Science is a transient discipline. Research priorities change with funding, students come and go, and faculty moves on. What remains are the published and unpublished articles, as well as the data and code they depend upon. Few projects outlive their immediate use, and even projects with thousands of hours of investment grow stagnant and irrelevant, causing confusion and draining time from the following generation of researchers.

However, a moderate investment in maintenance may save many hours of headache down the road. As a case study, I present practical lessons from the recent overhaul of ZMAP v6 to ZMAP7. ZMAP is a set of MATLAB tools driven by a graphical user interface, designed to help seismologists analyze catalog data. It debuted in 1994 as a collection of scripts written in MATLAB 4. The last official update was 7 years later, with a formal release of ZMAP v6 (in 2001). This version was the agglomeration of code written by a host of scientists and scientists-in-training as they tackled their various independent projects. In a way, ZMAP is already a success story, having survived 26 years of alternate development and stagnation, and is still in use around the world. Dozens of research papers have used ZMAP through time, with the 2001 publication having 825 google scholar citations.

With the release of MATLAB R2014b, changes to the graphics engine had rendered ZMAP 6 largely unusable. Over the interim, not only had both the MATLAB language and toolbox changed significantly, but so had common programming practices. The ZMAP7 project started as a “simple” graphical retrofit, but has evolved to leverage modern MATLAB’s new capabilities and updated toolboxes. Targeting a recent version of MATLAB (R2018a) while foregoing backwards language compatibility opened up a wide array of tools for use, and also extended the expected lifetime of ZMAP. A subset of techniques employed follows:

All changes were tracked in Git, providing snapshots and a safety net while avoiding a proliferation of folders. Code was simultaneously changed across the entire project using regular expressions. Variables were renamed according to their purpose. Unreachable files were removed to reduce the maintenance burden. Scripts and global variables were transformed to functions and classes, providing robust error checking and improving readability. Quoted scripts were extracted into functions where MATLAB itself could help evaluate their correctness. Wrapper functions allowed for global behavior changes and instrumentation. Time-consuming activities, such as determining UI placement for dialog boxes were automated.
Though irregular updates still occur, approximately 2000 hours have been devoted to the project, or one year of work by a full-time employee. This time may be amortized through both application speed-up and reliability for researchers across the globe and transparency/reproducibility provided by open-source, version-controlled code. ZMAP7 is hosted on GitHub, where the community is welcome to keep up with the latest developments. More information about Zmap7 can be accessed from the main SED page: http://www.seismo.ethz.ch/en/research-and-teaching/products-software/software/ZMAP/.