue tickets)
- Utilizes SCRUM and Agile frameworks for improved synchronization, collaboration, and planning.

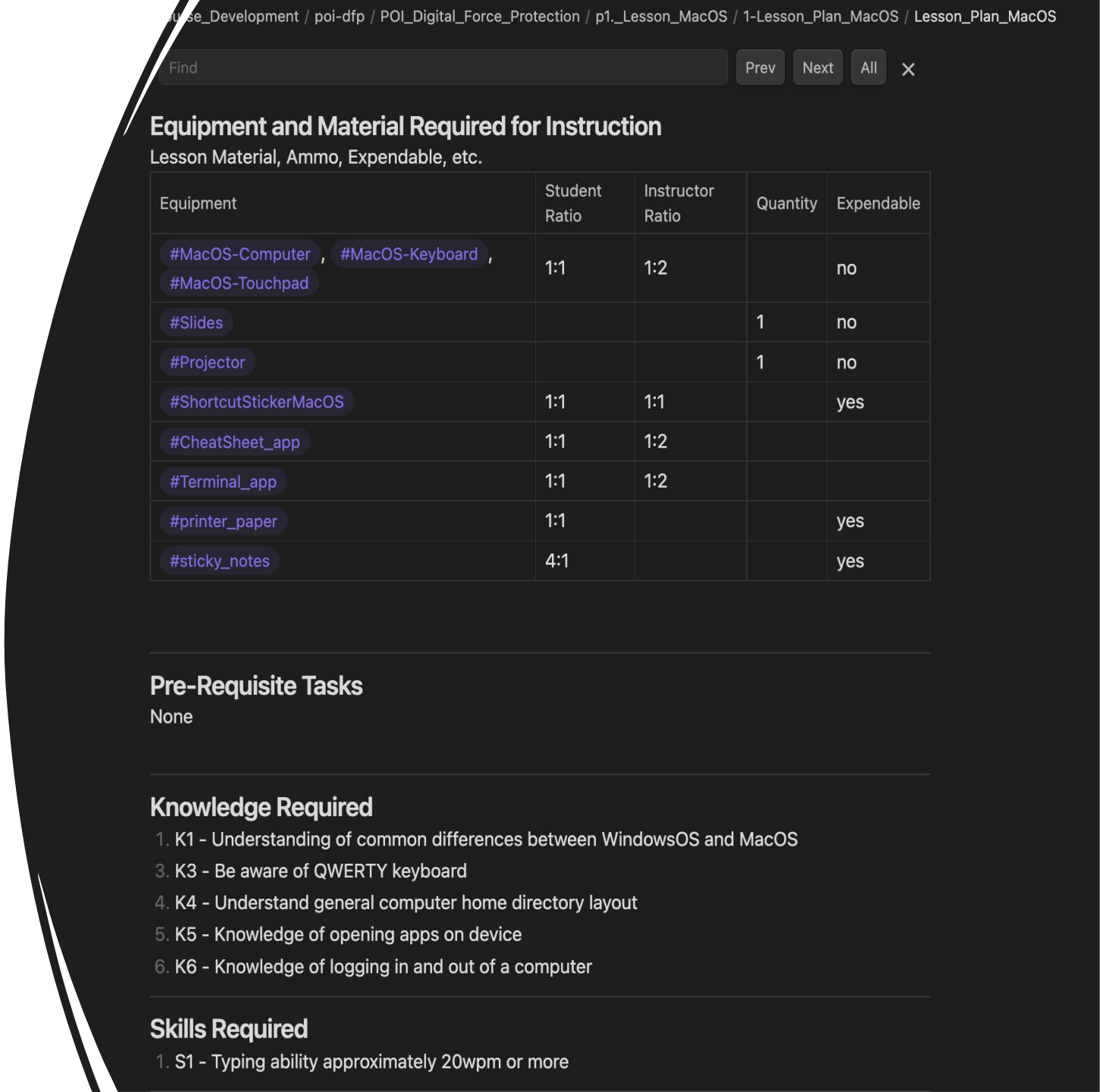


Current Problems with Unit Training Content

- Slides instead of Lesson Plans
  - Inconsistency in training environment setup (hard to communicate on slide-only content)
  - Unable to transfer actual instruction of content without extensive communication
  - Lack of transparency or processes to push updates and take feedback
- False Starts “recreating the wheel.”

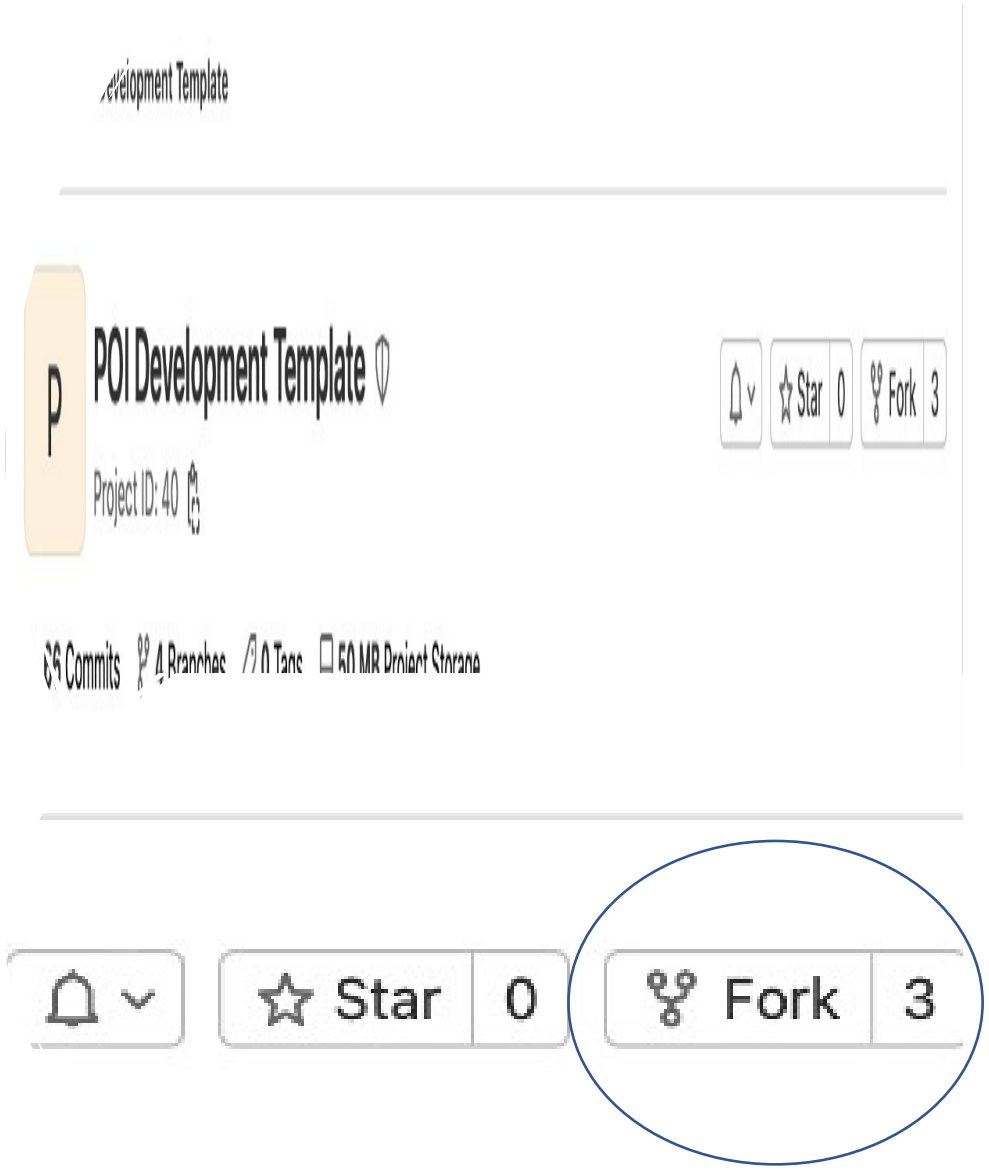
The Possible:
DET and Individual Lesson Plans

- Training Library using Git and Markdown Lesson Plans
  - Unit Training Knowledge Management
  - Instant Feedback (Issues)
  - Ground Truth Updates (Pull Requests)
- Lesson Plans
  - Lesson modularity is ideal for Unit Training limitations
  - Intent, Requirements, Configuration, and References available by default
  - Expand on who can teach



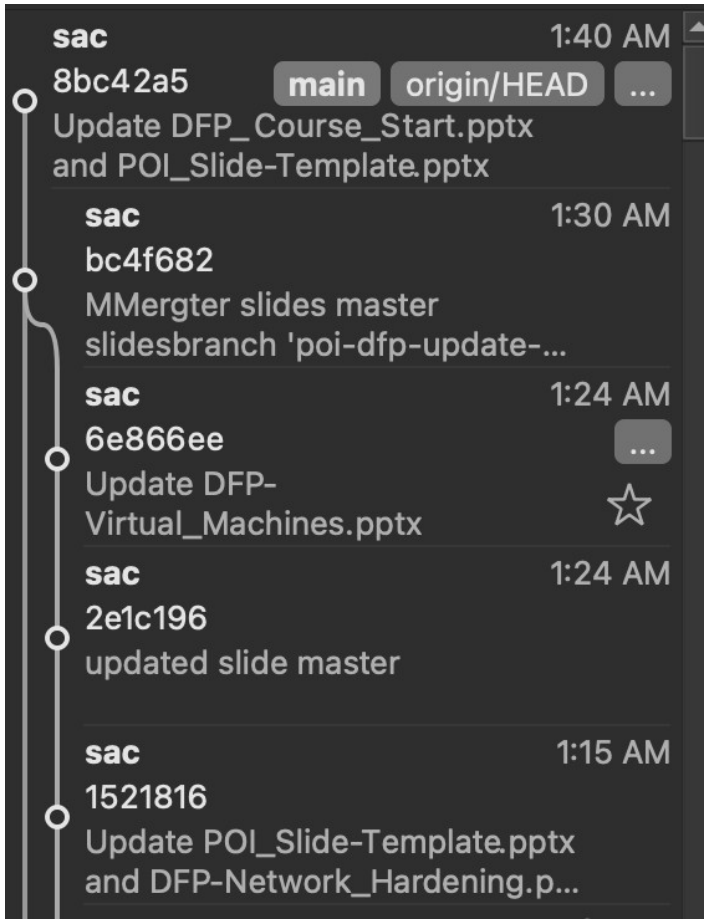
Use Prototype

1. Create Account (See SSO-Invite)
2. [POI Development Template](): https://git.irregularchat.com/TNGDEV
3. Create a Branch for a new Lesson



Beyond Prototype

- Move to DODIN
  - Gitlab (MME)
  - SIPR
  - Partners
- Expand Content
- Expand Access
- Develop Non-Formal Training for Unit Instructors
- Develop Command Level AMETL



References

- Why Git: https://page.gitlab.com/rs/194-VVC-221/images/gitlab-moving-to-git-whitepaper.pdf
- Learn Git Basics: https://yt.irregularchat.com/watch?v=RGOj5yH7evk
- PlatformOne: https://p1.dso.mil/resources
- Guidance for TRADOC Training and Education Development: TRADOC Pamphlet (TP) 350-70

Assumptions for the gaps identified in unit training include insufficient training on operationally relevant tasks, lack of training support packages (TSP), lack of training developed for partner forces, and lack of contracted training on capabilities available in support of operations.

- The Git infrastructure has a learning curve; however, students would not need to interact with the system in any way to receive training.
    - Students would instead have the option to leave comments using a familiar layout in the Git Issues section of training.
    - Unit Trainers would only need to know how to download already developed training and would have options to submit corrections easily with minimal knowledge of Git.
    - On the other hand, training Developers would need extensive training on using Git to use the system as intended. Developers can communicate with other unit developers but do not need to be Subject Matter Experts (SMEs) in the training content. For this reason, developers should attend the Instructional Design Course.
- Detachment and Individual level training is currently being developed from scratch or by reverse engineering performance steps of evaluations. The Non-Commissioned Officers (NCOs) developing this training are not taking advantage of experiential learning methods or other instructional design standards because they are unaware of the methods or resources. Additionally, training lacks adequate knowledge management, approval, dissemination, and update mechanisms. These mechanisms exist in proponent-level tools, such as the ITN, unavailable to unit-level trainers. Without these mechanisms, units continuously recreate training without fully benefiting from lessons learned in previous training iterations within their own unit or the entire community training on the same objectives.
- PowerPoint slides and hands-on imitation are nearly exclusively used to develop unit training. These slides do not adequately relay the intention of the training, the requirements for hardware, software, handouts, media, and configuration, or the context for unit trainers. Trainers who are subject matter experts at their craft often lack instructional design and instructor training, leading to the less effective transfer of knowledge and lower operational effectiveness and application.
- Students of Unit training often provide feedback to the trainer directly but do not have established processes to provide feedback to the training developer. This feedback can range from minor fixes to changes in the operational ground truth. Developers struggle to obtain this ground truth via emails, surveys, AARs, and interviews.
- Changes are made without clearly identifying what has been changed and why, as a change to a single sentence represents a complete change to the entire POI file with the current systems. With Git, changes are automatically labeled on precisely who changed what, when, how, and why. This changelog, along with detailed lesson plans, helps trainers to accurately train Units to the standards and intentions of the course developer.
- Contracted training is often viewed as “Trained for Life” and does not provide commanders with an accurate understanding of a Soldier’s current proficiency. Using a Git unit training library will allow further integration with contracted training and allow units to build off contracted training to develop more effective TTPs and better understand individual competency.
- Unit training content currently ranges from accessible on MilSuite or iSpace to being stuck on someone's computer or email. Allowing units to build off each other’s training content and to have a library of training will rapidly advance the quality and quantity of available training and enable units to train to standard more often and better understand what is required.