

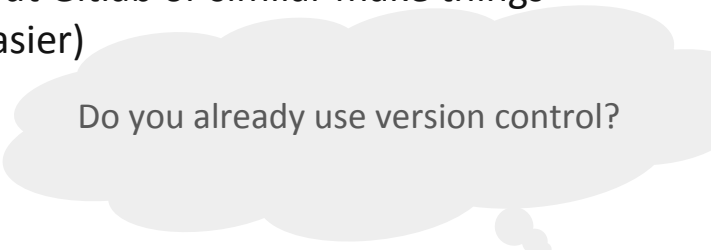
## CSV files on Git

### Pro

- No server only software on clients needed (but Gitlab or similar make things easier)
- No special frontend (“editor”) needed: lean development
- Version control helps to track changes and reset to previous version in case of errors
- Metadata easy accessible for programming (Ruby, R, Stata)
- Establish version control know-how

### Con

- Integrity of metadata not enforced
- Annoying issues with separators, encoding, quotes (LibO Calc and a macro helps)
- Transfer to database (for web-use) No server only software on clients needed (but Gitlab or similar make things easier)



Do you already use version control?

## Example: What DDlonRails preserves and adds

### Preserved

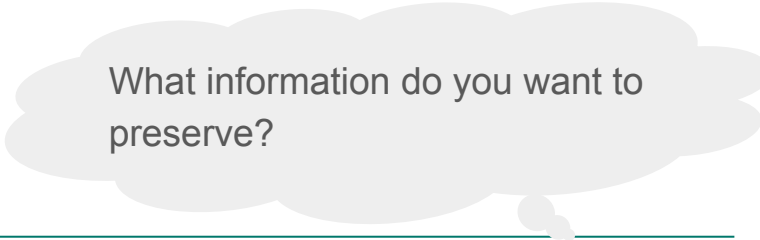
- Question numbers
- Textual information (question texts, instructions, answers)
- Routing (logical: filter, goto)

### Added

- Values for answers
- Concepts
- Links to variables (DDI QuestionsReference)
- Translations

### Not preserved

- Layout (horizontal/vertical arrangement, text prior/after open ended questions)
- Typography (bold, underlined)
- Graphical information
- Routing (textual)



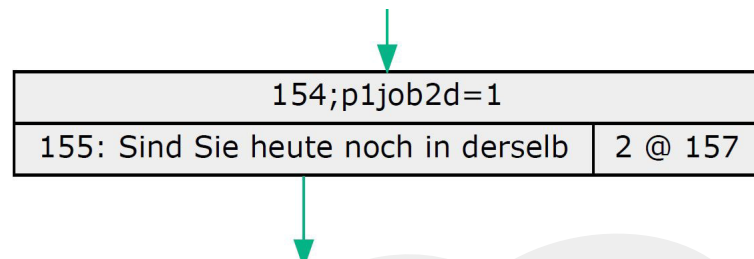
What information do you want to preserve?

## Some Notes on Routing

- Common default: go to next question
  - No more specification needed
  - Exceptions needed

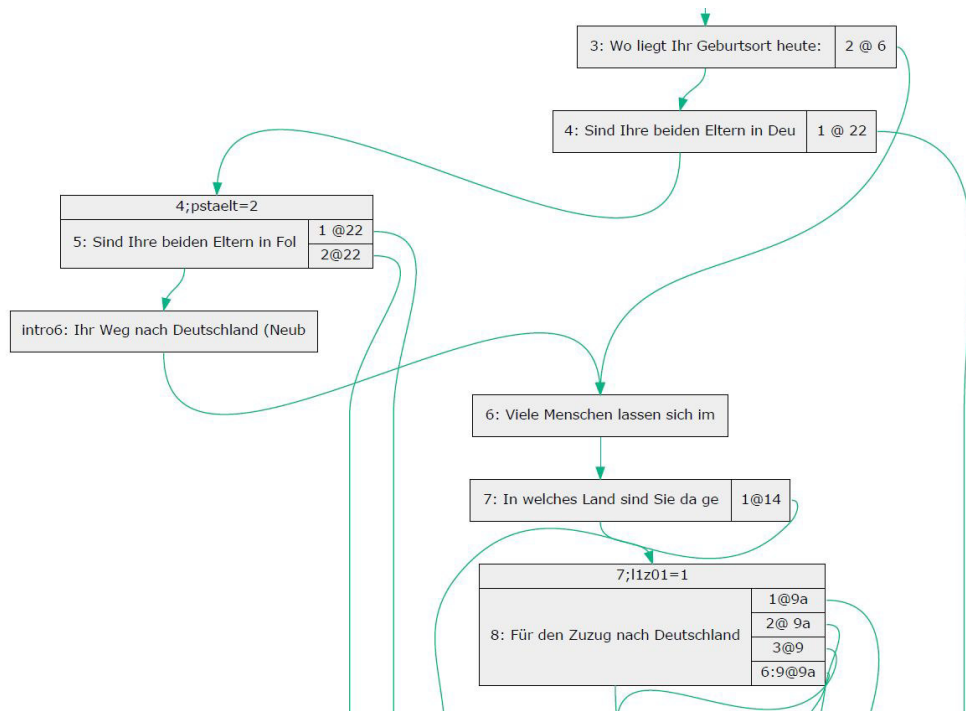
Two different approaches in instruments:

- Question's gatekeeper ("filter")
  - Defines the universe of this particular question
  - Condition which has to be true
- After a question ("goto")
  - Defines the way to the next question depending on the answer (and perhaps other information)



- Which approach is used in your institution? What are your experiences?
- What do data users like, what survey designers – and why?
- Which approach is more, which is less parsimonious?
- What about visualization?
- Will it convert?

## Example: Visualize routing



- Flow chart, algorithmic derived from DDionRails metadata
- Filters displayed
- Gotos parsed
- Layout/rendering by Graphviz

How is filter/goto-approach connected with visualisation?

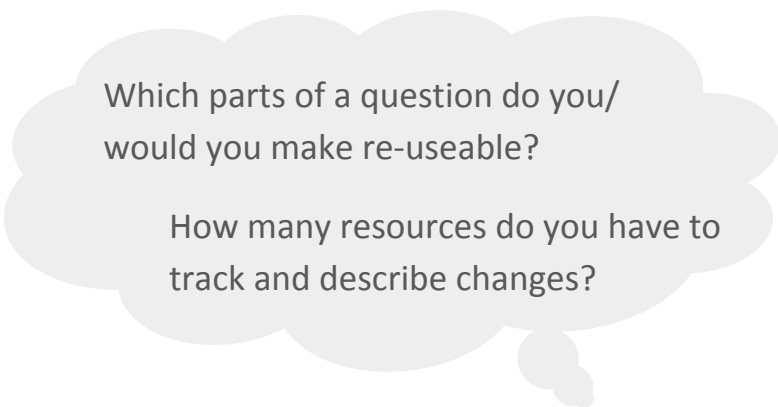
## Make information re-usable and deal with changes

### Re-use

- Means: Combine parts of a question and give them an identifier, which has to be used if the question appears again.
- Tracks permanence.
- Helps to limit amount of information, which has to be managed (entered, translated).
- Makes things more complicated: one more relation.
- Agency needed: assign IDs, ensure integrity, supervise corrections (internal question bank)

### Link over time

- Same methods like those presented for variables
- Comparison seems to be more appropriate



Which parts of a question do you/  
would you make re-useable?

How many resources do you have to  
track and describe changes?