7.10.2021

BAIN\_Blogbeitrag\_2

Als «nicht so Terminal-affin» habe ich mich ja bereits in vorhergehenden Blogbeiträgen geoutet. Nach der zweiten Lerneinheit muss ich feststellen, Terminal und ich, wir werden in diesem Leben keine grossen Freunde mehr.

Das Vorgehen der Dozenten war absolut nachvollziehbar und die Erläuterungen zu GitHub und GitLab lösten einen Aha-Moment aus. Ich hatte mich tatsächlich auch schon gefragt, warum es zwei solche Plattformen gibt. Gefallen hat mir ausserdem, dass Inputs, Erwartungen und Fragen von uns Studierenden zu Beginn der Lektion aufgenommen und behandelt wurden. Das nimmt dem Unterricht etwas von seinem Frontal-Charakter und fühlt sich nach Dialog an.

Aber zurück zu meiner Terminalscheu. Das Thema Versionskontrolle mit der relativ universalen Software Git erschliesst sich mir theoretisch. Grob gesagt ermöglicht es die Versionsverwaltung von Projekten, an denen mehrere Personen beteiligt sind oder an dem ich an verschiedenen Teilbereichen werkle und sie erst später zusammenführe. Es geht nichts verloren und die dokumentarische Eigenschaft von Git ermöglicht es auf ältere Arbeiten zurückzugreifen oder den Entstehungsverlauf eines Projekts nachzuvollziehen. Soweit so gut. Ab dem Moment, in dem ich das Ding via Terminal in der Linux-Umgebung auf meiner Virtual Machine installieren soll, biegt jegliches theoretisches Verständnis mit einem spektakulären U-Turn ab und ich frage mich bei jedem Schritt, was mache ich hier eigentlich. Sudo apt, clone, config, status, add, commit und wie sie alle heissen, die Befehle sind oft selbsterklärend. Trotzdem verschwimmt bei mir im Eifer des Gefechts zum Teil der Überblick. Sind wir hier noch bei einem Befehl, ist das hier eine Option, was von dieser Zeile ist jetzt schon wieder das Argument??! Wahrscheinlich sollte ich das nach ARIS alles wissen. Als ich dem Script im HedgeDoc hinterher-copy/pasted bin, blieb an jedem kleinen Erfolg auch das Gefühl einer Hochstaplerin hängen. Jede Befehlszeile wurde von den Dozierenden geduldig erklärt, dafür bin ich wirklich dankbar. Allerdings blieb mir beim hin- und herhechten zwischen Webex, Terminal und HedgeDoc meist nicht die Zeit noch konzentriert zuzuhören. Vielleicht möchte ich, sobald ich eine shell sehe, viel zu genau wissen, was ich dem Rechner da gerade antue. Bei einem GUI mache ich mir diese Gedanken nie – wird sich schon wer was überlegt haben zu diesem grossen grünen Install-Now-Button… bedenklich.

Die Übung mit den Metadatenstandards MARC21 und Dublin Core (DC) liess erahnen welchen Hafen wir als nächstes ansteuern. Der Scriptvergleich hat diese beiden Datenformate endlich «anfassbar» gemacht. Für mich als Nicht-Bibliothekarin war MARC21 immer etwas sehr Abstraktes und hat sich mir nur ansatzweise über die Funktion erschlossen. Von Dublin Core gar nicht zu reden. Aus einer Recherche zum Thema Metadatenschemata für eine Semesterarbeit vor etwa anderthalb Jahren wusste ich knapp noch, dass DC offener und in einem begrenzten Rahmen flexibler für die Bedürfnisse verschiedenartiger Institutionen entwickelt worden ist, als z.B. MARC21.

Mit der Installation von Koha (Hello Terminal again) sind wir vermutlich bei weiteren Hands-on-Übungen zu Bibliothekssystemen angekommen. Allen Terminal-Widerständen und dem Abstraktionsniveau von Metadatenschemata in Bibliotheken zum Trotz, gehe ich zuversichtlich in die dritte Lerneinheit. Die Unterrichtsform motiviert dem Thema zu folgen, obwohl ich nach dem Studium wohl nie mehr etwas mit MARC21 zu tun haben werde.