

Alles unter Kontrolle – Version für Version

Informatisch nachhaltiges Versionieren

Wer kollaborativ arbeiten möchte, muss meist vorher die papierbasierte Arbeitsorganisation informatisch modellieren: Bei jedem Austausch werden Dateien, ja ganze Dateibäume, komplett ausgewechselt, die einen Arbeitsstand darstellen. Sie müssen dann erneut aufgerichtet werden. Daraus resultiert die immer wiederkehrende Übertragung großer Datenmengen. Die hier vorgestellte Lösungsvariante modelliert den Prozess als Arbeit, die dezentral organisiert wird, und es ermöglicht, ausschließlich die Daten zu übertragen, die geändert wurden (auch bei binär vorliegenden Daten, wie beispielsweise PDF-Dokumenten). Das verknüpft die kollaborative Arbeit mit informatischer Nachhaltigkeit; die Beteiligten können ökologisch verantwortungsbewusst und ökonomisch zielorientiert über verschiedene Informatiksysteme hinweg zusammenarbeiten.

Versionskontrollsysteme, etwa *git* oder *svn*, stützen das kollaborative Arbeiten durch gestufte Synchronizität und Möglichkeiten zur Verzweigung in Teilprojekte, die zu einem späteren Zeitpunkt wieder zusammengeführt werden können. Sie sind unabhängig von Plattformen und portierbar. Wie bereits diese globalen Eigenschaften belegen, sind Werkzeuge zur Versionskontrolle auch für Vorhaben ausgezeichnet geeignet, bei denen keine Software produziert wird: Sie schonen lokale Speicherressourcen, weil sie Redundanz vermeiden (Beispiel: von *mein-dokument.txt* zu *mein-dokument2.txt* zu *mein-dokument3.txt*). Außerdem wird die Menge zu übertragender Daten dank des differenziellen Transfers *erheblich* gesenkt. Die Einreichung und der Abruf von Daten erfolgen über *eine* konsistente Verzeichnisstruktur statt über E-Mail-Anhänge, die Mail-Clients und Mail-Server belasten. Die typische E-Mail mit 13 MB Anhang ist eine potenzielle Gefahrenquelle, weil Schadsoftware die E-Mail-Kommunikation unterwandern kann: die gefährliche E-Mail mit der *installiermich.exe*. Gestuft eingereichte Versionen lassen sich kommentieren und *mergen*; bisher endlose E-Mail-Beschreibungen sind kürzer oder werden ganz überflüssig. Das spart lange Telefonate oder Videokonferenzen. Was für Software-Entwicklung selbstverständlich ist, empfiehlt sich ausdrücklich auch für prinzipielle Dokumentenverwaltung, stets auf Basis rein textbasierter Dokumente zu arbeiten. Versionskontrollsysteme laden verstärkt dazu ein, da sie bei den rein textbasierten Formaten (wie *md*, *txt*, *csv*, *tex*) sehr komfortabel die Differenz zwischen zwei Versionen veranschaulichen; das ist insbesondere beim Mischen

verschiedener Versionsstränge hilfreich. Mit der reinen Textbasis eines Dokumentes können Konvertierungswerkzeuge (wie *pandoc* oder eigene Scripte, ...) eine für Druck- und Präsentationszwecke geeignete Layoutsicht generieren. Dadurch lässt sich schließlich eine nachhaltige Aktualisierung von *Produktkränzen* (*pdf*, *html*, *odt*, *docx*) durch die Modifikation *einer einzelnen* Datei (*md*, *txt*, *csv*, *tex*) erreichen.

Welche Kompetenzen sollte ich entwickelt haben? Aspekte der informatischen Modellierung

Um ein Versionskontrollsystem informatisch vernünftig anzuwenden, bedarf es schließlich ausgewählter Kompetenzen aus dem Bereich informatischer Modellierung:

- Ich kann mir Dokumentenstrukturen und Verzeichnisbäume selbständig erschließen, sie verändern und neu aufrichten.
- Ich kann die Trennung von *Struktur – Inhalt – Form/Layout* begründen und praktisch umsetzen/anwenden.
- Ich kann diverse Schnittstellen zur Nutzung grundlegender Verzeichnisstrukturen anwenden: *shell first, gui second*.

Was meinen die Leserinnen und Leser der FIF-Kommunikation? Wir freuen uns auf Ihre Anregungen zur Diskussion.



Ludger Humbert und Daniel Losch

Ludger Humbert ist seit 1976 Diplom-Informatiker, Lehrer für Informatik und Mathematik an der Willy-Brandt-Gesamtschule in Bergkamen sowie seit 1997 Fachleiter für Informatik im Zentrum für schulpraktische Lehrerbildung in Hamm. Seit 2010 ist er Honorarprofessor an der Bergischen Universität Wuppertal und leitet dort das Fachgebiet *Didaktik der Informatik*.

Daniel Losch ist Lehrer für Informatik und Latein; seit 2017 forscht er als wissenschaftlicher Mitarbeiter an der Bergischen Universität Wuppertal im Projekt *Kohärenz in der Lehrerbildung* (im Rahmen der *Qualitätsoffensive Lehrerbildung*). Sein Forschungsschwerpunkt liegt in der Untersuchung der Bedeutung textueller Darstellungsformen der Informatik für die allgemeine LehrerInnenbildung.