SOURCETRAIL

THE NAVIGATION SYSTEM FOR SOFTWARE ENGINEERS
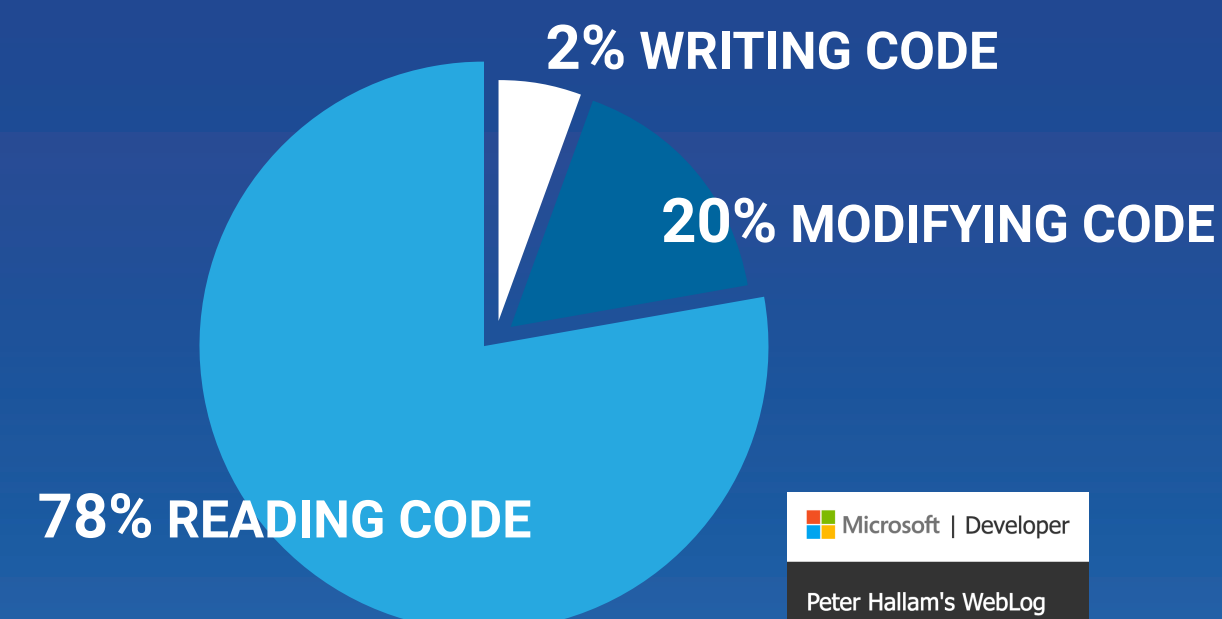
Eberhard Gräther (egraether@coati.io)

Software developers spend about 70% of their time figuring out how existing source code works, but common code editing tools are not made for this task.

Sourcetrail helps software engineers explore and navigate unfamiliar source code by combining an interactive graph visualization, a concise code view and a powerful search algorithm, all built into an easy-to-use cross-platform developer tool.
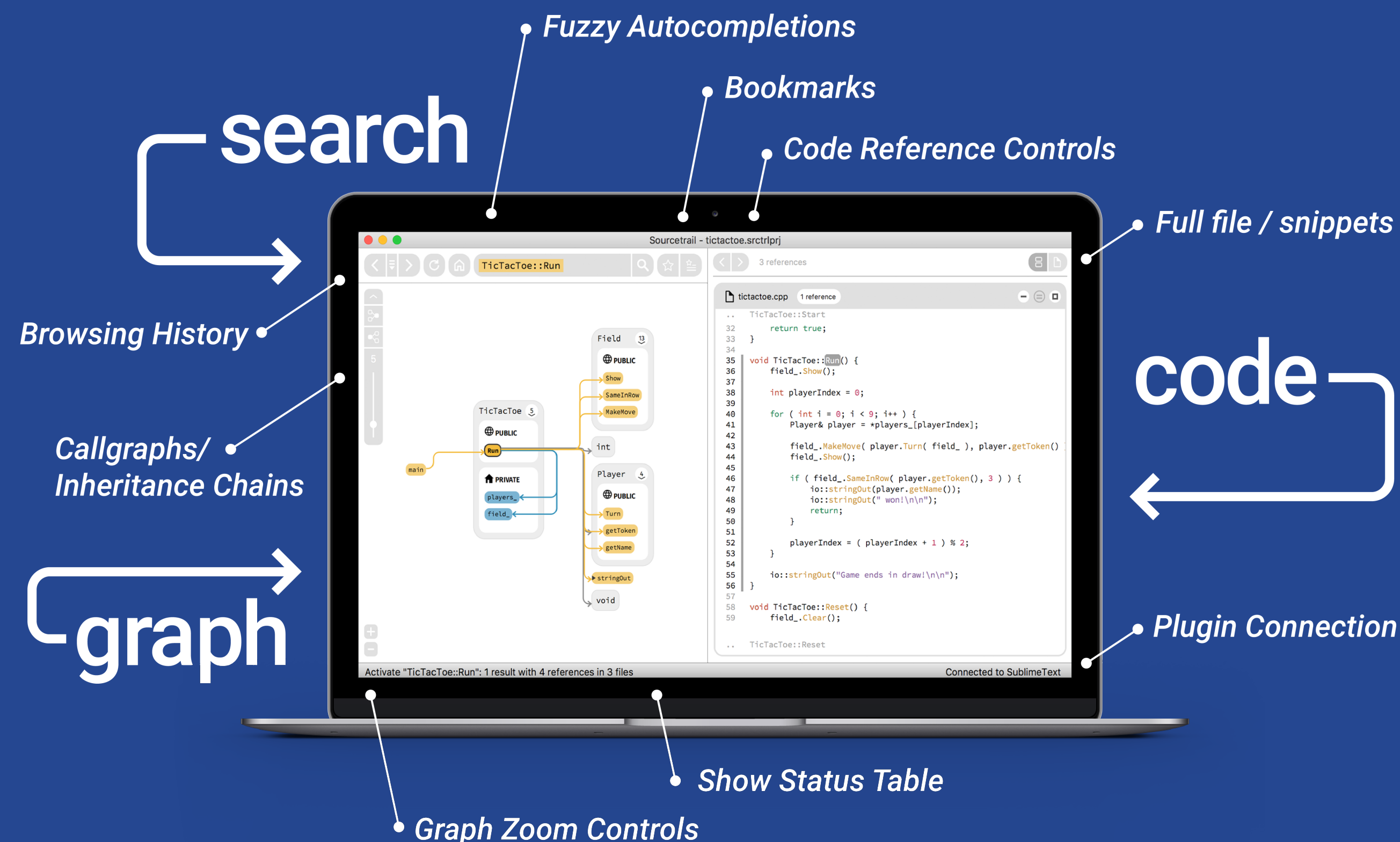
What Programmers Really Do

When a codebase grows, it becomes increasingly complex to keep an overview of the implementation. Peter Hallam[1] estimates that about 78% of a professional software developer's time is spent on reading and understanding source code. Helping software engineers to save time on this task greatly impacts their productivity and as a result the software industry as a whole.

- 2% WRITING CODE
- 20% MODIFYING CODE
- 78% READING CODE

Microsoft | Developer
Peter Hallam's WebLog

[1] https://blogs.msdn.microsoft.com/peterhal/2006/01/04/what-do-programmers-really-do-anyway-aka-part-2-of-the-yardstick-saga/
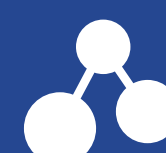

The User Interface for Exploring Source Code

Sourcetrail provides a simple user interface for exploring source code consisting of three components, which together result in an innovative UI concept, that makes Sourcetrail the perfect tool for exploring an unfamiliar code base.

- Fuzzy Autocompletions
- Bookmarks
- Code Reference Controls
- Full file / snippets
- search
- Browsing History
- Callgraphs/ Inheritance Chains
- code
- graph
- Plugin Connection
- Show Status Table
- Graph Zoom Controls

Search
Instantly provides autocompletion results for all classes, functions etc. in the code base.

Graph
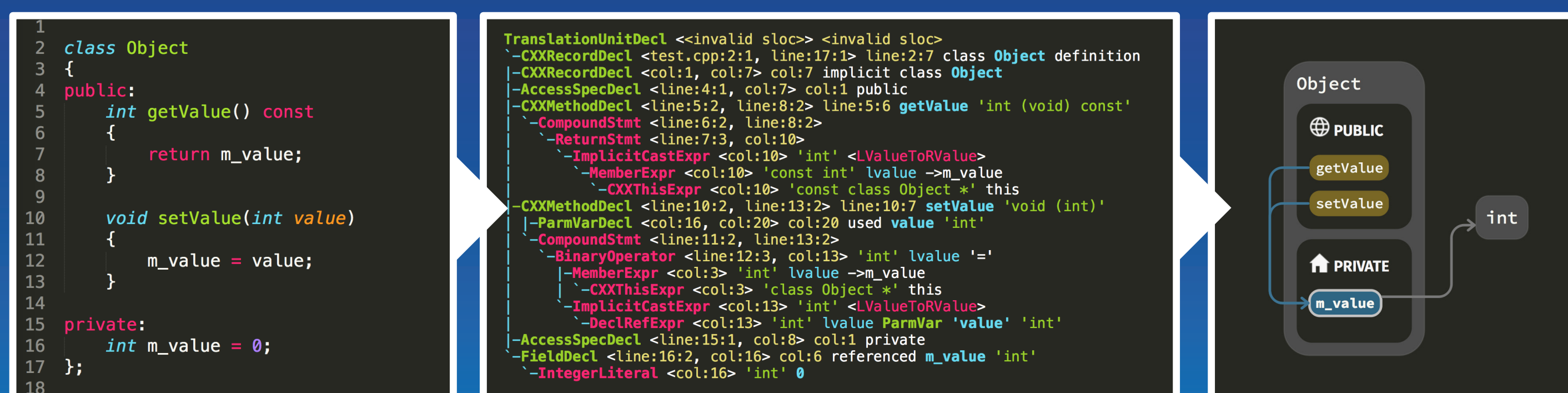Generates an interactive visualization of the currently active symbol and all its relations.

Code
Displays a concise list of all relevant source locations to understand implementation details.

Technical Details

Sourcetrail indexes C/C++ source code by traversing clang libTooling's[1] Abstract Syntax Tree (AST) and gathering data about all symbols and references. Source files are indexed in parallel and stored in a single local Sqlite[2] database file.

Once the project is fully indexed, only changed files need to get re-indexed. Lastly, the user interface is implemented in Qt[3] and quickly generates all visualizations on the fly, offering fully interactive exploration of your codebase.

[1] https://clang.llvm.org/docs/LibTooling.html   [2] https://www.sqlite.org/   [3] https://www.qt.io/

Highlights

- Saves time when reading an unfamiliar codebase.
- The lightweight interface is simple to understand.
- Standalone application that runs on all major platforms.
- Synchronizes via plug-ins with IDEs and code editors.
- Secure to use by functioning fully offline.
- Instant generation of inheritance-, call- and include-graphs.

«I was struggling to understand a huge legacy system my team was tasked with supporting. Sourcetrail (then Coati) helped us untangle the spaghetti code and its dependencies, letting us see what would be affected when we touched each function.»

Nelson Ferrari , USA

«Sourcetrail helps me getting a grip on a large unfamiliar codebase. It gives me a unique perspective on code between the big picture and the gory details which contemporary IDEs don't provide.»

Gilbert Röhrbein , Germany

Conclusion

The work on Sourcetrail started as Master's degree project and continued within the startup Coati Software OG. The user interface was tested and refined during multiple user studies and has proven to significantly reduce time spent on familiarizing with existing source code. After a public Beta until June 2017, Sourcetrail is now available as release version 2017.3 and is already used by hundreds of developers all over the globe.

Visit https://sourcetrail.com to get a free download!

Coati Software OG
Jakob-Haringer-Straße 1/127
5020 Salzburg, Austria
https://sourcetrail.com
mail@coati.io

FH Salzburg   austria wirtschafts service   aws   STARTUP SALZBURG   Visual Studio Partner