Whitepaper: Standard Editor for Teams

This whitepaper explains why programmers tend to cling to the text editors they are used to using and are not always eager to move to a standardized product. Here we explain why a standardized text editor, one that is complete Integrated Development Environment (IDE), benefits the development organization.

Standardized Software

In the past, in Europe, each country's railroads used a different width for their train tracks. When a train leaving, say, France, went into Spain, the train would have to stop and unload its cargo and load it onto a different train at the border. Then the railroad adopted standard rail width, thus increasing efficiency.

Standardization in software bring efficiency too. For example, web services lets companies write interfaces between customers or products without having to program a new, proprietary interface for a new customer or product.

Eclipse is the best example of a standardized text editor. It is the most popular IDE in the world, supporting many languages. It is Open Source too, so many third-parties have developed add-ons making it even more valuable. An example of that is the Android Development Kit, which is an Eclipse add-on.

Why Editors are Not Standardized

Editors are not standardized because they were developed for different platforms at different times. Vi is the most common editor for UNIX. It was written over 40 years ago when engineers at Bell Laboratories developed UNIX. Emacs was considered an improvement over that, but it still is difficult for the beginner.

Other editors were developed to replace Notepad, which is a very basic text editor for Windows with limited functions. Notepad++ and Ultraedit brought additional functions to Windows editing and gave these text editors some knowledge of the language the programmer was using.

Programmers tend to cling to that they know. An editor becomes an extension of their personality as they develop muscle memory. That means they can make changes rapidly to text without thinking about what keys to press. Programmers are defensive about which editor they use. When they use complicated black and white tools like vi, they make fun of people who use graphical interfaces. This is the hacker culture, where people take pride in complexity. But such complexity drives down the efficiency of a development team for many reasons.

Management does not always consider the choice of an editor important when programmers are not using compiled languages, since the output does not matter. Engineers writing C++

code, for example, are encouraged to use Microsoft Visual C++, so that everyone across the team sets the same compiler options.

Why Editors should be Standardized

Python is the only programming language where spacing matters, as spacing replaces curly braces as delimiters of logical blocks in that language. In other languages, having different indentation does not matter, except it makes one programmer’s block of code look different that another’s. Eclipse, for example, eliminates that by including a function to format code in a common manner derived from templates. Some text editors call this pretty print. Eclipse also includes logic to verify that a programmer’s code meets company coding standards.

If different programmers use different editors, the only way to make the code follow the same standards and have the same format is if each person configures the different editors to produce the same output. That takes time and effort and drives up costs for the project. It is better to use a standardized product to push out the same configuration to everyone.

Another reason to get programmers to use the same editor is so that the fastest programmers can teach those who are less experienced or who do not learn as quickly. This is because the experienced programmer can show the other programmers how to do something using a common tool.

A text editor that is language-aware speeds development. For example, a language like VHDL is strongly typed. That means all variables must be declared. A language-aware IDE enforces that requirement.

An IDE also checks code as the developer types. For example, it would not allow assigning a string to an integer. Plus it assists the developer with object-aware and language-aware code completion. For example, if a developer makes a reference to an object, the editor will list the object’s functions and arguments as the programmer types.

Programmers should be encouraged to make a small sacrifice in their short term productivity by moving onto a common platform for the benefit of the team. In the long run, this benefits the project.

Conclusion

Programmers often see themselves as artists. Artists do not like to be told how to create art. But they can still be artistic and clever in what they write, but do so with a common tool. Letting each developer write code with whatever editor they want is costing your company money. A common tool for VDHL programming is Sigasi. It is built on Eclipse, which is the world’s standard.