

iiiiii Updated upstream =====  
lllllll Stashed changes

Git CLI or GitHub Desktop

What are the factors that influence the choice of the  
interface among IT students?

Danil Burov and Giulio Raffaeli

December 18th 2023

Module: ARDA

Venlo, Limburg, Netherlands

## **Abstract**

When it comes to learning programming one of the main aspects an IT student needs to learn is how to version control (VS) their work. The most common tool for versioning is Git. In the beginning IT students need to choose between Command Line Interface (CLI) and Graphical User Interface (GUI) for Git usage.

This purpose of this research is to identify what are the main factors that influences the choice of the interface among IT students. In order to evaluate the reasons, a survey was conducted. Since the most efficient way of using Git is through the CLI an experiment was done to understand why students adhere to the GUI instead switching to the more efficient interface.

The results show that the majority of IT students prefer using Git through the GUI, finding it more visual appealing and easier to understand. Even though, students that CLI is more efficient using Git they still use the GUI because making the transition is harder than sticking to the interface. This statement is proven with our experiment.

# Contents

1	Introduction . . . . .	1
1.1	Context and Background . . . . .	1
1.1.1	Git . . . . .	1
1.1.2	Command line Interface and Graphical User Interface . . . .	1
1.1.3	Command line Interface and Graphical User Interface . . . .	2
1.2	Research questions and Hypothesis . . . . .	3
2	Methodology . . . . .	4
2.1	Gathering data . . . . .	4
2.1.1	Survey . . . . .	4
2.1.2	Experiment . . . . .	5
2.2	Data visualization and transformation . . . . .	6
3	Results . . . . .	6
3.1	Survey participant information . . . . .	6
3.2	Reasons to choose each interface . . . . .	8
3.3	External factors that influence the choosing . . . . .	10
3.4	Efficient usage of Git based on the interface . . . . .	12
3.5	Words association for the two interfaces . . . . .	15
3.6	Experiment results . . . . .	19
4	Discussion . . . . .	21
4.1	Evaluate Research Question . . . . .	21
4.2	Evaluate secondary research question . . . . .	21
4.3	Limitations . . . . .	22
4.4	Further Research . . . . .	23
5	Appendix . . . . .	24
5.1	Survey questions and possible answers . . . . .	24
6	Acknowledgment . . . . .	26

## List of Figures

1	Programming years . . . . .	7
2	Git years of experience . . . . .	7
3	Preferred interface . . . . .	8
4	Reasons why students choose GUI . . . . .	9
5	Reasons why students choose CLI . . . . .	9
6	Operational system affecting the chosen interface . . . . .	10
7	How much time do the students spent on Git . . . . .	11
8	Impact of Git years on the interface . . . . .	11
9	How many people want or have swapped from CLI to GUI . . . . .	12
10	How many people want or have swapperd from GUI to CLI . . . . .	13
11	People who have switched (Efficiency) . . . . .	13
12	People who haven't switched (Efficiency) . . . . .	14
13	If people think that their interface is the fastest (Switch) . . . . .	14
14	If people think that their interface is the fastest (No switch) . . . . .	15
15	What does a CLI user associate the CLI with . . . . .	16
16	GUI user words for CLI . . . . .	16
17	What does a GUI user associate GUI with . . . . .	17
18	CLI user words for GUI . . . . .	18
19	Average time for completing the experiment . . . . .	19
20	Experiment completed through the command line interface . . . . .	20
21	Experiment completed through the graphical user interface . . . . .	20

# 1 Introduction

The purpose of this chapter is to provide the research question and the importance of it, along with context about the topic. Furthermore, this chapter will point out potential external factors that may influence the evaluation of the hypothesis and its sub-questions.

## 1.1 Context and Background

### 1.1.1 Git

Updated upstream Up until this date Git is the most popular versioning control system, used by more than 93 percent of developers ('cite' -> StackOverflow survey). In order to understand Git it is crucial to know what is a versioning control system is (VCS). It is a system that enables developers to keep historical version of source code(\*) and project files that are under development and retrieve past version. It is required when developing projects above a few hundred lines of code or where more than one developer needs to collaborate on a project. It stores version information for every file in what is generically called - 'repository'('cite -> pdf history of version control') . The basic structure of a Git repository has three main components. First, a .git directory which has the functionality of storing the meta data and object database for you repository and all the committed changes. Second, a staging area where new features that still need to be committed are kept, waiting for a commit to take place. Lastly, a working directory where your plain running copy of the source code is held. Once a commit takes place, the changes are saved from the staging area to the .git directory. ('cite -> pdf for CLI compare to GUI').

### 1.1.2 Command line Interface and Graphical User Interface

When the first personal computer was invented in 1973 (Kenbak - 1) ('cite' -> Museum computerhistory.org) users were forced to use the command line interface as their only way of interacting with the machine. Later on, with more people using the personal computers the need for a graphical user interface increased. The first prototypes of personal computers that use the GUI were developed in the 1970s (XEROX Alto 1973) ('cite' -> wikipedia), however, it became popular with the release of the 'Macintosh' in 1984. ('cite' -> wikipedia ). From that point on, the majority of users started using the GUI as their main interface relegating the use of CLI only to a small percentage of users.

When Git was first released in 2005 ('cite' -> Gitpage) the only way to work with it was through the command line interface. Throughout the years many graphical user interfaces supporting Git were developed with the most important one being GitHub Desktop, created in 2017. ('cite' -> GitHub page)

When using the command line interface you type commands manually to perform the desired actions whilst in a graphical user interface you will have something visual to interact with,

such as buttons, input fields and so on. There are advantages and disadvantages of using either of the two interfaces not only for Git but in general. ('cite' -¿ pdf for CLI vs GUI). When it comes to programming, 83('cite' -¿ StackOverflow survey) percent of developers prefer the CLI instead of the GUI for various of reasons. Here are some of them:

1) Interaction speed, when it comes to GUI the speed is determined mostly on how fast you can navigate with your mouse and click speed while with the CLI what you need to do is just interact with the keypad making the process faster.

2) Efficiency, because of the way interaction is handled in the GUI it also requires more actions compared to the CLI to execute the same task. For example: If you want to commit changes to the main repository with the CLI you need to execute three consecutive commands, whereas in the GUI you need to navigate through three different buttons and you have to input a message with your keyboard.

3) One complaint about CLI tools is that they supposedly have a 'steeper learning curve' and no one can remember what commands to use. However, if you need to learn a complex GUI the 'learning curve' can be as steep as learning the CLI. Although, in the long run it is always better to learn the CLI because more efficient. ('cite' -¿ github repo) ===== Up until this date Git is the most popular versioning control system, used by more than 93 percent of developers (StackOverflow 2022, 2020). In order to understand Git it is crucial to know what is a versioning control system is (VCS). It is a system that enables developers to keep historical version of source code and project files that are under development and retrieve past version. It is required when developing projects above a few hundred lines of code or where more than one developer needs to collaborate on a project. It stores version information for every file in what is generically called - 'repository' (Ruparelia 2010) . The basic structure of a Git repository has three main components. First, a .git directory which has the functionality of storing the meta data and object database for you repository and all the committed changes. Second, a staging area where new features that still need to be committed are kept, waiting for a commit to take place. Lastly, a working directory where your plain running copy of the source code is held. Once a commit takes place, the changes are saved from the staging area to the .git directory. (Hultstrand and Olofsson 2015)

### 1.1.3 Command line Interface and Graphical User Interface

When the first personal computer was invented by XEROX (Smith and Alexander 1999) users were forced to use the command line interface as their only way of interacting with the machine. Later on, with more people using the personal computers the need for a graphical user interface increased. The first prototypes of personal computers that use the GUI were developed in the 1970s (XEROX Alto 1973) (Reimer 2005), however, it became popular with the release of the 'Macintosh' in 1984(Reimer 2005). From that point on, the majority of users started using the GUI as their main interface relegating the use of CLI only to a small percentage of users.

When Git was first released in 2005 (Straub 2014) the only way to work with it was through

the command line interface. Throughout the years many graphical user interfaces supporting Git were developed with the most important one being GitHub Desktop, created in 2017. (Forster 2017)

When using the command line interface you type commands manually to perform the desired actions whilst in a graphical user interface you will have something visual to interact with, such as buttons, input fields and so on. There are advantages and disadvantages of using either of the two interfaces not only for Git but in general. (Hultstrand and Olofsson 2015). When it comes to programming, 83(StackOverflow 2022) percent of developers prefer the CLI instead of the GUI for various of reasons. Here are some of them:

- 1) Interaction speed, when it comes to GUI the speed is determined mostly on how fast you can navigate with your mouse and click speed while with the CLI what you need to do is just interact with the keypad making the process faster.
- 2) Efficiency, because of the way interaction is handled in the GUI it also requires more actions compared to the CLI to execute the same task. For example: If you want to commit changes to the main repository with the CLI you need to execute three consecutive commands, whereas in the GUI you need to navigate through three different buttons and you have to input a message with your keyboard.
- 3) One complaint about CLI tools is that they supposedly have a 'steeper learning curve' and no one can remember what commands to use. However, if you need to learn a complex GUI the 'learning curve' can be as steep as learning the CLI. Although, in the long run it is always better to learn the CLI because more efficient. (Bagley 2020) `git stash` Stashed changes

## 1.2 Research questions and Hypothesis

Updated upstream The aim of the research is to find what are the reasons that influence the choice of the interface among IT students. Therefore our research questions is: "What are the factors that influence the choice of the interface among IT students?". Since surveys show that command line interface is more used ('cite' -> StackOverflow ) than graphical user interface ('cite' -> GitHub repo) and also more efficient the sub-question this research paper is aiming to answer is the following: "Is transitioning to the CLI hard for an IT student that still uses the GUI?". ===== The aim of the research is to find what are the reasons that influence the choice of the interface among IT students. Therefore our research questions is: "What are the factors that influence the choice of the interface among IT students?". Since surveys show that command line interface is more used (StackOverflow 2022 ) than graphical user interface (Bagley 2020) and also more efficient the sub-question this research paper is aiming to answer is the following: "Is transitioning to the CLI hard for an IT student that still uses the GUI?". `git stash` Stashed changes

We hypothesis that students do not base their choice of Git interface on efficiency but on how user-friendly and simple the interface is. Therefore, we also believe that making



the transition to the CLI from GUI is hard, because of the steep learning curve from the beginning and that is why IT students keep using the GUI.

Although, there are some external factors that need to be taken into consideration.

- 1) The years a student has already programmed for.
- 2) How long have they been working with Git.
- 3) What was their first chosen interface.
- 4) Which operating system do they use.

In order to take into consideration the given factors a survey was conducted to evaluate what are the aspects that influence the choice of interface for an IT student. Parallel to the survey, an experiment was conducted with students that have never used the CLI before in order to understand what difficulties they will encounter when learning basic commands and if it makes them faster at using Git.

## 2 Methodology

This chapter will explain how we performed our research and how we set up our comparative analysis.

### 2.1 Gathering data

Since our research aims to find on what do IT students base their choice of interface we had to gather all the data mainly by ourselves and some external sources. This data has been gathered through the survey and experiment that was conducted among them.

#### 2.1.1 Survey

In order to answer our main question, while also keeping in consideration all the external variables that may influence the answer, a survey was conducted. Since the research aims to give an answer that represents all IT students, the survey was sent not only to Fontys learners, but also to students from other universities. The survey itself consists of 14 multiple choice questions, plus an optional open question. It is structured as follows:

First of all general questions are asked, regarding how long has the participant programmed for, how long has he been using Git and what is his current knowledge of the tool. Next questions are related to which Operating System does the responder primarily use and how did he get into Git at first (from which sources did he learn to use it and with which interface). Afterwards, information regarding which interface does the participant currently use and why is gathered. Subsequently, the respondent needs to associate various words to both interfaces, in order to understand what is his opinion on them. Lastly, questions about whether the candidate has ever considered switching interface and if he thinks that doing so would improve his efficiency are asked. In order to easily convert all the answers into data, the survey was created with the tool Google Forms (Google - 2008). The list of all the questions and the possible replies can be found in the appendix.

### 2.1.2 Experiment

The experiment was conducted among second year IT students from Fontys. Given this information we chose the GUI to be GitHub Desktop since it is the most used graphical user interface.

Updated upstream The main purpose for the experiment conducted for our research paper is to prove that when a person wants to switch from using the graphical user interface (GitHub Desktop) for Git to the command line interface the transition will take time and effort when it comes down to getting used to the commands. Especially because as mentioned above the learning curve is very steep. (cite -i GitHub repo).

The exercise given to the students represent a very simple scenario where a developer would need to use Git through their interface. For the sake of our paper the experiment was done through both the graphical user interface and the command line. For a GUI we chose to use 'GitHub Desktop' since it is the most commonly used graphical interface (cite -i find somewhere to cite from this quote). The purpose of doing the assignment in both the interfaces is to prove that using the CLI for Git is much slower for students who have never used it and much more confusing. However, when a student had already used the CLI for Git purposes it was seen that the issues the newcomers had (mainly commands) weren't in the picture.

===== The main purpose for the experiment conducted for our research paper is to prove that when a person wants to switch from using the graphical user interface (GitHub Desktop) for Git to the command line interface the transition will take time and effort when it comes down to getting used to the commands. Especially because as mentioned above the learning curve is very steep. (Bagley 2020).

The exercise given to the students represent a very simple scenario where a developer would need to use Git through their interface. For the sake of our paper the experiment was done through both the graphical user interface and the command line. For a GUI we chose to use 'GitHub Desktop' since both of us have previously used it and we were confident creating an assignment with this graphical interface. The purpose of doing the assignment in both the interfaces is to prove that using the CLI for Git is much slower for students who have never used it and much more confusing. However, when a student had already used the CLI for Git purposes it was seen that the issues the newcomers had (mainly commands) weren't in the picture.

Stashed changes

The assignment is covering all the basic commands that need to be done in order to complete a commit changes in a repository. Both groups of IT students have never seen the assignment and were treated as they have never used Git through the CLI. They were given two sheets of paper and a repository containing the assignment they need to complete. The first paper, contained all the commands required to complete the assignment ('refer to a figure'). The second paper, had the description of the assignment. ('refer to a figure here'). Time was recorder during the conducting of the exercises.

## 2.2 Data visualization and transformation

After gathering all the data, in order to store it in an organized way, multiple sheets were created. The tool used to create them was 'Google Sheets'. After inserting the data gathered from the survey it was exported as a 'CSV' file in order to use it later on for the data transformation. The same process was executed for the experiment.

To perform the data transformation we had to choose between 'Python' and 'R'. We chose to use 'Python' with the library "pandas" v2.1.4 (McKinney, 2023) because we had previous knowledge of the language.

First of all, we had to convert our 'CSV' files into dataframes which can then be used to generate plots with the use of the library "plotnine" v0.10.4 (Kibirige, 2023). With this library multiple type of charts were created based on the variables that needed to be visualized or compared. One of the created diagram is the histogram. It was used to visualize single variable plots. The second type of diagram is the bar chart. We used it to display comparison between different entries of the same variable. And lastly, bar charts with multiple variables were used to point out correlations within two parameters.

Before creating the plots, we needed to refine our dataframe. Initially, we had to remove a column named 'Timestamp' which was generated automatically while converting the survey data from 'Google Forms'. Later on, to perform specific queries we had to filter the data based on particular factors which influenced greatly the outcome of our plots. Moreover, to count how many times each word was chosen we needed to separate group of words and evaluate each one individually. Lastly, for the experiment dataframe we had to calculate the average time of completion for both interfaces to have a clear view of the time difference

## 3 Results

In this section we are going to display and explain each diagram that was generated from the results of the survey and the experiment. Every diagram will have a short explanation why it was created and the impact it has for our research. The survey had 60 participants while the experiment had 10 candidates.

### 3.1 Survey participant information

In this subsections plots containing general programming expertise of the participants are going to be displayed, as well as their git experience and the interface they commonly use.

As we can see from the figure below, among all the participants the majority of them (25) has been programming for 1-2 years, followed by 5+ years experience (24). Only, a small group of students has been programming for 2-5 years (9). (Figure.1)

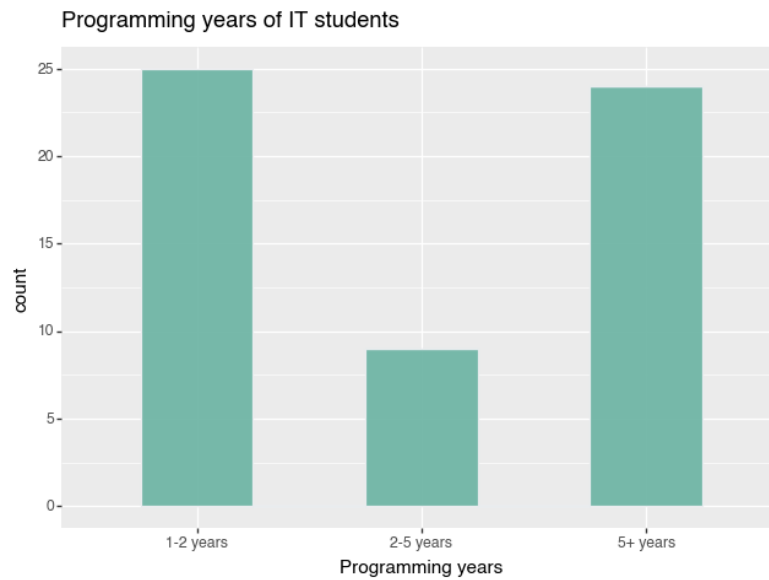


Figure 1: Programming years

The diagram shows that 31 participants have been using Git for 1-2 years. 19 participants between 3-5 years and only 9 students have answered more than 5 years. (Figure. 2)

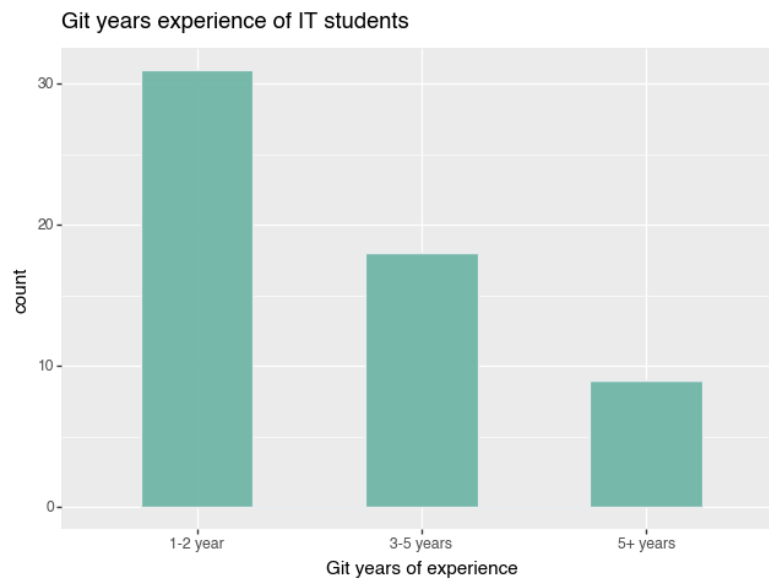


Figure 2: Git years of experience

In the figure below, the results of which is the most preferred interface is shown. From

the diagram (Figure. 3) it is very clear that most of the participants of our survey use the graphical user interface instead of the command line.

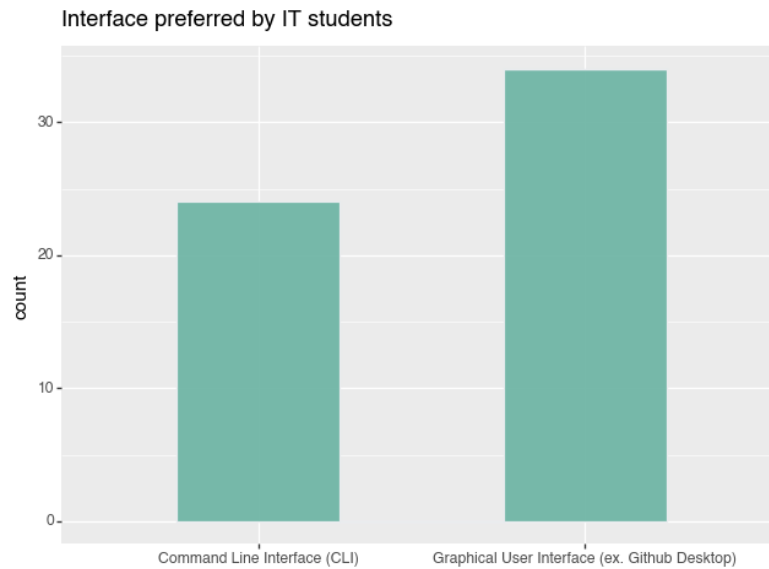


Figure 3: Preferred interface

## 3.2 Reasons to choose each interface

The plots displayed in this subsection will display the factors that influenced the student to choose the interface they are currently using. They had to select from a range of words that represent why they opted for the interface they are currently using. We have two plots which show the answers from graphical user interface users and command line interface users.

The following figure (figure.4) , presents the response of the IT students using the GUI as their main interface for usage with Git. It specifically highlights the words they associate with the GUI.

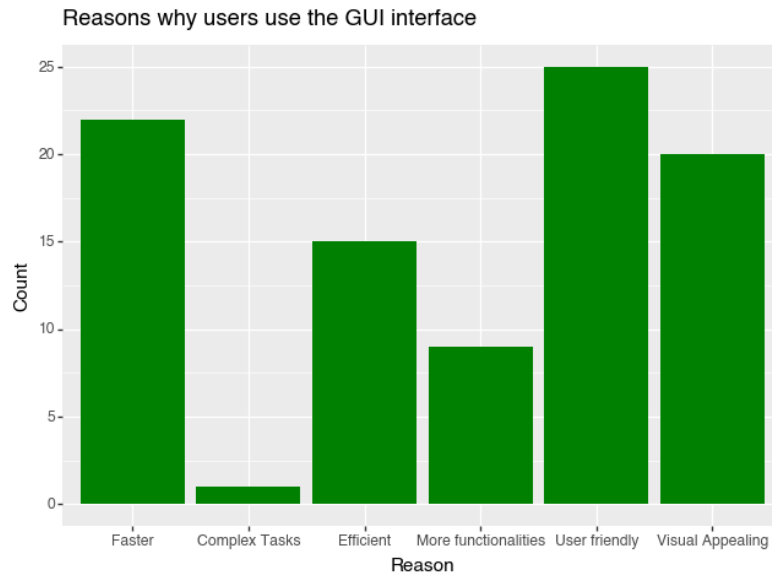


Figure 4: Reasons why students choose GUI

The figure below (figure.5) showcases the feedback from IT students who predominantly use the command line interface (CLI) for Git operations. It focuses mainly on the words the students used to associate their Git usage via the CLI.

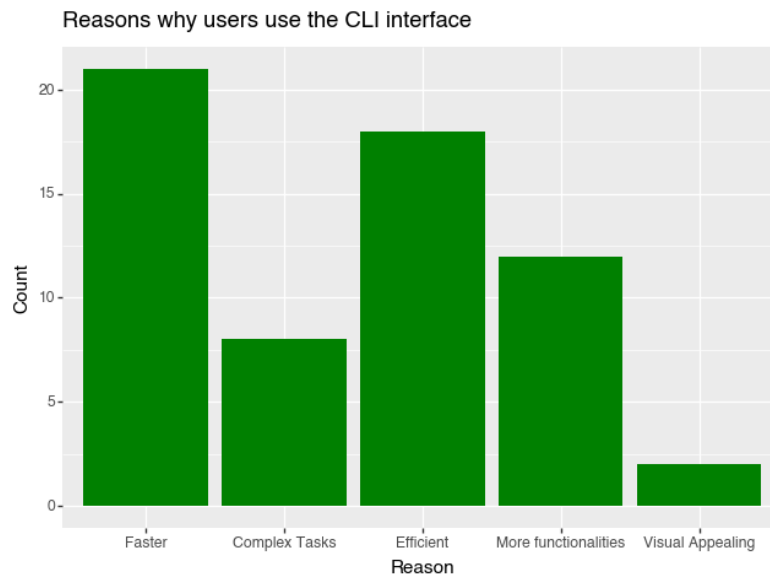


Figure 5: Reasons why students choose CLI

### 3.3 External factors that influence the choosing

Here we have derived all the external factors that influences the choice of the IT student's interface. We evaluated if the time spent in Git, the operational system that the student uses and years of Git experience, influence the interface of their choosing.

In Figure.6 it is shown that the majority of the IT students who did the survey use the operational system 'Windows'. Whereas the 'Mac' and 'Linux' OS is not that common among the students. Most of the people who use 'Windows' use the graphical user interface to operate with Git. And we can see that the other two operational system have very similar results when it comes down to choosing their interface for Git usage.

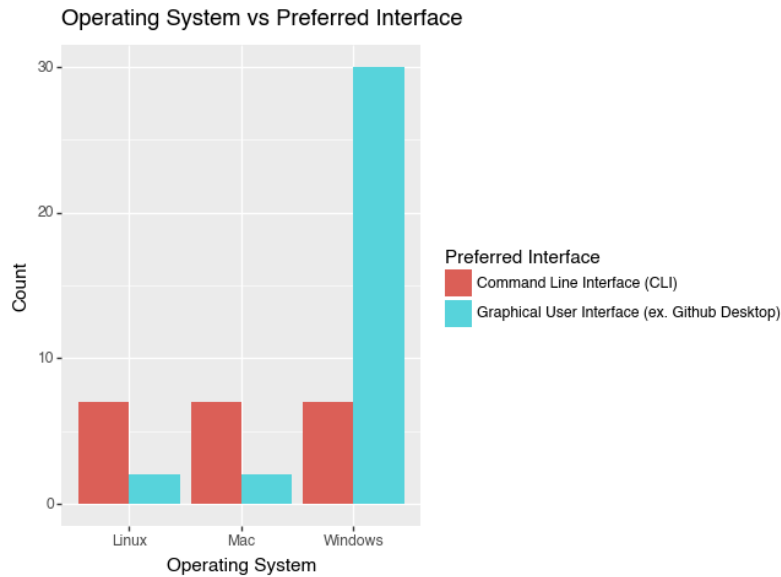


Figure 6: Operational system affecting the chosen interface

The time that a student spends in Git is crucial in choosing the interface. In figure.7 we have displayed the times that IT students spent in Git, executing commands or in general working with Git in one hour.

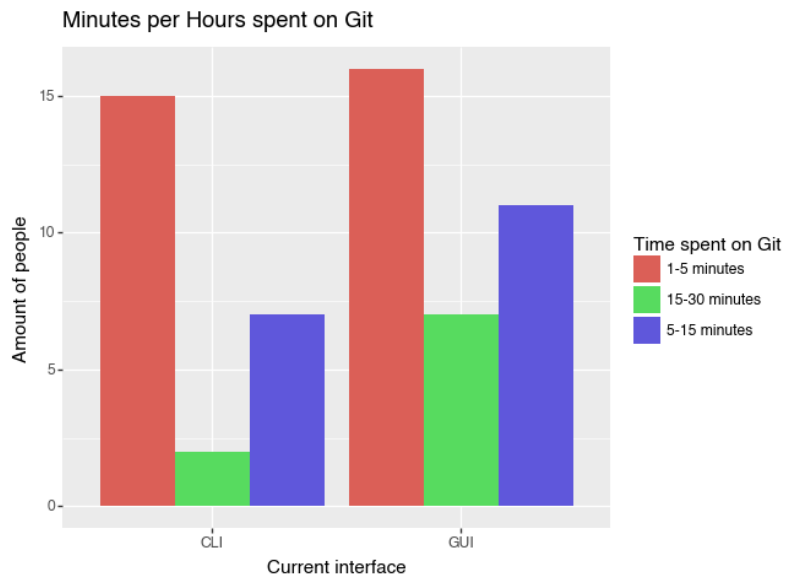


Figure 7: How much time do the students spent on Git

The following factor is answering if whether the years a student has been using Git has any sort of impact of their choosing of the interface they are currently using.

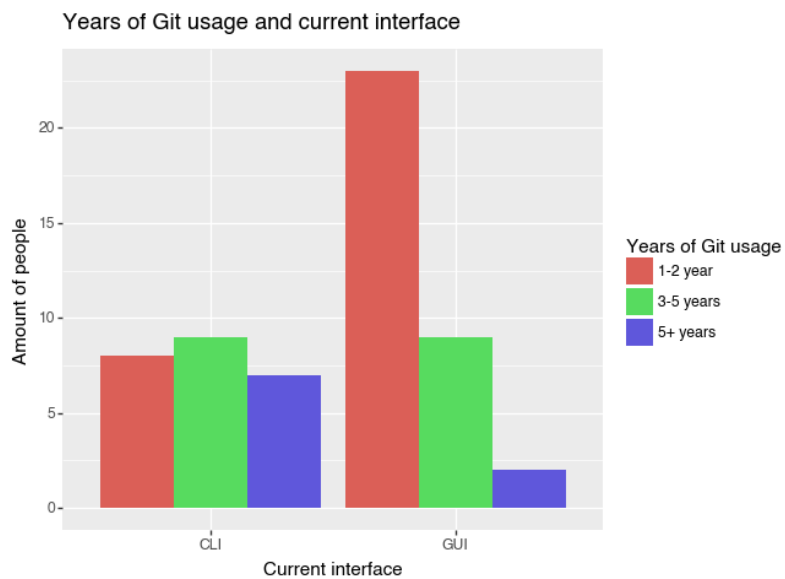


Figure 8: Impact of Git years on the interface



### 3.4 Efficient usage of Git based on the interface

This section is going to display the results about the reasons why a user would like to switch its interface. First of all, we are going to analyze, based on the current interface they use, whether they ever consider swapping to the other interface at one point of their programming career. Then, based on the question above, we are going to see if the participants believe that switching would improve their speed and efficiency, or if they stick to the current one because swapping would slow their process.

From the first plot (Figure 9) we notice that the majority of CLI users (14 candidates) never thought of switching, while 10 of them wanted to use the other interface at least once

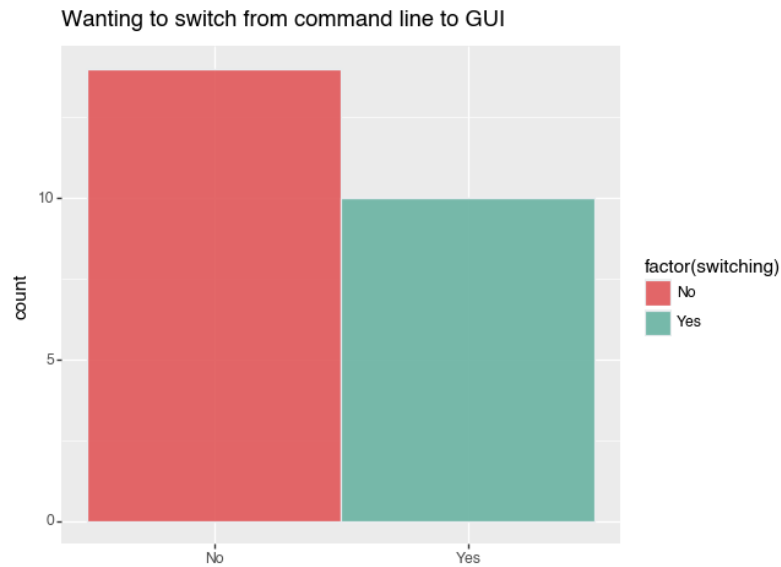


Figure 9: How many people want or have swapped from CLI to GUI

On the GUI side, 20 people stated that they never thought of swapping to CLI, while 14 candidates at least had this idea once. From both images (Figure 9 and Figure 10), we can see that the ratio is almost similar regardless of the interface

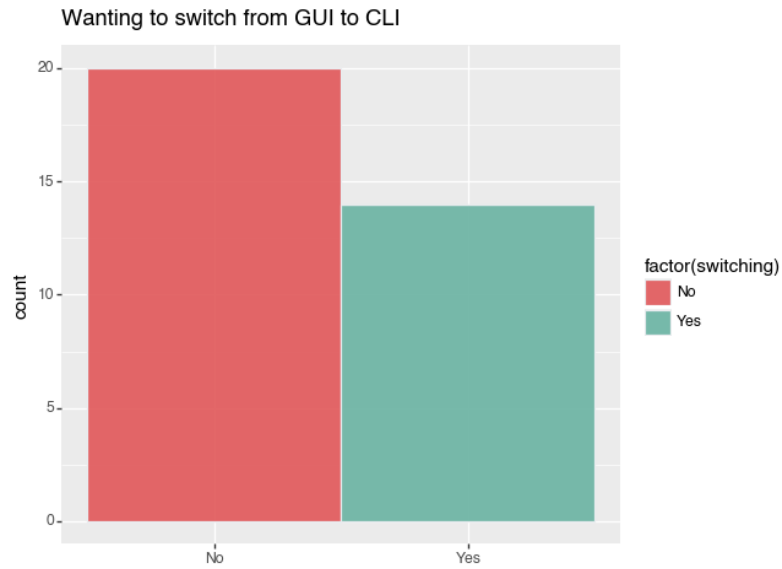


Figure 10: How many people want or have swapperd from GUI to CLI

The following diagram (Figure 11) is used to display whether people that considered swapping believe that it will make them faster. From the result, all the candidates that answered (24), are convinced that swapping would not improve their speed at all.

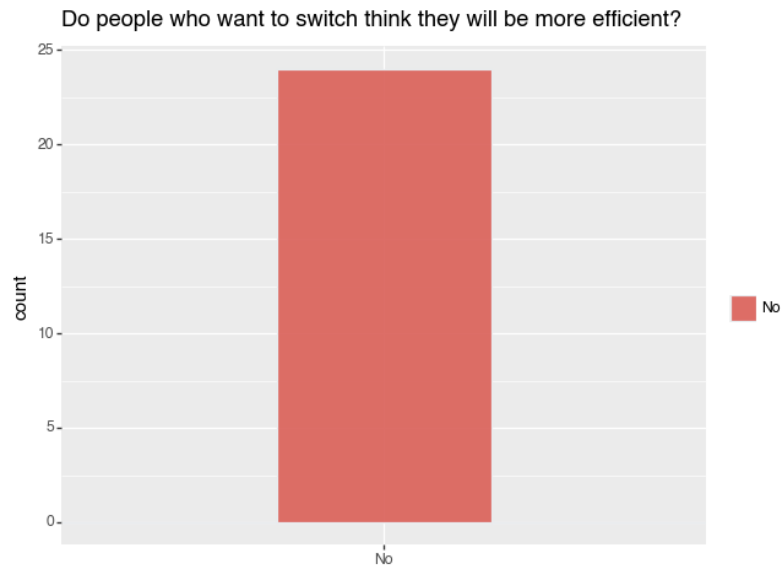


Figure 11: People who have switched (Efficiency)

Confirming this opinion, people that never considered switching believe that they are using

the faster method, while a small percentage (6 candidates) are doubtful about this statement

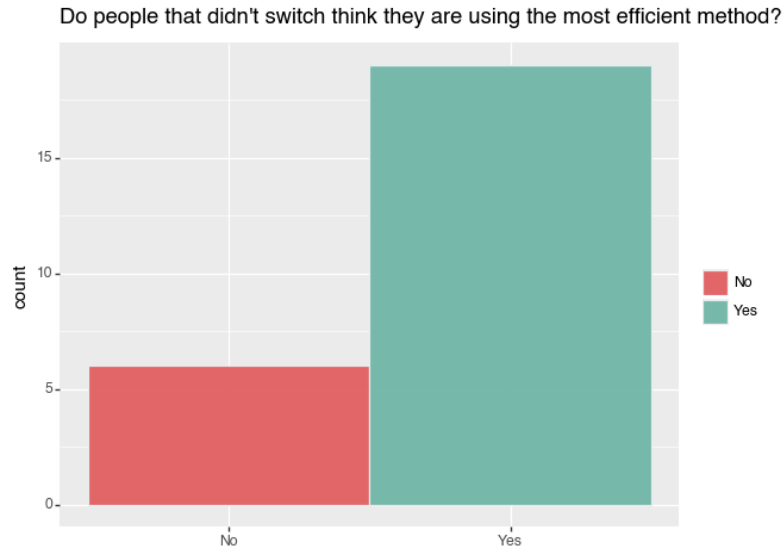


Figure 12: People who haven't switched (Efficiency)

Following this trend, when people that want to switch are asked whether they will be faster with their new interface, only 3 candidates believe that this is going to be the case, leaving the 21 remaining sceptical, as we can see from the diagrams below (Figure 13, Figure 14)

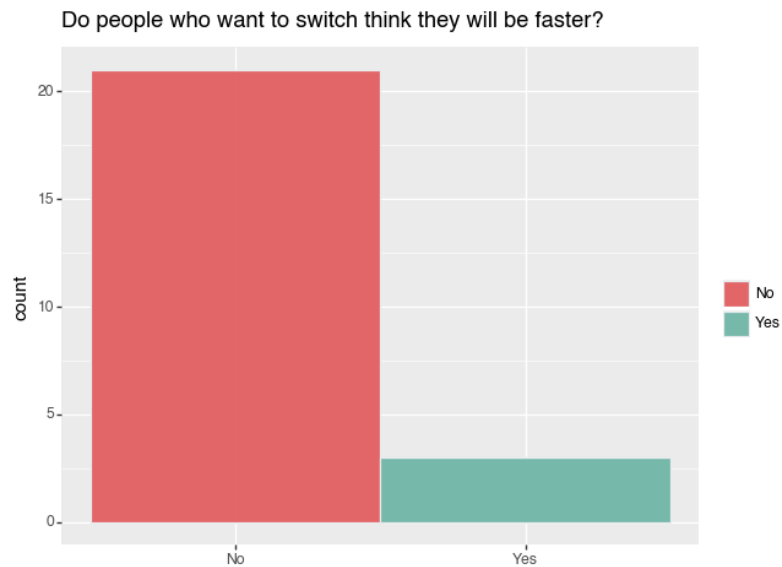


Figure 13: If people think that their interface is the fastest (Switch)

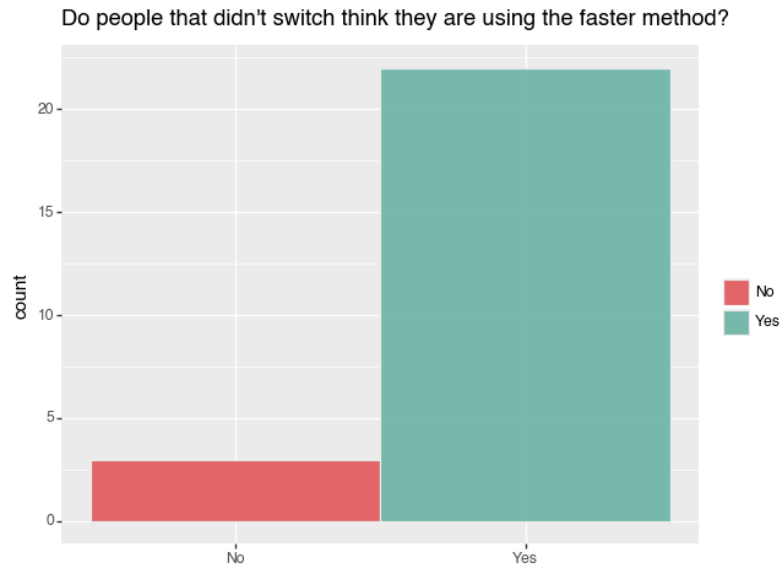


Figure 14: If people think that their interface is the fastest (No switch)

### 3.5 Words association for the two interfaces

In this sections we are going to have a deep look at what are the most common words associated with both interfaces. In order to have a bigger picture in the end, we created 4 separated diagrams, the first one displaying what do people that use CLI think about their interface, followed by their opinion on GUI. Next, the results about which are the most common words associated to GUI from its audience and their thought about CLI are going to be presented.

From the diagram below (Figure 15) we can acknowledge that 23 CLI users believe that the Command Line is more controlled, 21 that is more efficient and 15 that it helps to control and manage files better. Some of them also think that it is easy to use and helpful, but at the same time dispersive.

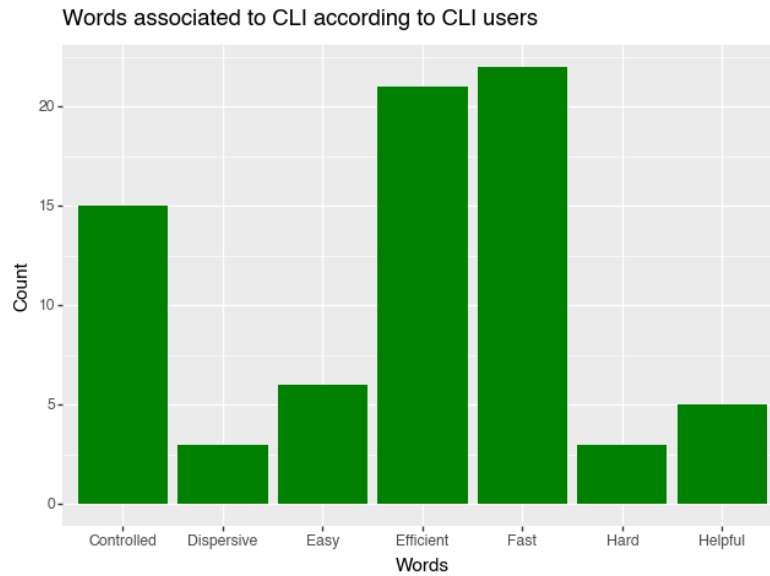


Figure 15: What does a CLI user associate the CLI with

On the other hand, from the following image(Figure 16) it is clear that the majority of CLI users ( 14 candidates) state that using the GUI is easier and helpful, but at the same time slower and more time consuming. Just a few participants believe that using the GUI is efficient, fast and controlled, while others said that is dispersive and hard.

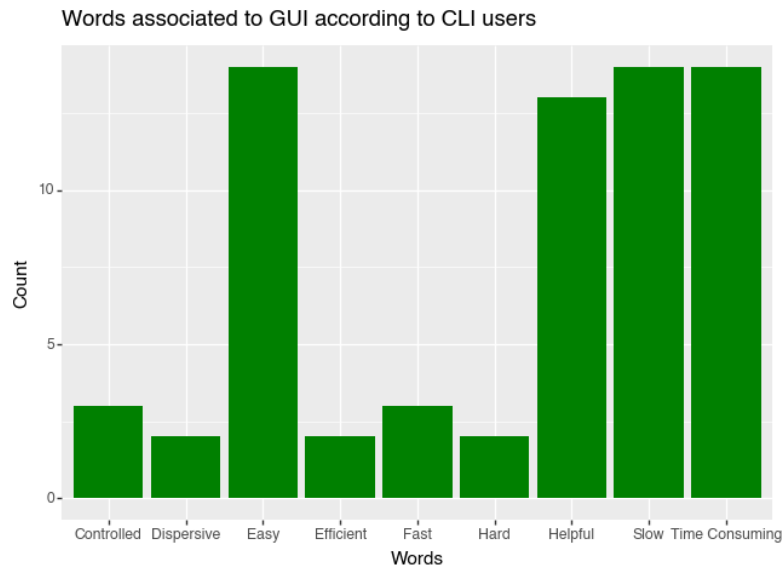


Figure 16: GUI user words for CLI

In the next diagram (Figure 17) we can see that the majority of GUI user's opinion(25 users) on their interface is that it's easy to use and helpful. Another big portion (15 people) believe that it is fast, efficient and controlled, although some of them also think that its is hard, slow, time consuming and dispersive

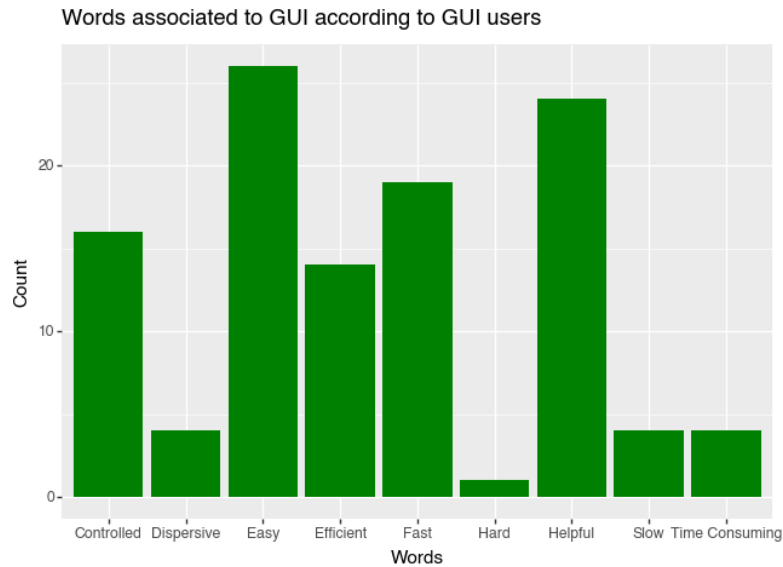


Figure 17: What does a GUI user associate GUI with

In the last image (Figure 18), we can notice that a lot of GUI users (31 candidates), still acknowledge the fact that CLI is more efficient, fast and controlled(14 users). However, they also state that it is hard, time consuming (16 participants) and dispersive. Lastly, a small portion (4 people) claims that this interface is easy and helpful to use.

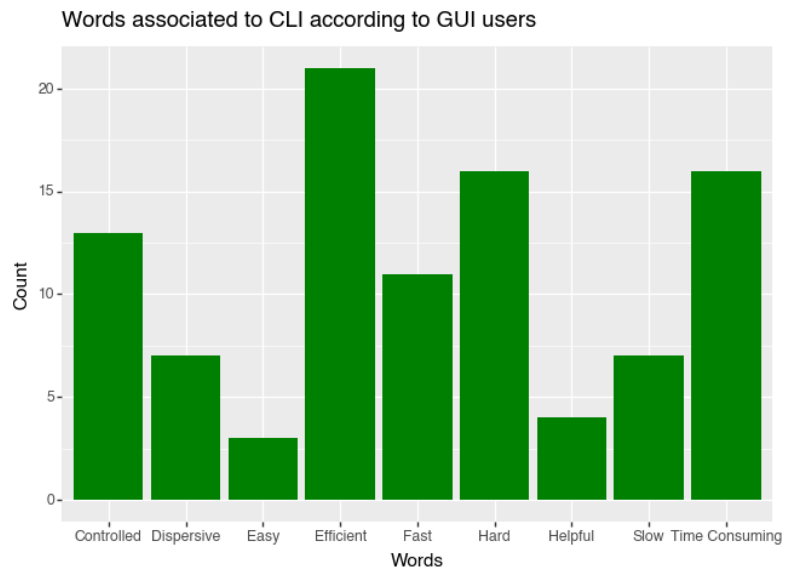


Figure 18: CLI user words for GUI

## 3.6 Experiment results

The data displayed in the subsection will be regarding the experiment that was conducted among the IT students. We derived three plots in total which are most relevant to our research paper. Here we have shown the average time of completion in seconds between the two interfaces, the time it took for each participant in our experiment finishing the assignment through each interface.

To have a better overview and make the calculation of the average of all the students who completed our assignment, we converted the minutes we had recorded to seconds. That way it is easier to see the different times that people using the CLI and GUI had for the completion of the experiment. On x-axis we have the interface that was used for the completion and on the y-axis the average seconds it took the students. (figure.19)

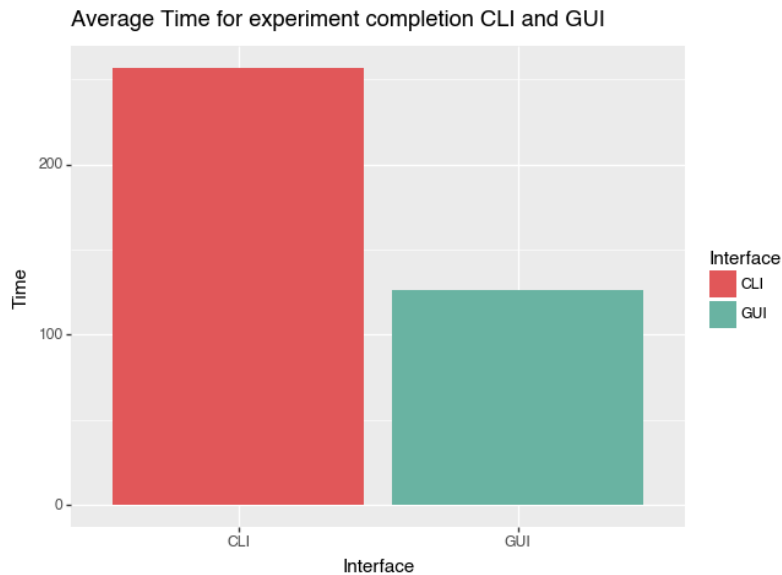


Figure 19: Average time for completing the experiment

In the following figure (figure.20) we have displayed all the completions of our assignment among all the candidates who used the CLI. Here we decided to keep the minutes and not convert to seconds since it has a better overview in this case of the actual times it took people.



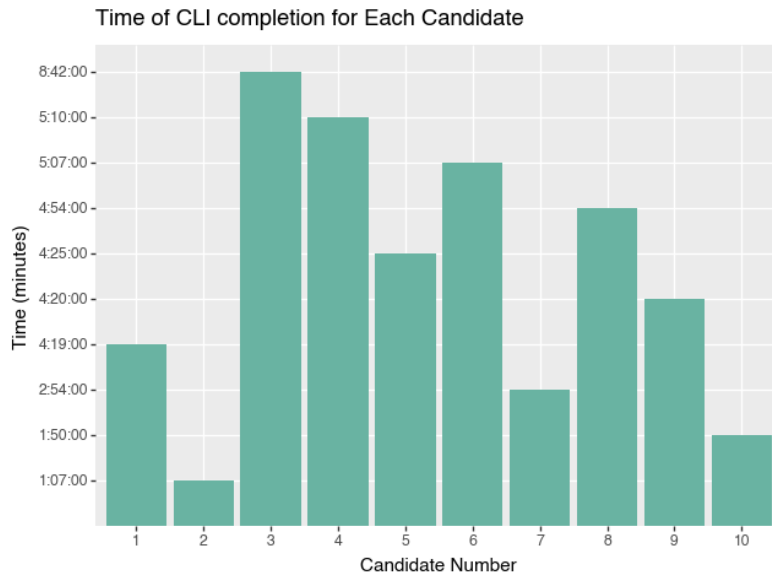


Figure 20: Experiment completed through the command line interface

Below, we have displayed the results of the people who finished the assignment with the GUI. As previously mentioned, all candidates who participated in our experiment had to complete the assignment with both interfaces.

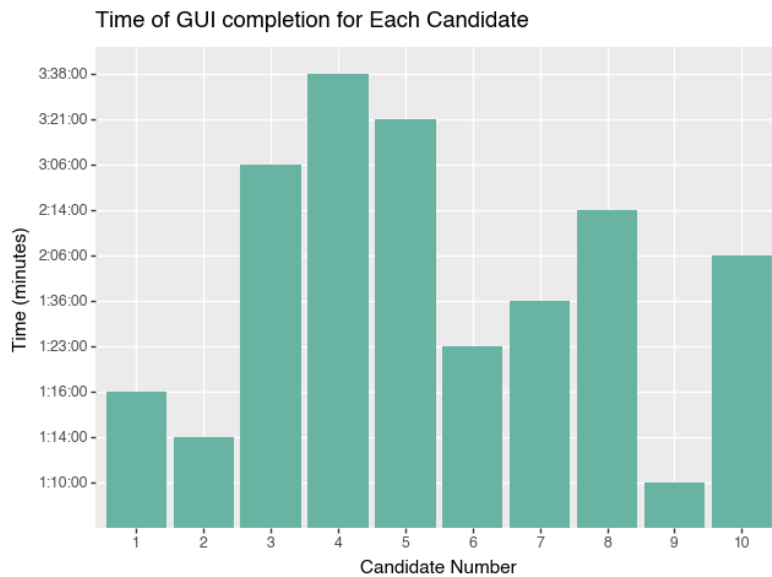


Figure 21: Experiment completed through the graphical user interface

## 4 Discussion

### 4.1 Evaluate Research Question

In order to answer our main research question, we need to analyze the data gathered and transformed from the survey. As mentioned in the introduction, there are multiple factors that may influence the choice of one interface instead of the other, starting from the experience a student already has with the tool. As we can see from Figure 8, the more years of programming someone has using Git, the least likely it is that he is using the GUI. On the other hand, the majority of novices decide to approach the GUI as their first interface. Nonetheless roughly the same amount of candidates with 2 to 5 years of Git experience uses one interface or the other, meaning that the transition to the more efficient CLI usually happens after the student is already confident on how Git works.

Next we have to consider whether the first interface chosen influences the outcome of why a user prefers an interface. As shown in Figure 9 and Figure 10, this doesn't seem to influence the final choice, since almost the same percentage of candidates from both interfaces have considered switching from one method to the other.

The last external factor that may condition an IT student is the main Operating System he usually uses. As shown in figure 6, this is a crucial factor, since both linux and mac users almost entirely rely on the CLI, while the majority of Windows users still prefer the other interface.

When we first started conducting the research, our hypothesis was that students do not choose their interface based on how efficient they are going to be using it, but how intuitive and user friendly it is. From the results of the conducted survey, this seems to be case. First of all, from Figure 4 and figure 17, it is evident that participant that use the GUI, prefer it mostly because of its user-friendliness, for its visual appealing and because it is easy and helpful, even though for the majority is also faster. On the other hand, from figure 5 and figure 15, we can derive that candidates that use CLI are more focused on their efficiency, choosing this interface for its efficiency, its speed and the controlled environment. Moreover, this users are also interested in the more functionalities that the CLI has, speeding even more the development process. Considering that 65 percent of the candidates uses the Graphical user interface (Figure 3), we can clearly state that IT students are more interested in the appearance of the interface instead of its efficiency. Now that the first research question is answered, we can proceed to trying to figure out why IT students that use the GUI do not want to transition to the more efficient Command Line Interface.

### 4.2 Evaluate secondary research question

As stated in the introduction, we believe that users that use the GUI won't change interface because it's not going to improve their efficiency and is not worth the time. This statement

can be confirmed by analyzing the data from both the experiment and the survey. First of all, although some users are aware that they are not using the most efficient and fast method (Figure 12, 14), all the participants believe that switching won't make them more efficient (Figure 11). Moreover, when asked opinions about the CLI, a lot of GUI candidates acknowledged that it is an efficient and more controlled environment, but at the same time hard and time-consuming (Figure 18). Another statistic that confirm that using the CLI is more efficient is the average minutes per hour the survey participants spend on Git (Figure 7). From the diagram is clear that the average time for the CLI candidate is around 5 minutes, while on the GUI side the average time is at around 10/15 minutes, with a lot of participants averaging 15 to 30 minutes per hour using this tool, drastically reducing the development process. Apart from the survey, the best way to evaluate whether our hypothesis is true is to discuss about the experiment's results. From the experiment it is evident that novice GUI users that never touched the CLI are dramatically slower when it comes to execute basic tasks on the CLI compared to their preferred interface. From the 10 candidates that conducted the experiment, the average time for completion of the CLI is more than doubled compared to the GUI (Figure 19). Moreover, while on the GUI side of the experiment there are 5 results above the average and 5 below it (Figure 20), meaning that the average reflect the overall outcome, in the CLI only 3 students completed the assignment in less than the average time.(Figure 21) From the results of the experiment it is evident that trying to transition from the GUI to the CLI is hard, time consuming and in the short run is going to reduce the student's already low efficiency.

## 4.3 Limitations

In this section we are going to discuss about the limitations related to the research methods we have chosen. First of all, for both the survey and the experiment, the sample size cannot perfectly reflect all IT students, since only 60 people participated in the survey and 10 on the experiment. Moreover, even though we tried to send the survey not only to fontys students but also to learners from other universities, the majority of the results came from the first group, making the research more about fontys IT students than the whole category. Talking about the survey, one problem we encountered during data analysis is that we asked whether switching is going to improve their efficiency or not to both GUI and CLI users, changing the result that needed to be readjusted during the data transformation process. Another error was to ask in the survey about how long has the candidate been programming for, since in the final data analysis only the Git experience became an external factor to point out. When it comes to the experiment the only problem that we believe may limit the final result is the structure of the assignment. In fact, although the tasks the candidate had to perform were well documented, some of them were uncertain on what to exactly do, increasing their time of completion. Since half of the candidates started with the GUI and half with the CLI, this shouldn't have influenced the overall outcome of the experiment. Lastly, some of the

candidates already had experience on using the CLI for executing other tasks (for example navigating through folders and editing files), making the performance of the assignment easier and faster

## **4.4 Further Research**

In order to improve the quality of the results of the experiment, we propose to recreate the assignment based on candidate's suggestion and to create a second assignment to analyze their improvement. Moreover, the sample of both the survey and the experiment should be increased and the number of IT students from different universities should be improved.

## 5 Appendix

### 5.1 Survey questions and possible answers

1 - How long have you been programming?

- a) 1-2 years
- b) 2-5 years
- c) 5+ years

2 - How long have you been working with Git?

- a) 1-2 years
- b) 3-5 years
- c) 5+ years

3 - How would you rate your knowledge for Git?

- a) Novice
- b) Average
- c) Expert

4 - How did you learn to use Git?

- a) Self taught
- b) Friends
- c) Online Tutorials
- d) Other

5 - What operating system do you primarily use?

- a) Windows
- b) Mac
- c) Linux
- d) Other

6 - What type of user interface are you primarily using in Git

- a) Command Line interface (CLI)
- b) Graphical User Interface (ex. Github Desktop)

7 - Why do you use this interface?

- a) More efficient
- b) Faster
- c) More functionalities
- d) More visual appealing

- e) More user friendly
- f) More complex tasks

8 - With which interface did you start using Git?

- a) Command Line Interface
- b) Graphical User Interface

9 - Which words would you associate with using Git through the CLI?

- a) Easy
- b) Efficient
- c) Fast
- d) Hard
- e) Time Consuming
- f) Helpful
- g) Controlled
- h) Slow
- i) Dispersive

10 - Which words would you associate with using Git through the GUI?

- a) Easy
- b) Efficient
- c) Fast
- d) Hard
- e) Time Consuming
- f) Helpful
- g) Controlled
- h) Slow
- i) Dispersive

11 - How many minutes per hour of developing do you spend using Git on average?

- a) 1-5 minutes
- b) 5-15 minutes
- c) 15-30 minutes
- d) More than 30 minutes

12 - Have you ever considered switching interface? (CLI to GUI or GUI to CLI)

- a) Yes
- b) No

2-13 - Do you Believe using you interface is the fastest way of using Git?

- a) Yes
- b) No

2 - 14 - Do you believe using your interface is the most efficient way of using Git?

- a) Yes
- b) No

3 - 13 - Why have you not already switched interface?

- a) \*Short answer Text\*

3 - 14 - Do you believe switching to the other interface would improve your efficiency / productivity?

- a) Yes
- b) No

3 - 15 - Do you believe switching to the other interface would improve your speed at developing?

- a) Yes
- b) No

## 6 Acknowledgment

## References

- Bagley, J. C. (2020). *Reasons to Prefer Command Line Interface (CLI) Software to GUIs*. Gist. URL: <https://gist.github.com/justincbagley/95cfaf9601b4af6f3afa93b4d2155abb> (visited on 12/15/2023).
- Forster, B. (Sept. 19, 2017). *Announcing GitHub Desktop 1.0*. The GitHub Blog. URL: <https://github.blog/2017-09-19-announcing-github-desktop-1-0/> (visited on 12/17/2023).
- Hultstrand, S. and R. Olofsson (2015). *Git - CLI or GUI : Which is most widely used and why?* URL: <https://urn.kb.se/resolve?urn=urn:nbn:se:bth-10539> (visited on 12/15/2023).
- Reimer, J. (May 5, 2005). “A history of the GUI”. In: (visited on 11/05/2023).
- Ruparelia, N. B. (Jan. 25, 2010). “The history of version control”. In: *ACM SIGSOFT Software Engineering Notes* 35.1, pp. 5–9. ISSN: 0163-5948. DOI: 10.1145/1668862.1668876. URL: <https://doi.org/10.1145/1668862.1668876> (visited on 12/15/2023).
- Smith, D. K. and R. C. Alexander (1999). *Fumbling the Future: How Xerox Invented, Then Ignored, the First Personal Computer*. iUniverse. 278 pp. ISBN: 978-1-58348-266-7.
- StackOverflow (May 22, 2022). *Stack Overflow Developer Survey 2022*. Stack Overflow. URL: [https://survey.stackoverflow.co/2022/?utm\\_source=social-share&utm\\_medium=social&utm\\_campaign=dev-survey-2022](https://survey.stackoverflow.co/2022/?utm_source=social-share&utm_medium=social&utm_campaign=dev-survey-2022) (visited on 12/15/2023).
- Straub, S. C. a. B. (Nov. 9, 2014). *Git - About Version Control*. URL: <https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control> (visited on 12/15/2023).