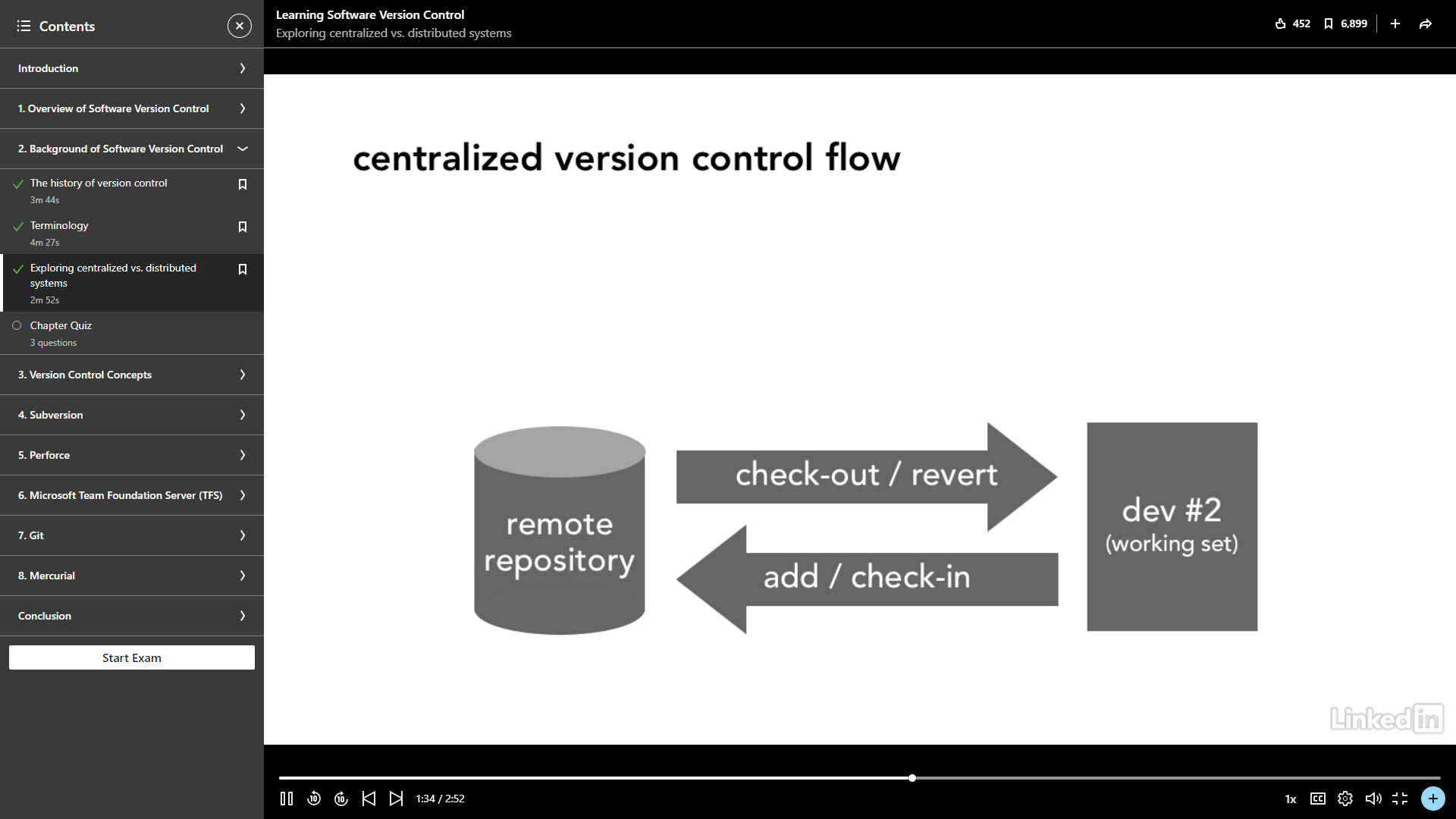
**Software Version Control System**

In software engineering, version control is a class of systems responsible for managing changes to computer programs, documents, large web sites, or other collections of information.

* Keep track of your creative output.
* It tracks what is changed.
* It tracks who makes the changes.
* It tracks why changes were made.

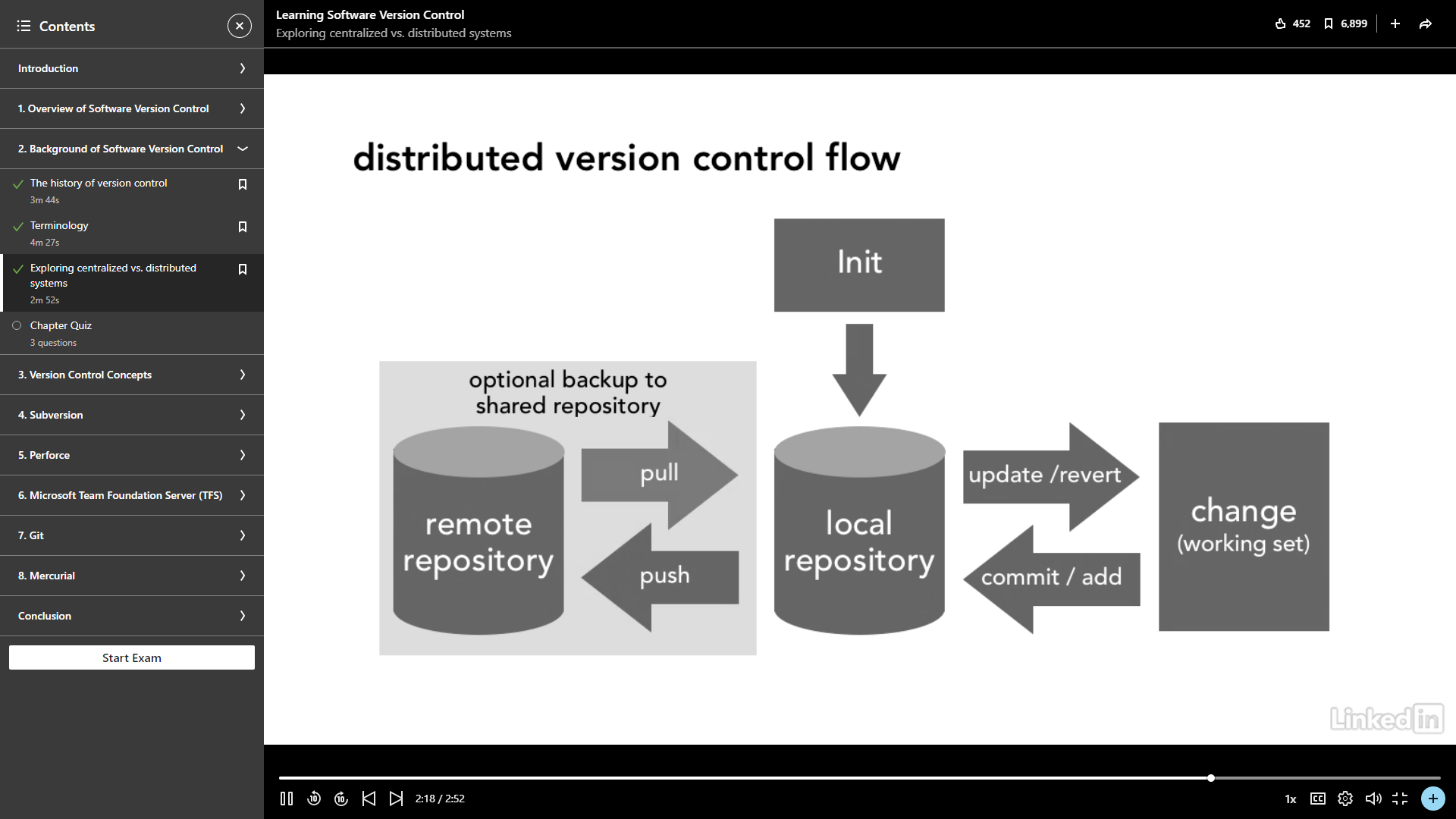
**Centralized Version Control System**

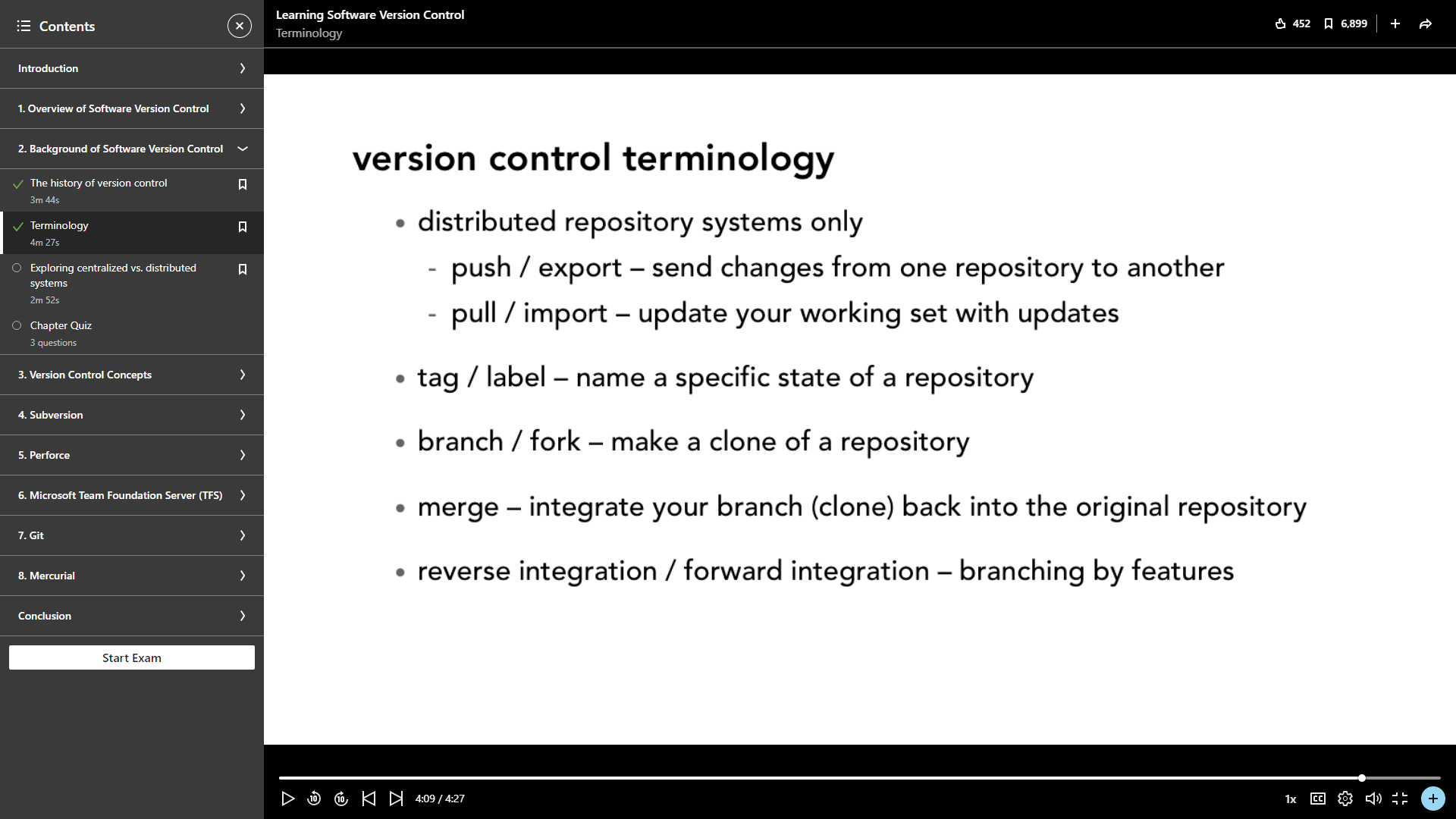
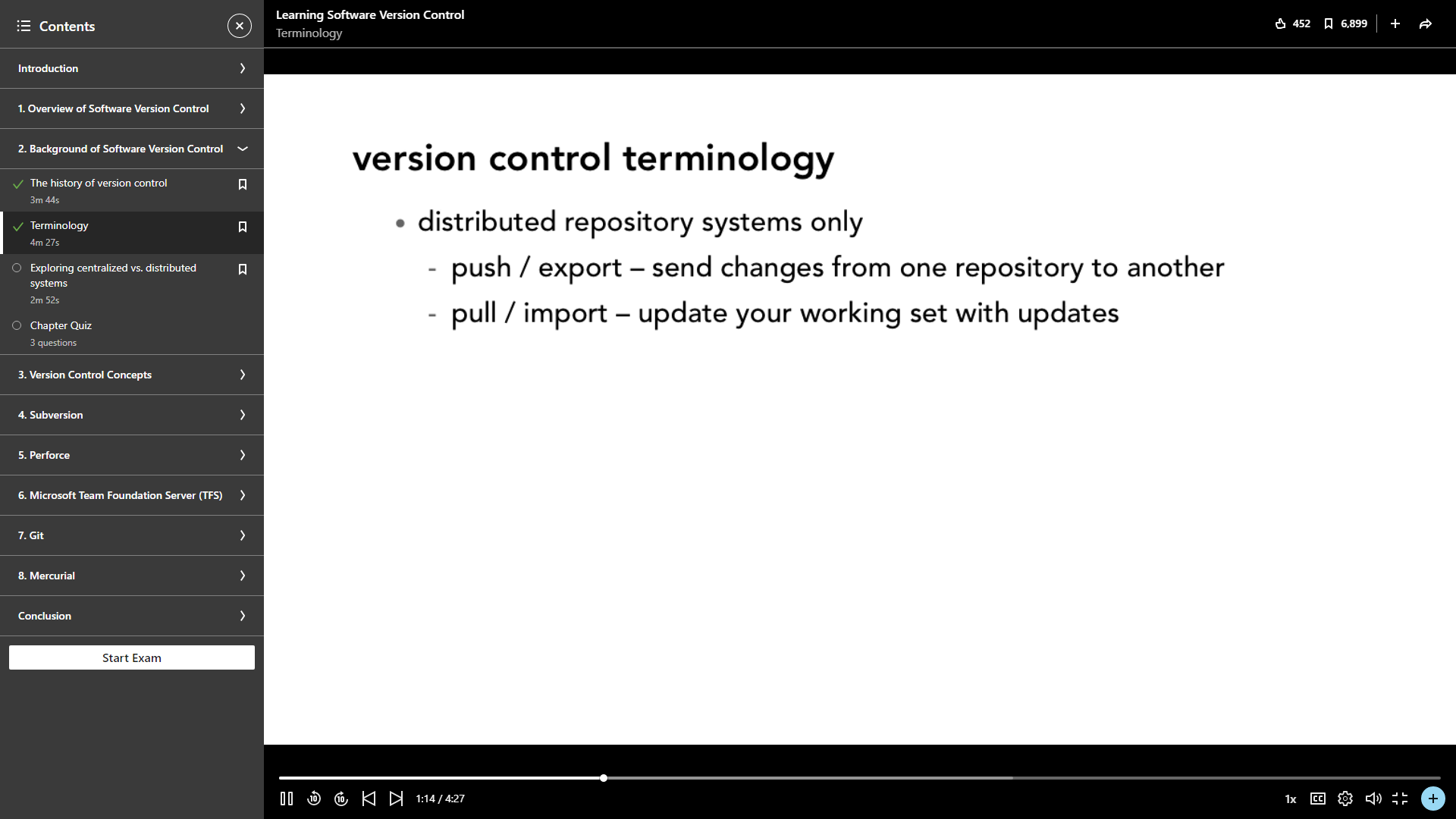
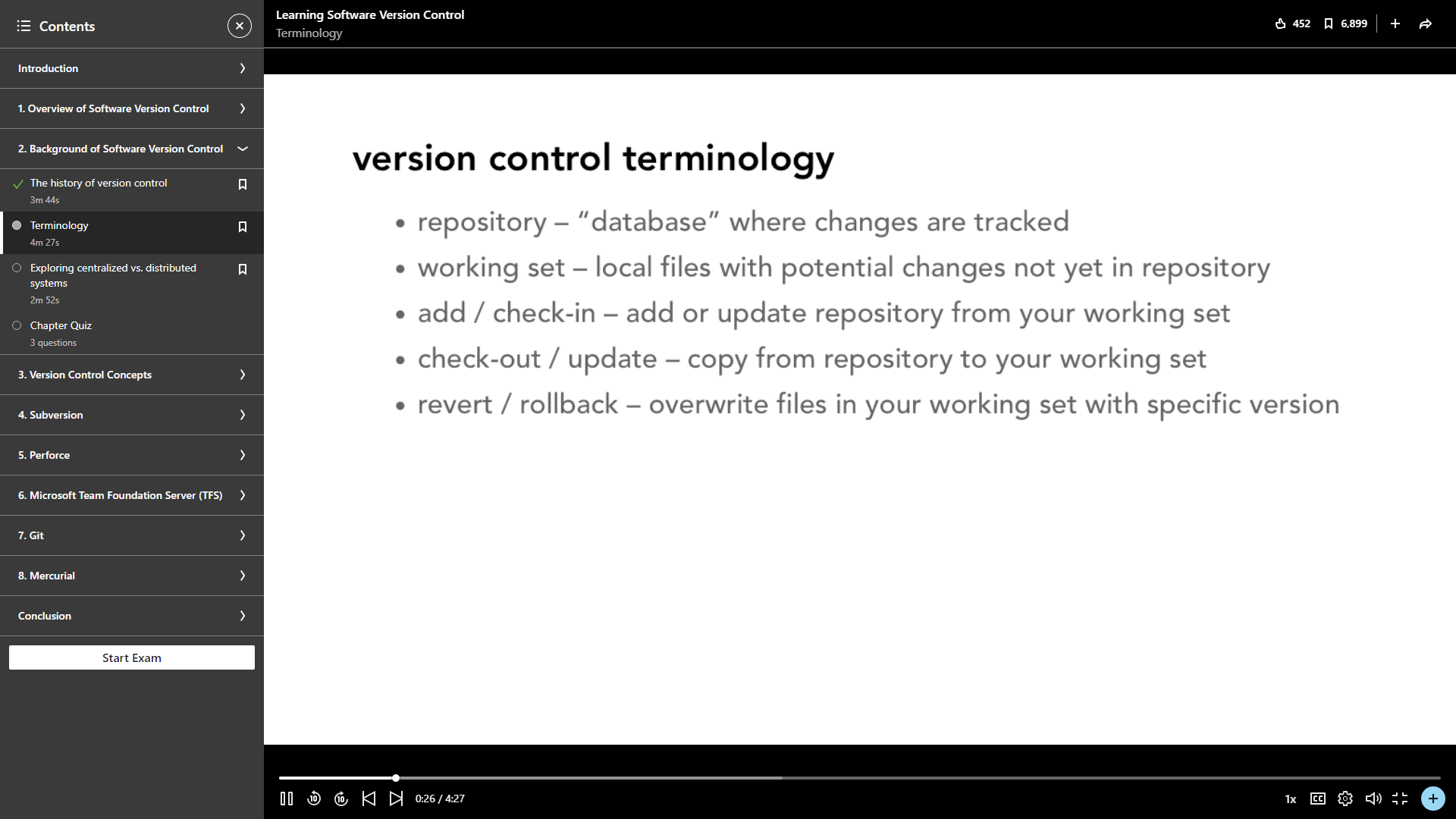
In a centralized [version control system](https://about.gitlab.com/topics/version-control/), a server acts as the main repository which stores every version of code. Using centralized source control, every user commits directly to the main branch, so this type of version control often works well for small teams, because team members have the ability to communicate quickly so that no two developers want to work on the same piece of code simultaneously.



**Distributed Version Control System**

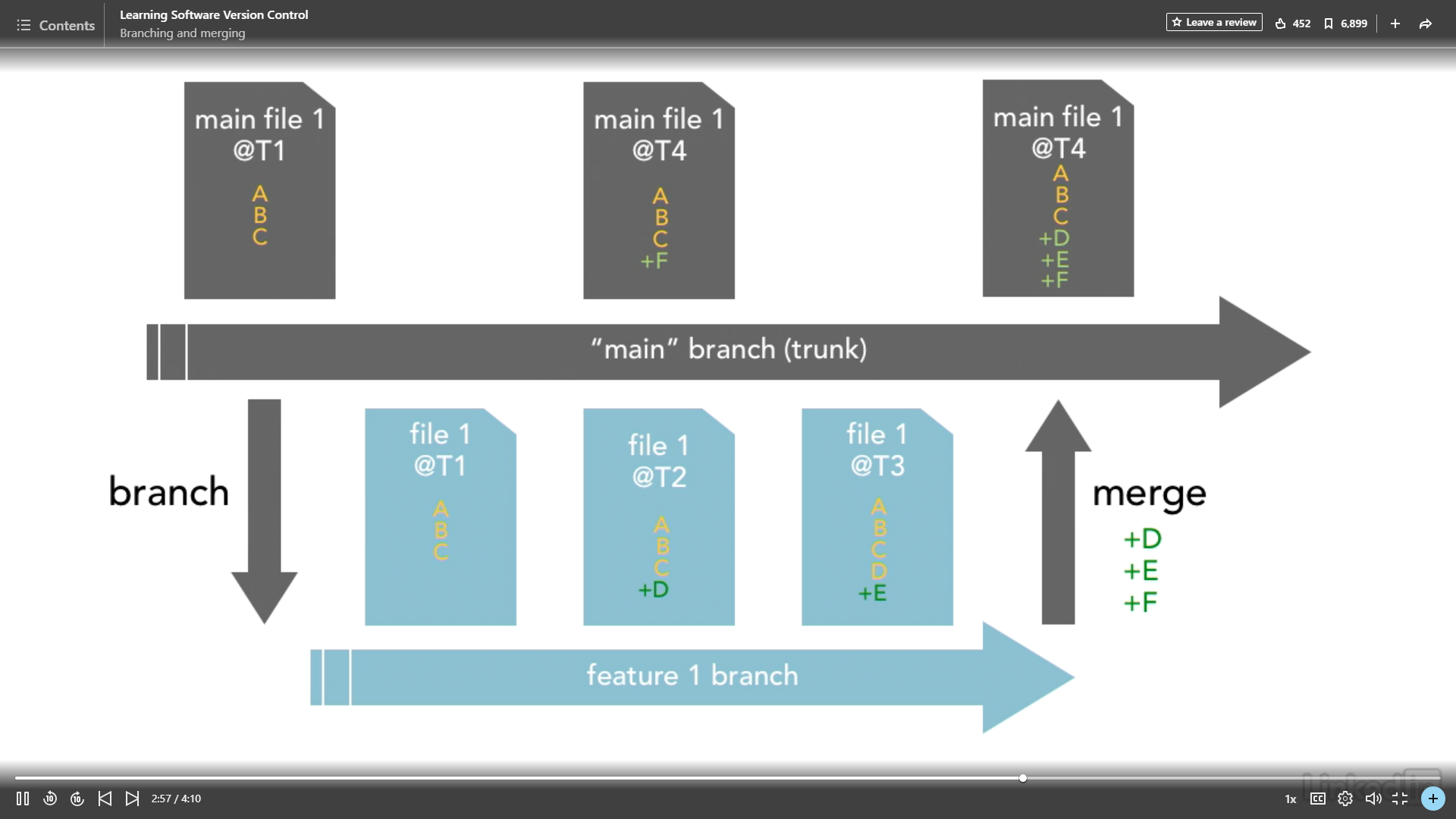
A distributed [version control system](https://about.gitlab.com/topics/version-control/) (DVCS) brings a local copy of the complete repository to every team member’s computer, so they can commit, branch, and merge locally. The server doesn’t have to store a physical file for each branch — it just needs the differences between each commit.

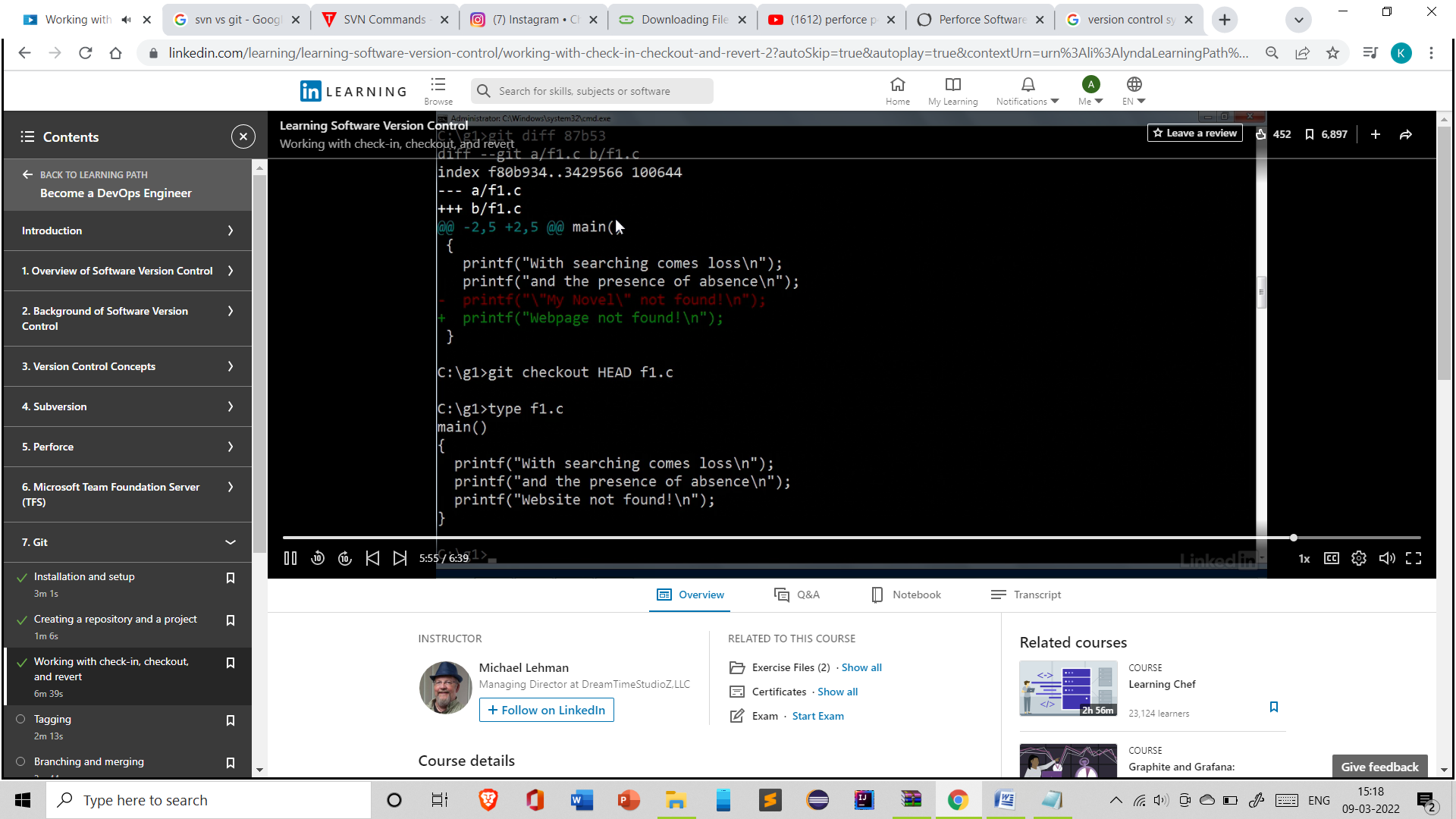
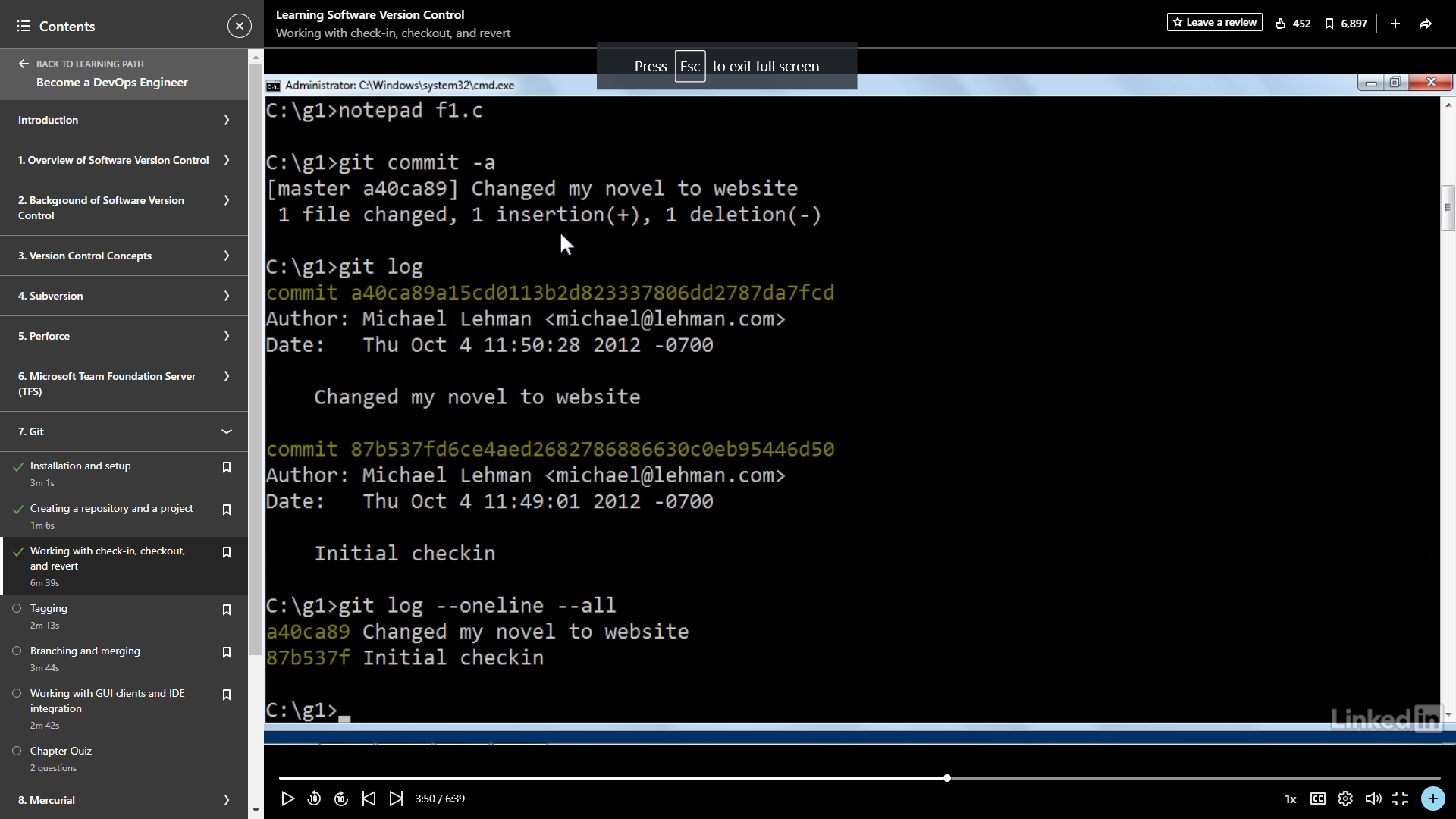
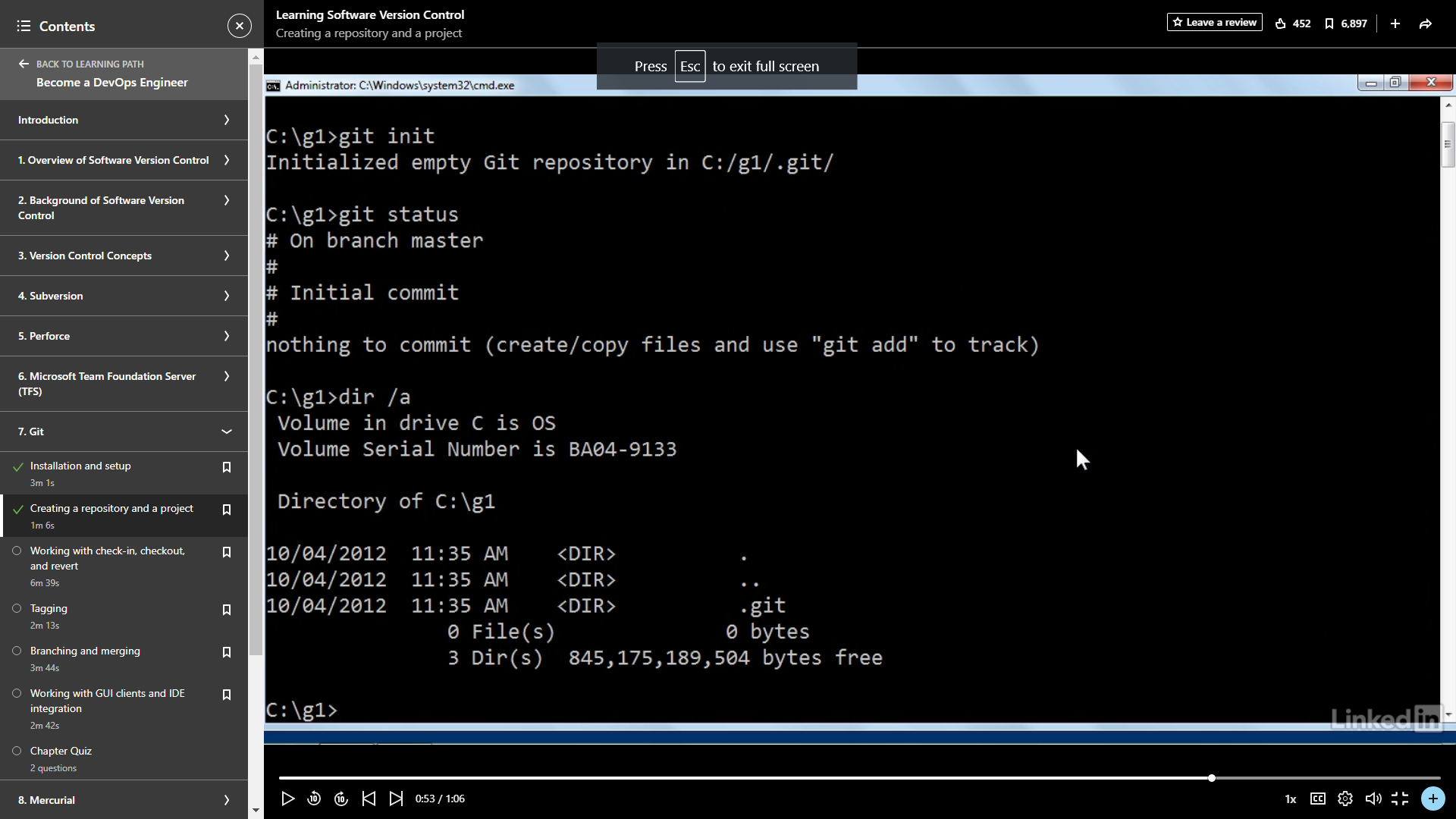


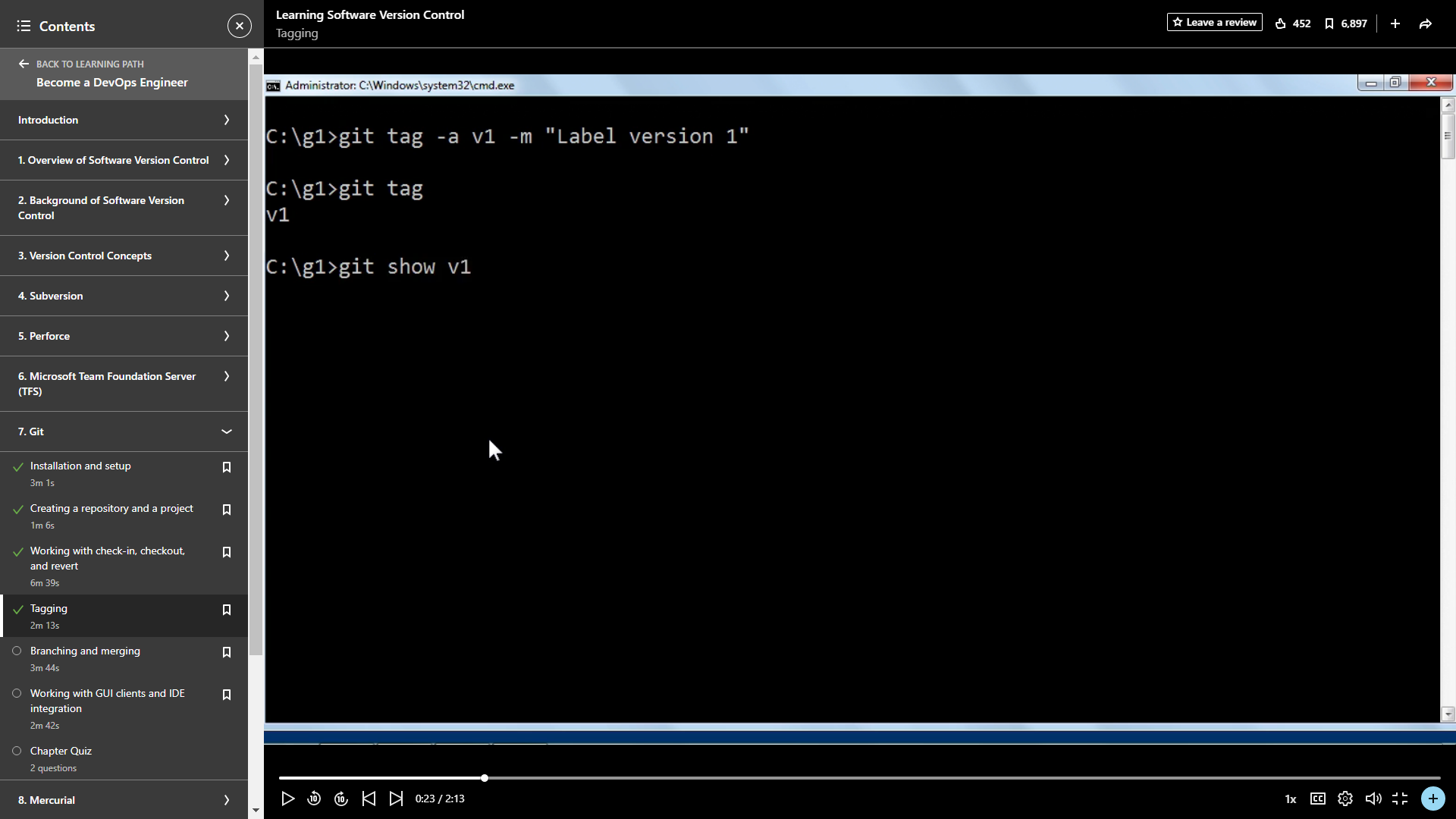
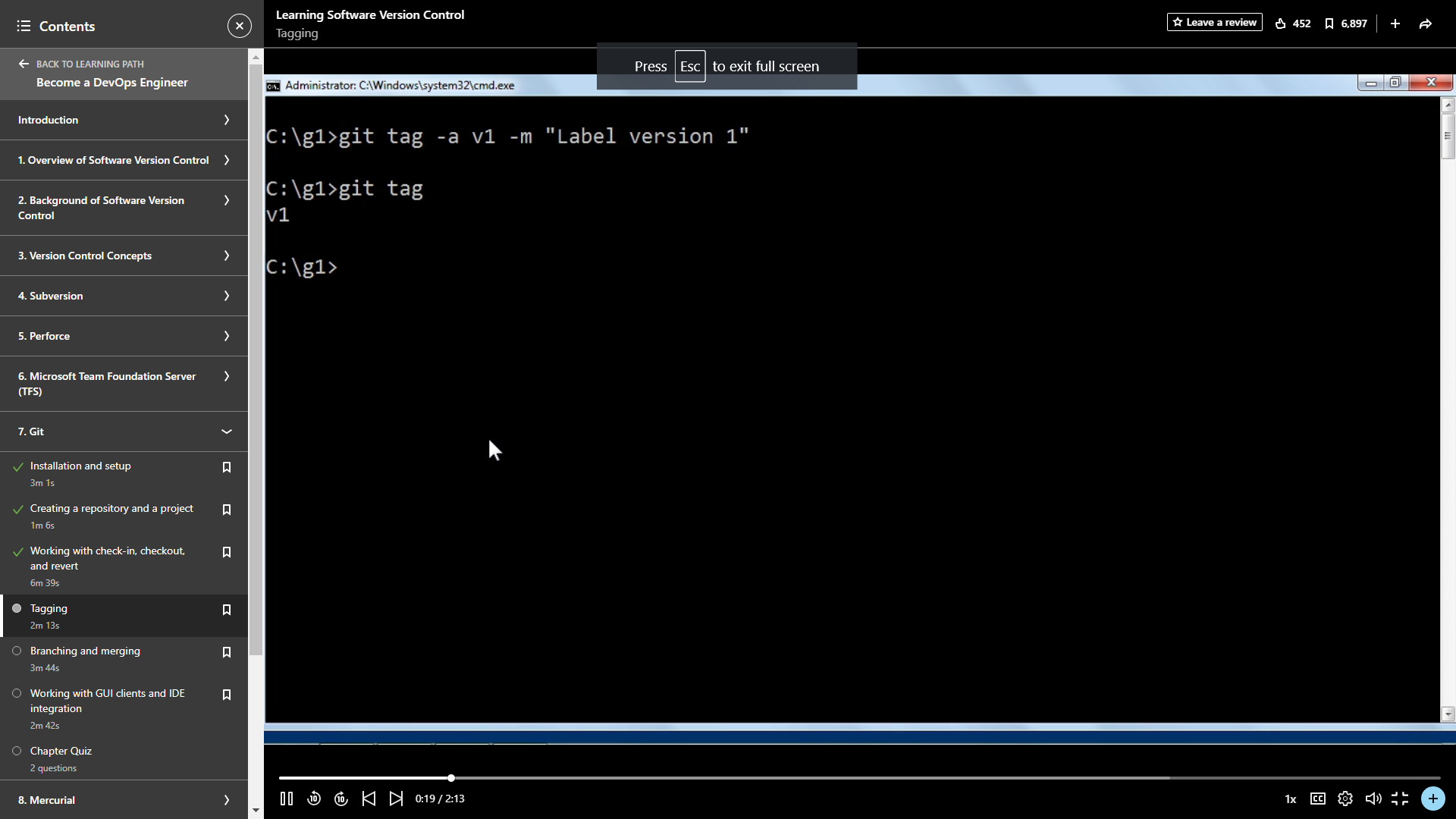


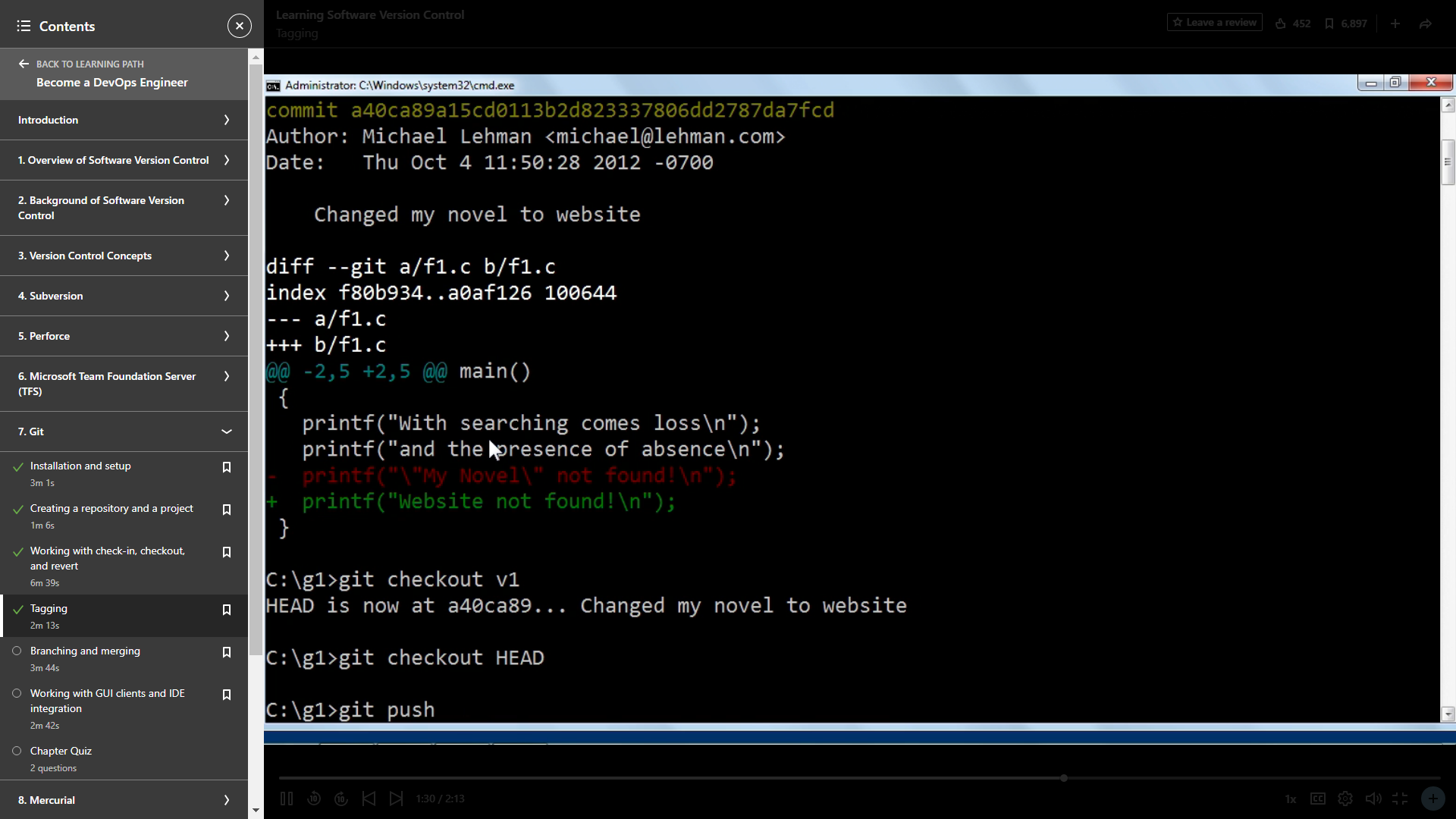
**Branching and Merging**

Once you've completed work on your branch, it is time to merge it into the main branch. Merging takes your branch changes and implements them into the main branch.

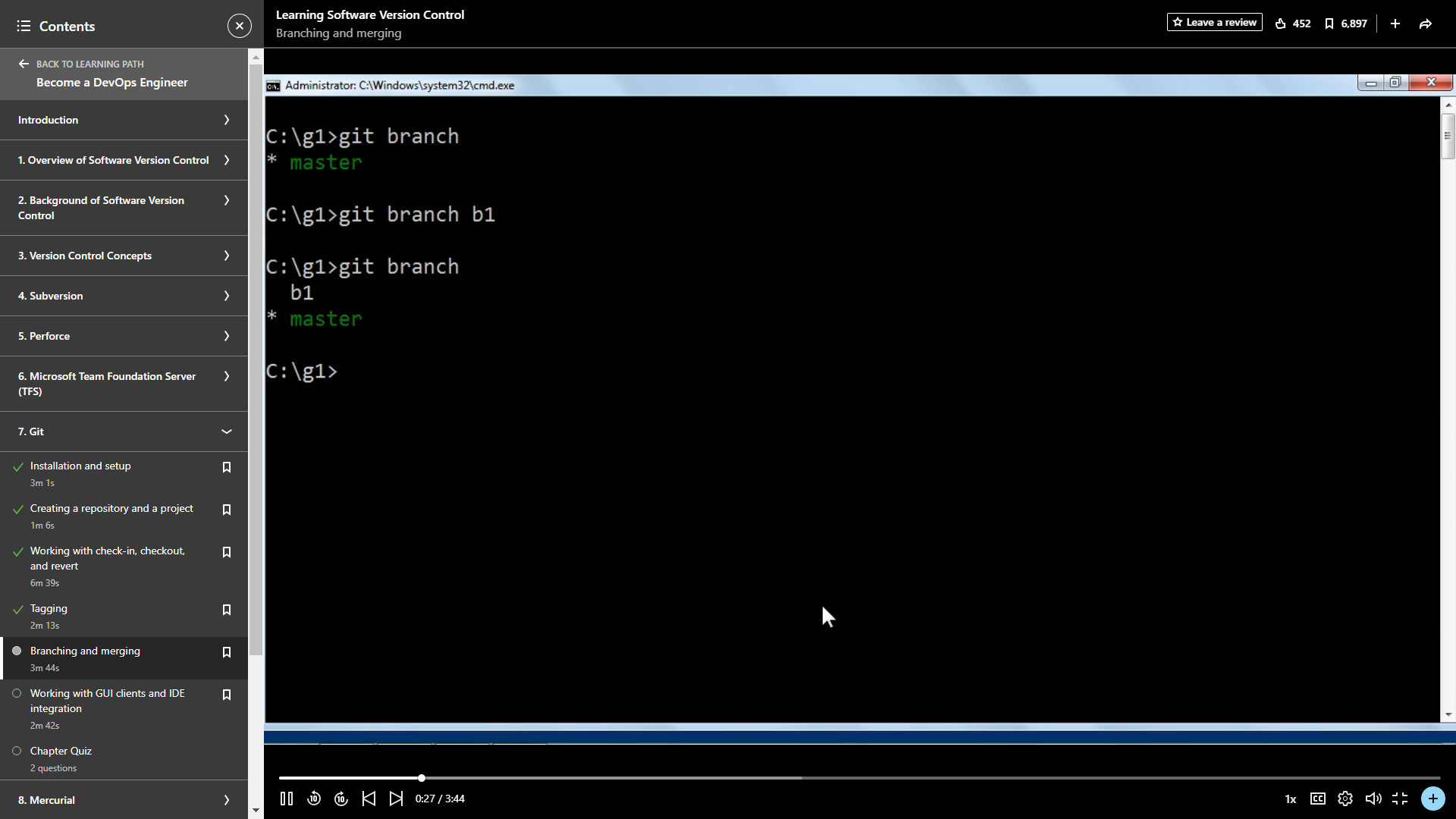
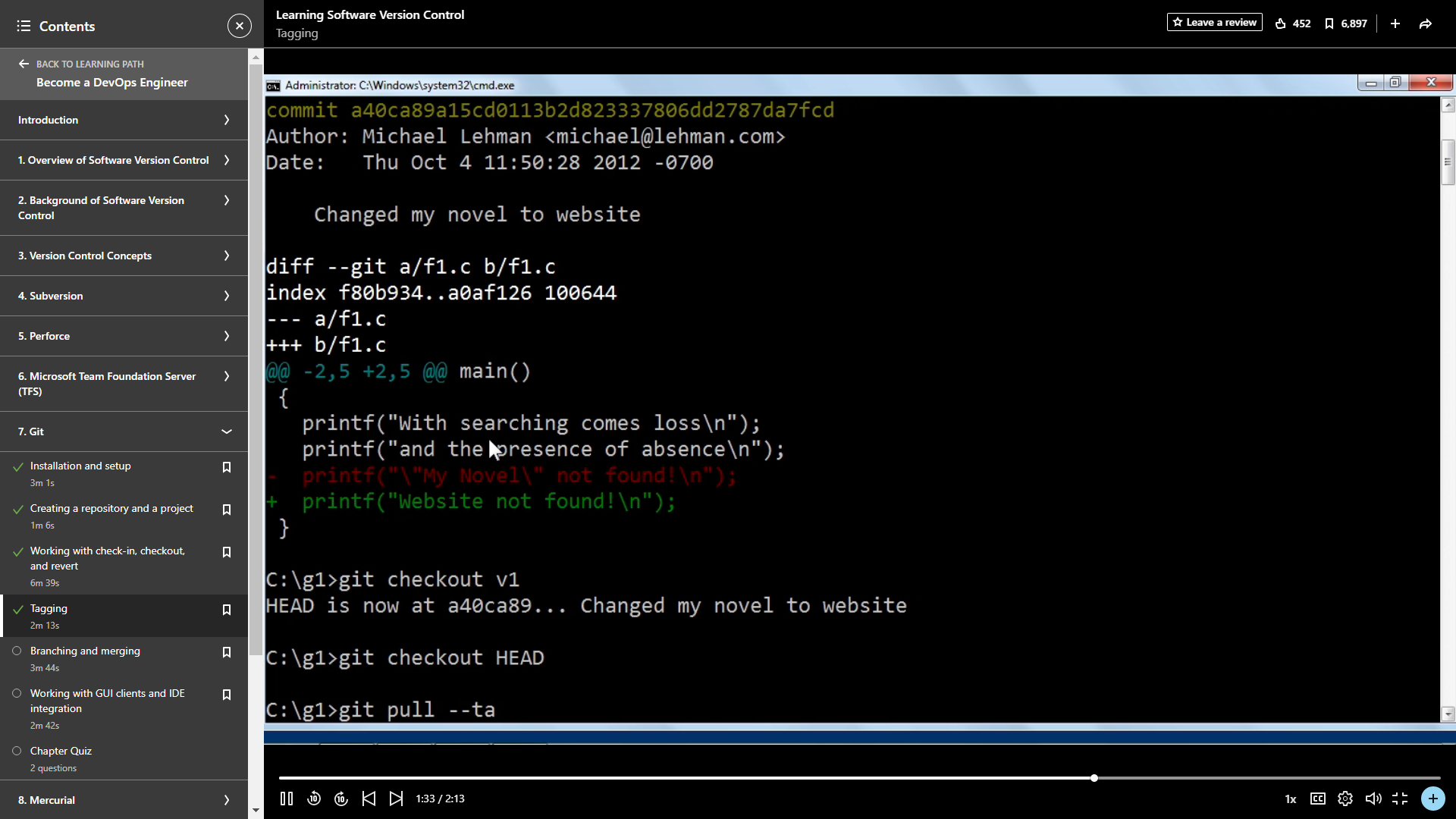


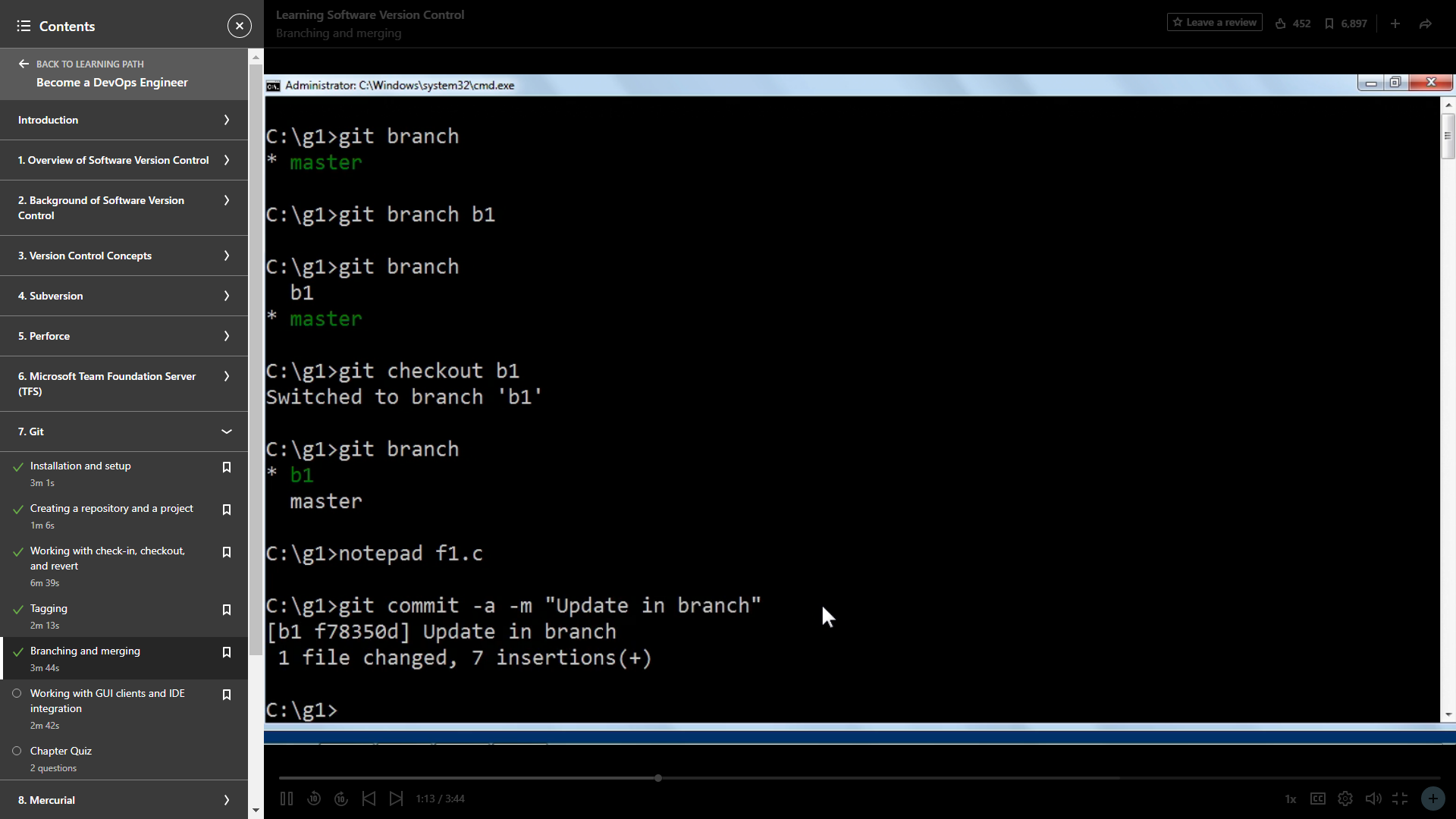


Tagging



Branches makes it easy to work with the repositories while not making the change to the main repositories.





These are the software version control system in which git is the best as **Git is a distributed version control system and better in many ways.**

