

SE 3XA3: Problem Statement

Lines Per Minute (lpm)

Team #16, Lines Per Minute (lpm)
Jay Mody - modyj - 400195508
Jessica Lim - limj31 - 400173669
Maanav Dalal - dalalm1 - 40017811508

January 28th, 2021

Table 1: Revision History

Date	Developer(s)	Change
January 28, 2021	Jay/Jessica/Maanav	Initial document write-up. Completed.
Date2	Name(s)	Description of changes
...

Problem & Relevance

With the rise of work-from-home situations, being proficient with your keyboard is more important than ever before. Increasing one's average typing speed has become a goal of many virtual workers/students around the globe. Words Per Minute (*wpm*) is a command-line package that allows users to practice typing, and get statistics regarding their speed and accuracy. The *wpm* python package works directly through the command line, so it can be used offline and without requiring access to a web browser.

Software developers are constantly trying to increase their typing speed in order to increase productivity and efficiency. While many tools exist for increasing typing speed using plain-English text, there are very few tools that are targeted specifically towards programmers & developers. The keys and patterns used when coding differ from natural language, so plain English phrases are less beneficial towards typing speed for programmers. Furthermore, command-line packages are more accessible for developers, as they frequently use the command line to code, rather than a web browser.

We propose Lines Per Minute (*lpm*), a python package built upon *wpm*. *lpm* will include additional features targeted to programmers and software developers.

Description & Issues

The main issue *lpm* will tackle is the addition of code snippets as an option to make the experience more tailored for developers. *lpm* will utilize various code snippets from open source software from a variety of programming languages to provide this functionality.

In addition, the existing *wpm* package misses a few core components of a well-architected software project. For example, despite the inclusion of a testing framework, there are no unit tests present. As we have learned, testing is an essential part of a software project to ensure it is working as expected. Since there is a lack of testing throughout the project, we as users and developers are unable to know how well the software works, or test if new changes break the existing functionality. Specifically, we'd like to introduce testing for different versions of Python and tests to ensure the accuracy of various stats, including words per minute, lines per minute, and error rates.

Furthermore, there is a lack of options for users to choose between when testing their *wpm*. They may be training their typing skills to become better programmers, or better at typing numbers and there is no support for code snippets or different varieties of *wpm* tests (no punctuation, numbers, capital letters, etc.).

Context

Stakeholders

The clients (or users) are the primary stakeholders. The clients are individuals looking to increase their typing speed when coding. While *wpm* focuses on typing speed on plain English text, *lpm* focuses on typing speed on code snippets. This makes our target audience software developers who frequently use the command line and want to increase their typing speed when coding.

Other stakeholders include PyPI (The Python Package Index) who will be distributing our package, GitLab who is hosting our source code, and the authors of the open source projects that the code snippets are taken from.

Software Environment

The original *wpm* package is written in plain python (with no additional dependencies). The package is managed and distributed via pip. Given a valid python and pip installation, *wpm* can be installed via:

```
$ pip install wpm
```

The *wpm* interface is accessed and delivered as a CLI (command-line interface), which can be run via:

```
$ wpm
```

We plan to use the same technologies to redevelop, build-upon, and deliver Lines-Per-Minute. The *lpm* package should be accessed and delivered in a similar way

```
$ pip install lpm
```

```
$ lpm
```