

# From an Idea to a Product

MIKE UNWALLA | Senior Member

IN OCTOBER 2012, TechScribe released an open-source term checker for the controlled language ASD-STE100 ([www.asd-ste100.org](http://www.asd-ste100.org)). Approximately nine years earlier, I had the initial idea.

A controlled language helps to make text as clear as possible. For example, in ASD-STE100, the term *make sure* is approved, but synonyms such as *check*, *confirm*, *ensure*, *insure*, and *verify* are not approved.

In 2003, I thought about buying software to help me conform to ASD-STE100. Software was available, but it cost many thousands of dollars. I did not buy software. Instead, I tried to customize Microsoft Word.

My first idea was to use an exclude dictionary, but to exclude a multi-word term is not possible. I tried to “hack” the Microsoft Word dictionary, but that attempt was not successful. I tried to make Word use only custom dictionaries, but that is not possible (<https://groups.google.com/forum/?fromgroups=#!topic/microsoft.public.word.spelling.grammar/AeKcZSmS80Y>).

For four years, I did not think about the problem. Then, at the ISTC technical communication conference in 2007, I met an expert VBA coder. She wrote some Word macros that highlight different types of terms in different colors.

The macros are excellent, but they give no guidelines about the problems that they identify. For example, if a term is highlighted in yellow, I know that the term is not approved. However, the macros do not tell me whether approved alternatives exist. If an approved alternative exists, the macros do not tell me what it is.

By now, I wanted more than a lookup tool.

A local software developer was interested in the project and started to develop software. After some weeks, he lost interest.

I searched the Internet to find low-cost software that I could customize. Many times, I found software that initially appeared to be what I wanted. However, none of the software could deal with multi-word terms.

I found LanguageTool, which is open-source proofreading software ([www.languagetool.org](http://www.languagetool.org)). Unfortunately, I did not evaluate LanguageTool sufficiently well to know that it was perfect for the project.

In 2009, I learned about the SALTcymru project ([www.saltcymru.org](http://www.saltcymru.org)), which is related to the Language Technologies Unit at Bangor University ([www.bangor.ac.uk/canolfanbedwyr/technologau\\_iaith.php.en](http://www.bangor.ac.uk/canolfanbedwyr/technologau_iaith.php.en)). During the next few months, I exchanged email messages with members of the team, attended meetings, and spoke about international-

ized English for international communication. We spoke about the possible development of software.

I thought that the term checker had a commercial future. Advice for business development was available from regional development agencies. I took what was available. I thought about price strategies, publicity, brand building, selling on the Internet ...

In 2010, TechScribe was awarded an innovation voucher, which is a small grant ([www.innovateuk.org/delivering\\_innovation/innovation-vouchers.ashx](http://www.innovateuk.org/delivering_innovation/innovation-vouchers.ashx)). Now, money was available to pay for the development of prototype software. The Language Technologies Unit developed an online term checker for TechScribe ([www.techscribe.co.uk/ta/st2-term-checker-bangor-prototype.htm](http://www.techscribe.co.uk/ta/st2-term-checker-bangor-prototype.htm)).



www.shutterstock.com/ Antonov Roman

As proof-of-concept, the software from Bangor University is excellent. However, it is a lookup tool. Also, I cannot change the terms that are in the software.

In 2011, I looked again at LanguageTool. Rules for a language are in two XML files. LanguageTool gives a framework in which to create an effective term checker. I decided to make the effort to learn how to make LanguageTool do what I wanted. After 18 months, I had a term checker that was sufficiently good to release.

The term checker is more than a lookup tool. For example, in ASD-STE100, the term *pump* is approved as a noun but not as a verb. The term checker finds the term *pump* only when it is used as a verb. The analysis uses pattern matching. For example, in the text “the X is ...,” X is a noun. X cannot be a verb or another part of speech. If the term *pump* is not in a pattern in which *pump* is a noun, then the term checker identifies the term as a possible error.

I continue to develop the LanguageTool rules for ASD-STE100. The project time is now more than 1350 hours. I hope to get a return on my investment by selling a term checker for ASD-STE100 issue 6.

A free term checker for ASD-STE100 issue 3 is on [www.simplified-english.co.uk](http://www.simplified-english.co.uk). Let me know what you think about it at [mike@techscribe.co.uk](mailto:mike@techscribe.co.uk). 