Software Engineering - Report of an Influential Software Engineer and their work

Don Ho, Creator of Notepad++

Choosing a topic for this essay was no simple task. There are a number of great Software Engineers to choose from, many of which have had great Biographies written of them already. Instead, the focus of this essay will be a slightly lesser-known developer by the name of Don Ho, the creator of popular windows development environment Notepad++. Writing a biography of a software engineer can be challenging, as they tend to let their work speak for them. This why the focus of a large part of this essay will be on said work, and how it affected the development landscape.

History:

Every developer has their preferred development environment(s). Developers value different things in their IDE's of choice. Some prefer to have thousands of plugins available, and are willing and able to sacrifice speed and performance to get them. However, others focus more on the speed and performance of an IDE than the additional features many development suites bring nowadays at the cost of increased load and startup times.

Don Ho was one of these developers. In 2003, Don Ho was dissatisfied with his company's choice of java-based development environment, JEXT, due to its poor performance.
He set out to create a new, better editor using C++ using Scintilla, which is a free open source library that supports text editing, to replace JEXT in his workplace.
His company quickly rejected his idea, so he continued development on Notepad++ in his spare time. He knew that he wanted to create a fast, efficient tool to help himself and other developers enhance their workflow. In an interview in 2017, Ho describes how he was inspired by the Free Software Foundation(FSF) and Linux, and that it played a huge role in the open-source "Copyleft" development of Notepad++. The first version of Notepad++ was made available that same year on Sourceforge.
Ho describes the early years of Notepad++ as quiet. He didn't receive many feature requests, and mainly focused his time on enhancing the GUI and adding features he thought were necessary. However, as time went on, pull requests began popping up that were beyond the scope of the original program, so the plugin system had to be built and put in place.
The Notepad++ community grew, and with it, came more and more plugins and features, to grow the list to over a hundred plugins, ranging from autosavers to autocomplete plugins for certain languages. The combined effort of the Notepad++ community has guided the development of the app from a simple text-editor to a feature rich development environment, which can be adjusted to suit any developers individual needs.
This is the main strength of Notepad++. A strong, efficient, performance-focused base, supplemented by a slew of useful plugins tailored to most developers' individual needs.

Popularity and Impact:

These features, along with Notepad++'s Open Source and free business model, contribute to it being one of the most popular development tools in the world, despite it only being available on windows. A 2017 stackoverflow survey shows that Notepad++ is the world's second most popular development environment across the board, and certainly the most popular free solution.

The most important feature of N++ is its performance and efficiency. Ho build the program from the ground up with speed in mind. Notepad++ boasts a sub-second startup time, which, cumulatively, saves developers a huge amount of time, as restarts and reopenings/closings are a common part of the development process; sometimes accidentally, sometimes out of necessity. This, combined with the autosave feature, results in a very seamless and document-safe experience.

A further important feature of N++ is its focus on multi-language support. This flexibility gives it an edge over many other development environments which specialise in a specific language or set of languages. They may have some features which N++ does not, for example better code autocompletion, but N++'s status as a competent jack of all trades, which handles anything you throw at it fairly well, as well as being highly customisable and fast, are the features which have earned it the respected position it holds in the development sphere. One of the features most integral to this flexibility is the auto-highlighting feature, which comes built-in for most of the most popular programming languages, and supports customization for others not yet implemented. These custom highlighting settings can and have been shared among the community, with some being added to the default installation.

Of course, one of the biggest and most important reasons for Notepad++'s success is its price tag. The other most popular editors, such a Microsoft's Visual Studio or Sublime Text, cost a significant sum of money to non-students. One could compare N++ to Vim, another free and open-source text editor, the main difference being that Vim has an extremely steep learning curve, whereas Notepad++'s is almost non-existent, as it appears as any other text editor.

All these factors combined resulted in Notepad++'s rise to popularity. This popularity, in turn, results in Notepad++ having had a huge impact on the development world in general. A simple program, developed in a software engineer's free time, which nonetheless improves the workflow of a huge number of developers globally.

Trade-offs:

The uses of Notepad++ are varied. Whether it is your main development environment or it's used only as a scratchpad, N++ performs. This was Don Ho's intention. To create something simple, easy to use and efficient. He developed Notepad++ because he was fed up with the ineffective tools of another man.

However, there are certain drawbacks to creating something as quick and efficient as Notepad++. Its heavy use of the win32 api has caused portability issues, as the program cannot be easily ported using, say, wxWidget, to another platform, like Linux. The plan to port Notepad++ had to be put on hold due to this, though Ho has been quoted as having said that that isn't to say it will never happen.

Additionally, the focus only on the performance aspect of the program has also resulted in a very dated appearance, which, in the modern age, can actually be a significant turn-off for any new user, tough that may in fact appeal to some of the more old-school users of the program.

Don Ho:

Not much is known of Don Ho. He does not live a life of secrecy, but, like most other software engineers and developers, he does not live a high-profile lifestyle. He is not seen by the public eye. Ho completed his Bachelor's in Computer Science in Paris Diderot University 1999, his Masters the year after, and his Post-Graduate Diploma in CS in 2001.

Ho is relatively unknown, even among his peers, despite his efforts personally affecting millions on a daily basis. A far cry from "rock-star" software engineers and developers such as Bill Gates. Ho's work has had such a widespread impact on the windows development world, and yet his efforts go widely unappreciated. Of course acclaim was not his intention, nor can it be expected, but it is interesting to examine the general anonymity of software engineers, and how subtle the ways in which their work can have an impact are.

Of course, Ho is not *solely* responsible for the product we have today. The open source nature of the project means that any number of people have contributed and enhanced the program in ways unknown. From small typo fixes to whole language translations, the N++ community has no doubt had a massive impact on the program we use today. In fact Ho was so stauch in his support for a FOS product, that in 2010, when the U.S. Government imposed upon open-source developers to restrict access to their projects from certain countries (North Korea, Sudan, Syria etc.), Notepad++ released a version on TuxFamily, in order to escape U.S. jurisdiction.

Ho's guidelines for pull requests also lend an interesting insight into the development process. To only submit one feature at a time, respecting the existing coding style, and without reformatting source code. These rules, if followed, result in a very easy code review process, which, on a one-person project such as this, is imperative for a reasonably quick workflow.

Features:

Quick workflow was the entire point of Notepad++, and has been its focus since its inception. Every aspect of N++ is built with efficiency of workflow in mind. The incredibly quick opening speeds, the multiple tab support, the autosave feature in case of crashes, etc. There really isn’t much bad that can be said about Notepad++. It is certainly a program which has accomplished the goals set out for it. With time, many projects can accumulate bloat and unnecessary frills, but Don Ho has stayed true to his goals, and, by maintaining full control of the project with his original vision still in mind, hasn’t allowed it to become corrupted by others.

A project which Ho started in his free time, to solve a problem for himself and his company, has become a mainstay in most windows developers’ repertoire.

Conclusion:

To bring this to a close, some thoughts on how Software Engineering in general.
When creating a new project, it is important to have a vision for said project from the get-go, and to allow that vision to influence every aspect of the development process. That way, one will be sure to accomplish the goals one set out for. Also, the importance of taking criticism and feedback from users, and handling that feedback with dignity and respect, accepting the good and the bad, all for the good of the final product. Community was a huge part of what made Notepad++ such a success story, and continues to be a hallmark of good open-source development. Bug fixes and crash reports are number one, because end user experience is everything. Lastly, the importance of the recognition of the many software engineers and developers who go unrecognized, despite their contributions to the field.