

Rhetorical Analysis Memo

To: Ana Leigh Cooke

From: Team Git

May 5, 2017

Professor Cooke,

The purpose of this memo is to identify, articulate, and justify our visual and verbal strategies while writing and revising our project 4.

Rhetorical Situation

Git is notoriously complicated. Our instruction manual has a git novice as its audience; perhaps a freshman CS major at CMU who is tasked with quickly learning git at a hackathon. Git's online documentation, while thorough, is not written with the novice in mind. This project aims to fill that gap.

To this end, we structured our document as an easily-scannable reading-to-do manual. Readers are met with a “how to use this document” disclaimer at the beginning, explaining that they should feel free to skip around the document as needed. We employ explicit *problem and solution headers* to help the reader quickly jump to the salient information. These headers are given ample surrounding whitespace for emphasis and clarity.

Visual Strategies

We employed a number of visual strategies to aid reading.

For the section on installing Git, we included screenshots of the different steps of the installation process so that users are not lost while going through the steps to install Git on the computer.

We chose to make inline code and block-style code chunks both blue, in order to clearly distinguish between code and prose. We chose to use the monospace font **Computer Modern Typewriter** for code sections.

Furthermore, we found that in practice explaining Git's version history through words alone can often confuse listeners. Thus, to better explain the changes to Git's version history when rebasing and cherry-picking, we incorporated the use of diagrams with individual nodes representing commits. These provided a visual representation of changes to Git's version history when using these commands.

Verbal Strategies

Facilitating Random-Access-Reading via Sectioning

After our first draft, we changed our document to be organization into very specific, well defined sections. The main reason for doing so was that given our document is quite long and not necessarily meant to be read all at once, a user should be able to consult the table of contents to address a specific problem and find the corresponding section. Whereas before we had a section on Branching, now we have sections on Making Branches, Deleting Branches, etc.

Adapting to Beginner and Intermediate Audiences

The guide is written for introductory and intermediate Git users. It is possible that our introductory users have never used Git before. Therefore, we included detailed instructions on how to install and set up Git. The commands introduced in the Getting Started section is organized in an order of how introductory users should use them. In each section, we give clear instructions on what to type in terminal (command line) and what message Git will return to help users cross validate. For intermediate users, they may want to skip the first half of the guide. The table of contents at the beginning of the guide allows them to directly access the contents that they are interested in. Since the guide is only for introductory and intermediate Git users, we did not include everything that Git offers. In addition, for most commands that we include, we avoid talking about all possible flags to keep the guide concise. Although the guide does not include every possible command, it is more than enough for basic version control. One place we imagined it will be helpful is at a hackathon, with the guide providing sufficient information to facilitate collaboration with Git for hackathon par-

ticipants. Finally, we conclude with intermediate to advanced use cases of Git. We make it clear that these operations aren't necessary to learn but we reserved a section for them at the end.

Separating Reading-To-Do from Reading-To-Learn

Another strategy we incorporated was the use of the Problem, Solution, and Discussion subsections. Our original document faced issues where readers would have to read the entirety of a section in order to understand what was required to accomplish their required task. This did not facilitate the random-access-reading that we envisioned for this manual. Furthermore, our document often combined both reading-to-do and reading-to-learn in this manner. As such, the choice to break down each section into these three subsections separated reading-to-do to the Problem and Solution sections while reading-to-learn was left to the Discussion section. Thus, if a reader wished to quickly look up a command for a particular situation, they would simply find the respective section, check if the problem description matches their situation, and apply the solution. If they wished to get a better understanding of the command, the Discussion section would go into more detail on the under-workings of the command.

Choice of Medium

Our manual is intended to be used both by beginners who are learning how to use Git and by experienced Git users who want to look up how to use a specific command. For this reason we decided to make the manual as an online pdf to facilitate easy lookup of commands and random access reading. The table of contents page in the pdf contains links to access a specific section or subsection quickly. Making the manual online also allows the user to check out the manual whenever they need to use Git on their computer and they don't need to carry the manual around with them.