Terminal Fever

Bringing human interface design to the command line

Tyler Cecil Louis Jencka Jesse Maes

Background

- Programmers and data scientists use the terminal
- We all love the efficiency and expressive power
- But the interface was designed in 1960 for a PDP-11
- Why should you have to memorize 100 three letter words?
- Can't we do better?

Problem Solved!

- Terminal Fever is a radically modern approach to the command line
- It knows the commands you use, and helps you discover the ones you don't
- Provide heads up information so you're never unsure of what's going on
- Allows any destructive command to be undone - so you can feel safe
- *It automates automation!* - bash one-liners are at your fingertips!

Target Audience

- Beginners who want to learn to use the command line
- Wizards who want to tab complete an entire rails app
- Everyone in between!

Beginners

- No experience with UNIX operating systems
- No experience with a CLI
- No experience with the state inherent to a shell, mainly the current working directory
- No knowledge of what programming is, or how to automate tasks
- No knowledge of the anatomy of a command (command name, arguments, options, et cetera)
- No knowledge of the software available on their system
- No knowledge of basic OS tasks, such as installing software
- Has experience with common GUI techniques - click-to-open, drag-and-drop, et cetera

Casual Users

- Understand the basics of how computers work
- Understand what computer programs are or has knowledge of how to program
- Only familiar with a GUI, or are more comfortable with pointing and clicking than typing
- Regularly needs to perform some repetitive task on their computer, but does not imagine the task could be automated
- Finds themselves lost if/when they use the CLI - doesn't understand the larger context of the system surrounding the shell
- Is unaware of most of the programs, commands, and command line options they could be using (has only seen the tip of the iceberg)
- Does not know how to apply technical knowledge or programming skills to system tasks in the CLI

Advanced User

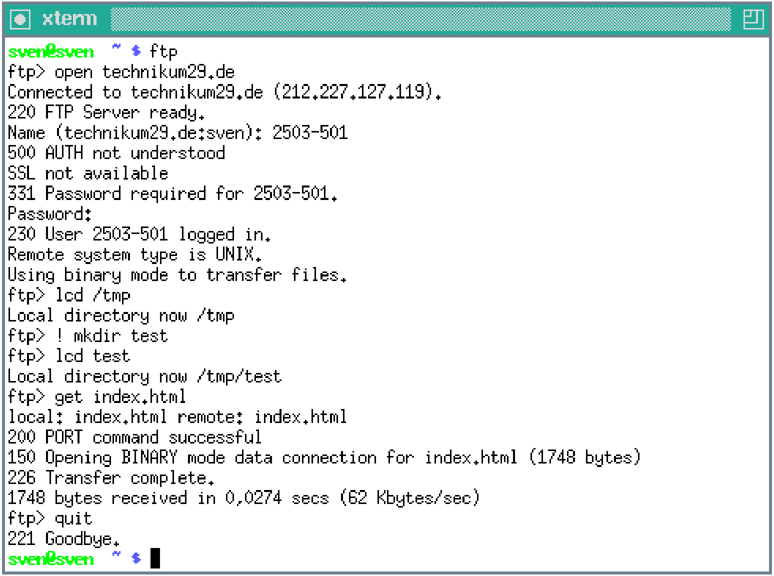
- Deep understanding of OS structure and interfaces
- Mastery of command line usage
- Commonly constructs aliases and automates tasks
- Consults system documentation to find advanced command options
- Understands that task inference is frequently achievable from context in the shell

#GOALS

- As any user, I want to be able to undo what I've done.
- As a beginner, I want to know what I can do with a file.
- As a casual user, I want to automate my workflow.
- As a beginner, I want to know what the command is going to do.
- As an expert user, I want heads-up information about my system as I use it.

Hasn't this been done before?

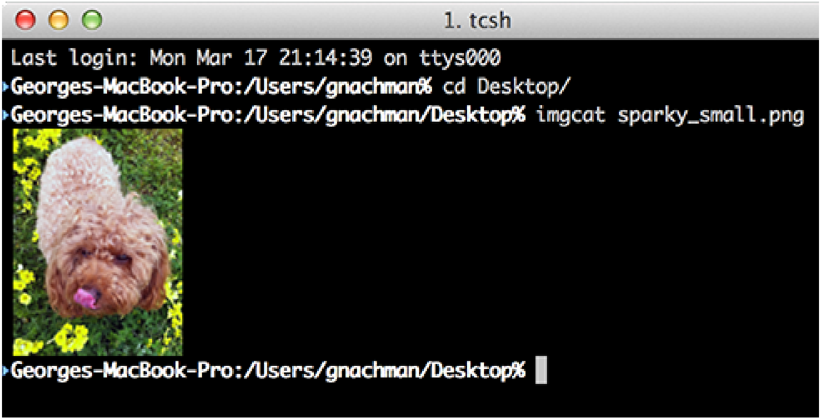
God no



```
sven@sven ~ $ ftp
ftp> open technikum29.de
Connected to technikum29.de (212.227.127.119).
220 FTP Server ready.
Name (technikum29.de:sven): 2503-501
500 AUTH not understood
SSL not available
331 Password required for 2503-501.
Password:
230 User 2503-501 logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> lcd /tmp
Local directory now /tmp
ftp> ! mkdir test
ftp> lcd test
Local directory now /tmp/test
ftp> get index.html
local: index.html remote: index.html
200 PORT command successful
150 Opening BINARY mode data connection for index.html (1748 bytes)
226 Transfer complete.
1748 bytes received in 0.0274 secs (62 Kbytes/sec)
ftp> quit
221 Goodbye.
sven@sven ~ $ 
```

Figure 1: xterm

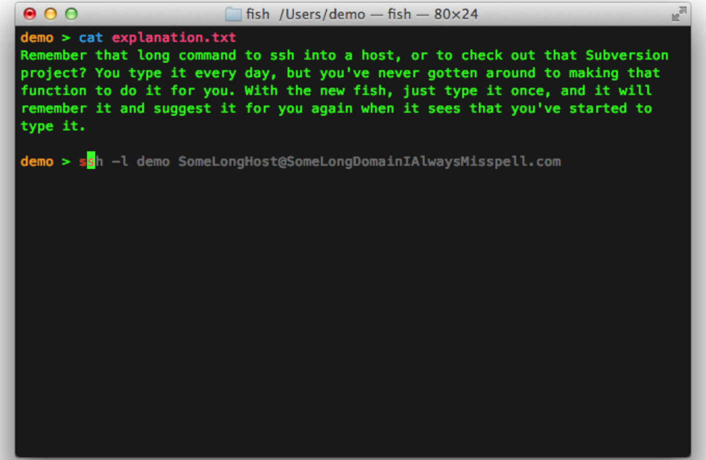
Still not really. . .



```
Last login: Mon Mar 17 21:14:39 on ttys000
Georges-MacBook-Pro:/Users/gnachman% cd Desktop/
Georges-MacBook-Pro:/Users/gnachman/Desktop% imgcat sparky_small.png

Georges-MacBook-Pro:/Users/gnachman/Desktop%
```

Figure 2: iTerm 2

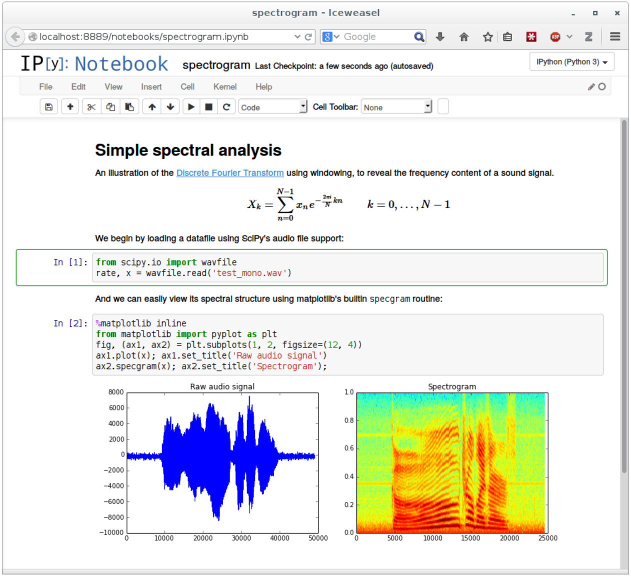
Not far enough!



```
demo > cat explanation.txt
Remember that long command to ssh into a host, or to check out that Subversion
project? You type it every day, but you've never gotten around to making that
function to do it for you. With the new fish, just type it once, and it will
remember it and suggest it for you again when it sees that you've started to
type it.

demo > ssh -l demo SomeLongHost@SomeLongDomainIAlwaysMisspell.com
```

Figure 3: `zsh && fish`

I NEED MORE POWER



spectrogram - Iceweasel
localhost:8889/notebooks/spectrogram.ipynb
Google
IP[y]: Notebook
spectrogram Last Checkpoint: a few seconds ago (autosaved)
IPython (Python 3)
File Edit View Insert Cell Kernel Help
Code
Cell Toolbar: None

Simple spectral analysis

An illustration of the Discrete Fourier Transform using windowing, to reveal the frequency content of a sound signal.

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}kn} \qquad k = 0, \dots, N-1$$

We begin by loading a datafile using SciPy's audio file support:

```
In [1]: from scipy.io import wavfile
        rate, x = wavfile.read('test_mono.wav')
```

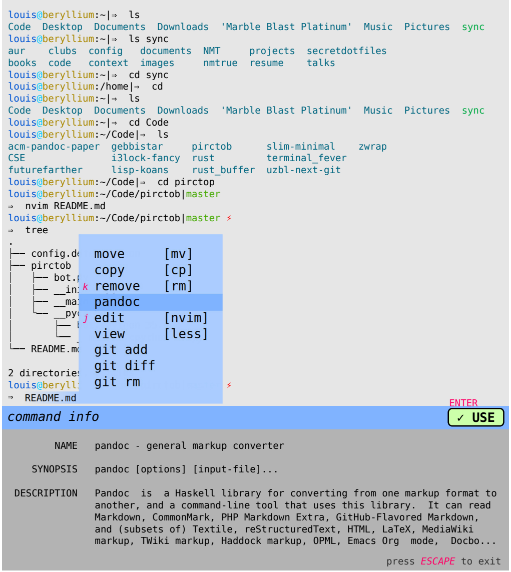
And we can easily view its spectral structure using matplotlib's builtin `specgram` routine:

```
In [2]: %matplotlib inline
        from matplotlib import pyplot as plt
        fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
        ax1.plot(x); ax1.set_title('Raw audio signal')
        ax2.specgram(x); ax2.set_title('Spectrogram');
```

Figure 4: Jupyter

Features

Autocomplete



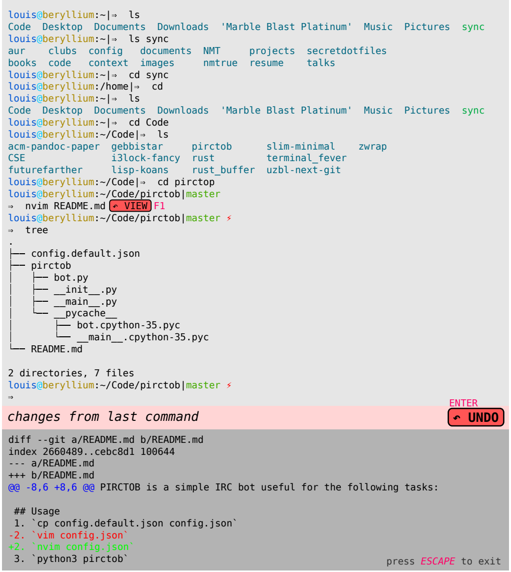
```
louis@beryllium:~|⇒  ls
Code  Desktop  Documents  Downloads  'Marble Blast Platinum'  Music  Pictures  sync
louis@beryllium:~|⇒  ls sync
aur    clubs  config  documents  NMT     projects  secretdotfiles
books  code   context  images    nmtrue  resume    talks
louis@beryllium:~|⇒  cd sync
louis@beryllium:/home|⇒  cd
louis@beryllium:~|⇒  ls
Code  Desktop  Documents  Downloads  'Marble Blast Platinum'  Music  Pictures  sync
louis@beryllium:~|⇒  cd Code
louis@beryllium:~/Code|⇒  ls
acm-pandoc-paper  gebbistar     pirctob      slim-minimal    zwrap
CSE               i3lock-fancy  rust         terminal_fever
futurefarther     lisp-koans    rust_buffer  uzbl-next-git
louis@beryllium:~/Code|⇒  cd pirctop
louis@beryllium:~/Code/pirctob|master
⇒  nvim README.md
louis@beryllium:~/Code/pirctob|master ⚡
⇒  tree
.
├── config.de
├── pirctob
│   ├── bot.p
│   ├── __ini
│   ├── __mai
│   └── __pyc
│       ├── b
│       └──
└── README.md

2 directories
louis@berylli                                            ⚡
⇒  README.md
```

command info

```
       NAME   pandoc - general markup converter

   SYNOPSIS   pandoc [options] [input-file]...

DESCRIPTION   Pandoc  is  a Haskell library for converting from one markup format to
              another, and a command-line tool that uses this library.  It can read
              Markdown, CommonMark, PHP Markdown Extra, GitHub-Flavored Markdown,
              and (subsets of) Textile, reStructuredText, HTML, LaTeX, MediaWiki
              markup, TWiki markup, Haddock markup, OPML, Emacs Org  mode,  Docbo...

                                                                press ESCAPE to exit
```

Figure 5

Undo Window



```
louis@beryllium:~|⇒  ls
Code  Desktop  Documents  Downloads  'Marble Blast Platinum'  Music  Pictures  sync
louis@beryllium:~|⇒  ls sync
aur    clubs  config   documents  NMT     projects  secretdotfiles
books  code   context  images     nmtrue  resume    talks
louis@beryllium:~|⇒  cd sync
louis@beryllium:/home|⇒  cd
louis@beryllium:~|⇒  ls
Code  Desktop  Documents  Downloads  'Marble Blast Platinum'  Music  Pictures  sync
louis@beryllium:~|⇒  cd Code
louis@beryllium:~/Code|⇒  ls
acm-pandoc-paper  gebbistar    pirctob      slim-minimal    zwrap
CSE               i3lock-fancy rust         terminal_fever
futurefarther     lisp-koans   rust_buffer  uzbl-next-git
louis@beryllium:~/Code|⇒  cd pirctop
louis@beryllium:~/Code/pirctob|master
⇒  nvim README.md
louis@beryllium:~/Code/pirctob|master ⚡
⇒  tree
.
├── config.default.json
├── pirctob
│   ├── bot.py
│   ├── __init__.py
│   ├── __main__.py
│   └── __pycache__
│       ├── bot.cpython-35.pyc
│       └── __main__.cpython-35.pyc
└── README.md

2 directories, 7 files
louis@beryllium:~/Code/pirctob|master ⚡
⇒
```

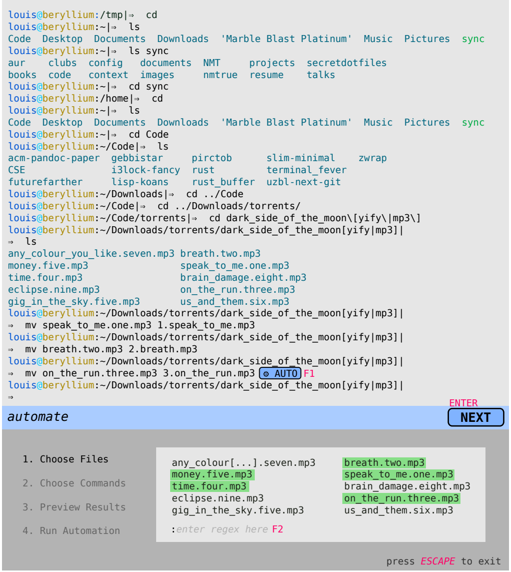
changes from last command

```
diff --git a/README.md b/README.md
index 2660489..cebc8d1 100644
--- a/README.md
+++ b/README.md
@@ -8,6 +8,6 @@ PIRCTOB is a simple IRC bot useful for the following tasks:

 ## Usage
 1. `cp config.default.json config.json`
-2. `vim config.json`
+2. `nvim config.json`
 3. `python3 pirctob`
```

press *ESCAPE* to exit

Figure 6

Automation Helper



```
louis@beryllium:/tmp|⇒  cd
louis@beryllium:~|⇒  ls
Code  Desktop  Documents  Downloads  'Marble Blast Platinum'  Music  Pictures  sync
louis@beryllium:~|⇒  ls sync
aur    clubs  config   documents  NMT     projects  secretdotfiles
books  code   context  images     nmtrue  resume    talks
louis@beryllium:~|⇒  cd sync
louis@beryllium:/home|⇒  cd
louis@beryllium:~|⇒  ls
Code  Desktop  Documents  Downloads  'Marble Blast Platinum'  Music  Pictures  sync
louis@beryllium:~|⇒  cd Code
louis@beryllium:~/Code|⇒  ls
acm-pandoc-paper  gebbistar     pirctob      slim-minimal    zwrap
CSE               i3lock-fancy  rust         terminal_fever
futurefarther     lisp-koans    rust_buffer  uzbl-next-git
louis@beryllium:~/Downloads|⇒  cd ../Code
louis@beryllium:~/Code|⇒  cd ../Downloads/torrents/
louis@beryllium:~/Code/torrents|⇒  cd dark_side_of_the_moon\[yify\|mp3\]
louis@beryllium:~/Downloads/torrents/dark_side_of_the_moon[yify|mp3]|
⇒  ls
any_colour_you_like.seven.mp3 breath.two.mp3
money.five.mp3                speak_to_me.one.mp3
time.four.mp3                 brain_damage.eight.mp3
eclipse.nine.mp3              on_the_run.three.mp3
gig_in_the_sky.five.mp3       us_and_them.six.mp3
louis@beryllium:~/Downloads/torrents/dark_side_of_the_moon[yify|mp3]|
⇒  mv speak_to_me.one.mp3 1.speak_to_me.mp3
louis@beryllium:~/Downloads/torrents/dark_side_of_the_moon[yify|mp3]|
⇒  mv breath.two.mp3 2.breath.mp3
louis@beryllium:~/Downloads/torrents/dark_side_of_the_moon[yify|mp3]|
⇒  mv on_the_run.three.mp3 3.on_the_run.mp3 ⚙ AUTO F1
louis@beryllium:~/Downloads/torrents/dark_side_of_the_moon[yify|mp3]|
⇒
```

automate

ENTER NEXT

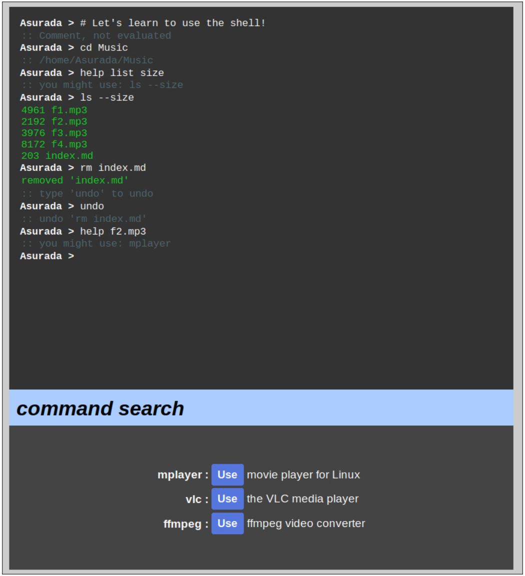
1. Choose Files
2. Choose Commands
3. Preview Results
4. Run Automation

```
any_colour[...].seven.mp3    breath.two.mp3
money.five.mp3               speak_to_me.one.mp3
time.four.mp3                brain_damage.eight.mp3
eclipse.nine.mp3             on_the_run.three.mp3
gig_in_the_sky.five.mp3      us_and_them.six.mp3

:enter regex here F2
```

press *ESCAPE* to exit

Figure 7

Prototype



```
Asurada > # Let's learn to use the shell!
:: Comment, not evaluated
Asurada > cd Music
:: /home/Asurada/Music
Asurada > help list size
:: you might use: ls --size
Asurada > ls --size
4961 f1.mp3
2192 f2.mp3
3976 f3.mp3
8172 f4.mp3
203 index.md
Asurada > rm index.md
removed 'index.md'
:: type 'undo' to undo
Asurada > undo
:: undo 'rm index.md'
Asurada > help f2.mp3
:: you might use: mplayer
Asurada >
```

command search

mplayer : Use movie player for Linux

vlc : Use the VLC media player

ffmpeg : Use ffmpeg video converter

Figure 8: Link!

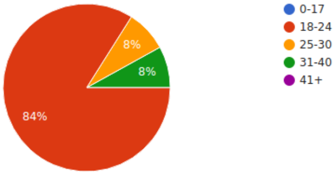
Testing (one, two)

- Survey test users on prototype
- Collect demographic data:
  - Age
  - Gender
  - Education
  - Major
  - Primary OS
- Guage familiarity, knowledge, and efficiency

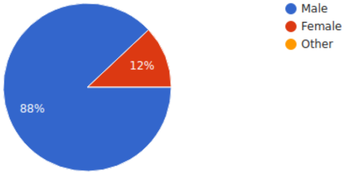
Did you like it?

- Yes or yes?
- Was it useful?
- Was the interface usable?
- Could you learn from it?

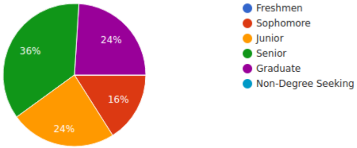
Results



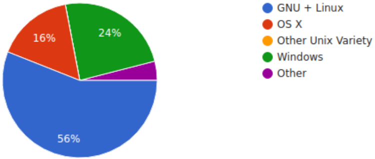
8%
8%
84%
● 0-17
● 18-24
● 25-30
● 31-40
● 41+



12%
88%
● Male
● Female
● Other



36%
24%
16%
24%
● Freshmen
● Sophomore
● Junior
● Senior
● Graduate
● Non-Degree Seeking



16%
24%
56%
● GNU + Linux
● OS X
● Other Unix Variety
● Windows
● Other

#DATASCIENCE

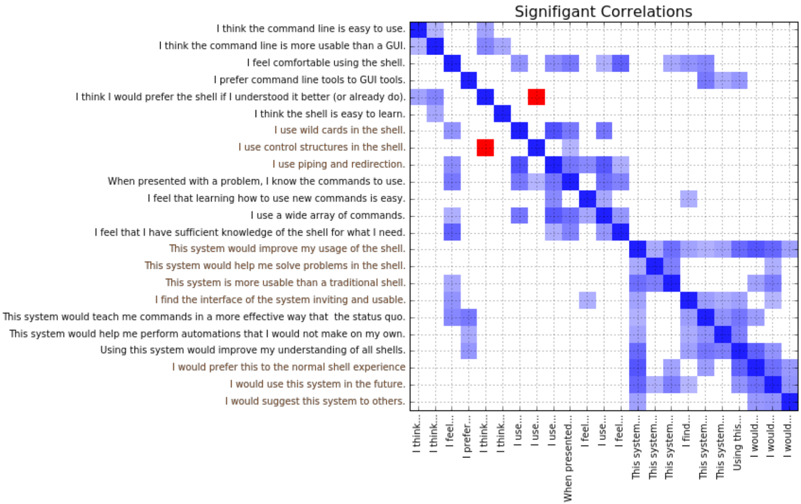


Figure 9

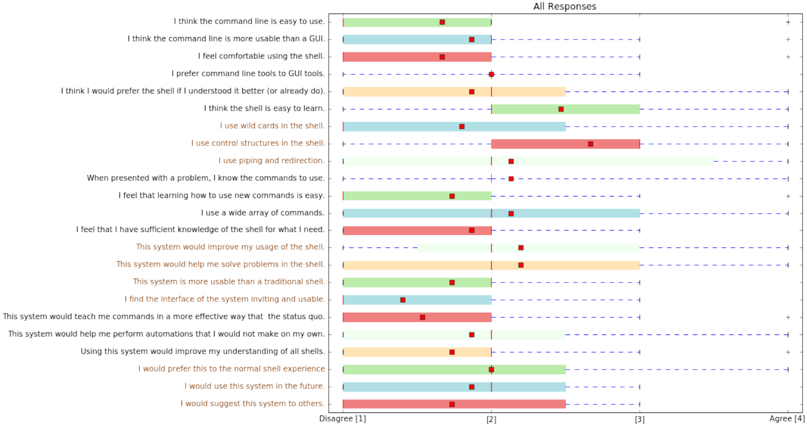

All Responses
I think the command line is easy to use.
I think the command line is more usable than a GUI.
I feel comfortable using the shell.
I prefer command line tools to GUI tools.
I think I would prefer the shell if I understood it better (or already do).
I think the shell is easy to learn.
I use wild cards in the shell.
I use control structures in the shell.
I use piping and redirection.
When presented with a problem, I know the commands to use.
I feel that learning how to use new commands is easy.
I use a wide array of commands.
I feel that I have sufficient knowledge of the shell for what I need.
This system would improve my usage of the shell.
This system would help me solve problems in the shell.
This system is more usable than a traditional shell.
I find the interface of the system inviting and usable.
This system would teach me commands in a more effective way that the status quo.
This system would help me perform automations that I would not make on my own.
Using this system would improve my understanding of all shells.
I would prefer this to the normal shell experience
I would use this system in the future.
I would suggest this system to others.
Disagree [1]
[2]
[3]
Agree [4]

Figure 10

Conclusion

- People found the prototype confusing (30% dropped before taking the exit survey)
- *Even more* information needs to be given to the users
- Experienced users felt they didn't need a teaching tool
- More research, with an improved prototype, can tell us even more

Questions?