Rhetorical Analysis Memo
To: Ana Leighe 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.

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 Skill Levels via Document Ordering

Now in terms of order, we had an Introduction that introduced the document as well as Git. This section is critical to users with little familiarity with the topic. Our user tests showed us that inexperienced users were confused by the documents and instructions. Then we proceed to installation which covers all possible platforms. With that, we move on to describe the various procedures in Git and they are presented in an order that they will be used in (initializing a repository, making changes, committing the changes). 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.