Assignment 3

User Research & Analysis

GROUP 3
RUI QIU #999292509

1. Summary of Research Results

1.1 Executive Summary

Using the questionnaire we designed in Phase II, I conducted an online survey in this semester's CSC207 class. Since version control is one of the early topic in this course, all students in this class have more or less some experience in command-line interface (CLI). So the response from this cluster would somehow reflect common problems encountered in learning and using command-line interface. And according to the survey, the major problems of command-line interface are its error forgiveness and rather steep learning curve. On the other hand, it is commonly agreed that CLI is a very efficient and simplified tool once used proficiently by the user.

1.2 Methodology Section

The questionnaires were designed for a face-to-face sampling procedure, but due to some personal reasons (not in Toronto during reading week), I made it online instead. Considering the fact that this does not involve any participation of researchers, the result should be as accurate as desired. The sampling subjects are all in CSC207 session, with a certain level of experience in CLI (mostly less than 3 years).

The questionnaire has three major parts: personal information, rating scales and subjective questions. The first part is used to divide our sample into different clusters. As this survey is conducted intentionally in a computer science class, the test subjects share lots of similarities, this part does not fully utilizes except we can divide our samples into different groups by their experience, and we also get to know what their most used CLIs are. The second part is used to quantify users' responses toward distinct aspects of CLI. Specifically, this part includes overall reaction, screen and appearance, terminology and system information, learning, last but not least, interface capabilities. Each aspect contains several features which need to be judged separately. The last part contains three questions, asking for users' subjective view over CLI. And some extra good / bad points of CLI are expected to be mentioned in the answers as well.

1.3 Results

There is only one artifact that needs to be pointed out: since the survey results heavily depend on sampling subjects' usual experience on CLI, especially the ones they use most, then the user experience would be different to some extent. In other words, we are not doing the survey in a completely controlled environment.

The general result of data is concluded in the following table:

Aspect	Feature	Average Score
Overall reaction	Wonderfulness	2.83
	Easiness	2.83
	Satisfaction	2.67
	Adequate Power	2.83
	Stimulation	2.33
Screen and appearance	Reading the screen	3
	Organization of information	2
Terminology and system information	Use of terms in the system	2.83
	Is terminology related to task?	2.83
	Position of message on screen	2.5
	Prompts for input	3.17
	Error messages	2.17
Learning	Exploring new features by trial and error	2.83
	Learning to operate the system	2.83
	Remembering names and use of commands	2.33
	Performing tasks is straightforward	3
Interface capabilities	Speed	3.17
	Reliability	3.17
	Mistake correction	2.17
	Designed for all levels of users	1.83

Based on data, the following problems are revealed:

- the organization of information in CLI is not clear;
- the way to deal with error is convenient (both error message displaying and correction);
- CLI is also not very user-friendly to all levels of users, particularly to amateurs.

However, on the other hand, we have to admit that CLI is a very efficient and effective tool when one user becomes proficient with it. So conclusively, CLI still receives acceptable overall scores. Besides, a sample subject also points out that customization is one of advantages of CLI.

Generally speaking, the scores of a selected feature centralize at a certain point, hence the data we collect are not quite polarized. But scores divide greatly in some features, such as **easiness, exploring new features by trial and error etc.** So, considering the years of experience the users have, we can assume that they are grouped into two main groups: amateur users and skilled users.

2. User Needs List

- To organize information on screen clearly (so that user can locate their desired information quickly)
- To improve error display (location on screen, highlight, with correction advice?)
- To remember and use commands more easily (probably with command input suggestion, auto correction?)
- To correct mistake smartly.
- To be more amateur-friendly.
- To be more 'tiredness-proof'.

3. Stakeholder Descriptions

The stakeholders of our system are all computer users who need to issue commands to control a program or an operating system and those prospective computer science students who will learn to use CLI in the future.

4. Personas

4.1 Primary Persona

CS freshman Eric (19)

Our primary persona is a frosh year student in university with limited programming knowledge and great eager to learn new stuffs. He also lacks of experience in CLI, therefore what often happens to him is that he always does not know what to input when he wants to perform a certain action.

4.2 Additional Persona

CS nerd Jason (23)
Forgetful Azu (30)
High school student Alice (16)

5. Scenario

Eric spent hours in his programming assignment, and finally finished it before the deadline. He was using svn to version control his files. He first wanted to move the files to the local copy folder. But unluckily he forgot the command and the detailed location of it. So he opened Finder to look for the address, afterwards he googled the internet and found the command he needed was '`mv dir1 dir2`'. Then he typed '`svn commit -m`' (which was wrong), but the error message returned. He typed '`svn commit m-`' (without changelog). Of course, the commitment is still unsuccessful, because he forgot the changelog. After retyping it again, because he did not know he can use up arrow to repeat previous command, adding the changlog 'Updated assignment 1', the commitment failed again, indicating his files are not under version control. Eric struggled with this problem for a while, finally resolved it with the help from one of his classmates. It turned out that he forgot to '`svn add filename`' before committing.

6. Task analysis

- Open Terminal.
- Move files into local copy.
  - Use `ls` and `cd` commands to locate the local copy.
  - Use `mv dir1 dir2` to move files to desired location.
- Commit the changes.
  - Use `svn add filename` to add files to version control.
  - Use `svn commit m- ‘updated assignment 1’` to commit the changes made and comment with what changes were made in English.
- Close Terminal.

7. Appendix

This part includes the research protocol, questionnaire and sample raw data.