Software Sustainability in Remote Sensing?

An interview with Simon Hettrick from the Software Sustainability Institute

What is the Software Sustainability Institute?

We are a national facility for cultivating world-class research with software. What does that mean? We're a group of experts with many years experience of software in research: everything from the nuts and bolts of getting software to work, to the publicity and community development needed to create interest in a research project. We use our expertise to help researchers use and improve software in their research.

So, I'm a busy scientist - why should I care about 'software sustainability'?

It's like asking why should an experimentalist care about their apparatus? If you use software to generate results, then that software becomes a research tool. Sustainable software is reliable. If your research comes under scrutiny, it's easier to show how your results are generated and to prove their repeatability. Sustainable software is easier to build upon. If you want to extend your research, sustainable software is well structured and described so that it's easier to add new functionality.

I don't write any code, I just use commercial software like ArcGIS, ENVI or Erdas Imagine with nice graphical interfaces - surely I don't need to worry about any of this?

Are you sure that you don't? There aren't many research groups who don't use home-grown code of some form: that handy programme written by a Master's student, or even a complex spreadsheet. All of this 'throw away' code could warrant some investigation to understand its sustainability.

And even if you only use commercial software, there are still best practices that can help. What do you do if the commercial software disappears (or if its cost is massively increased)? Can you migrate to another software package? Are there licensing concerns with the use of your software? What do you do with your results? All of these factors come under the umbrella of sustainable software practices.

Ok, so you've convinced me, software sustainability is important - but what should I do about it?

Simply recognising that software is important and that it needs to be managed is the first step!

The next thing is to assess the sustainability of your software. We have a free online software evaluation guide that will help you investigate everything from the reliance of your code on open standards, to the quality of your documentation, to the existence of a community of users. The guide will highlight where your software might need some work, and make recommendations about how to improve it.

If you need help addressing issues raised by the evaluation guide, we have lots of free guides and other resources on our website. You can also look into learning more about sustainable software development through schemes like Software Carpentry. Of course, you can also get in touch with us so that we can provide direct help.

But I haven't got any time or money - what can I do?

I don't think I've ever met a researcher who has time and money to spare! It's a question of how limited resources are invested.

Software sustainability might not be the easiest sell. It's about preparation, long-termism, investing for the future… it's not a particularly sexy topic in that regard. But we believe that investing resources early will pay off in the long run, when your software is easier to maintain, easier to extend, easier to share (if you choose to share it) and has a thriving community built around it.

Sustainability isn't an all or nothing approach. Even a small investment of time – into something like commenting your code, or just learning a bit about sustainable software practices – will pay dividends in the long run.

In other words, even if resources aren't abundant,

SSI Resources

Software sustainability evaluation:
`http://www.software.ac.uk/online-sustainability-evaluation`

Guides:
Including *How to decide if software is 'good'* and *Testing your software*
`http://www.software.ac.uk/resources/guides`

Top tips:
`http://www.software.ac.uk/resources/top-tips`

Software Carpentry:
Training courses in good practice for scientific software development
`http://www.software.ac.uk/software-carpentry`

we suggest investing some time into sustainability. It's only after some time has been invested that researchers will clearly see how sustainability can help their project.

Q Surely I can't influence the sustainability of commercial software - are you saying I should switch everything I do to use free software?

No, that's not a policy of ours. There's a place in this world for free software, and a place for paid-for software. It depends on what's right for your project.

Q I've heard about something called 'reproducible research' - is that something else I should worry about?

I'd argue that reproducible research is the goal of science, so yes it is very important. In relation to software, reproducible research is about being confident in the results your software generates.

Obviously, your software has to be stable enough to repeatedly generate the same results with the same input parameters. But reproducibility is about more than this. The way that the results are generated has to be transparent so that other people can assess your software, and have confidence in the way that it generates results.

Because software is changing all the time, it's vitally important to be clear about which version of software was used to produce which results. Without this information, it can be impossible to verify results. Researchers should clearly version their software and remember to cite the software and its version number whenever they discuss results. (We also suggest that researchers use a repository for their software which will provide them with robust version control.)

Simon Hettrick is responsible for the public face of the Software Sustainability Institute. Simon chairs the Institute's publicity committee, which develops strategy for publicising the Institute's work. Before working for the Software Sustainability Institute, Simon coordinated publicity for OMII-UK and prior to this, he worked in patent law. Simon was awarded his PhD in Laser Physics from the University of Southampton in 2004.

He was interviewed for this article by Robin Wilson, Editor of SENSED.

`s.hettrick@software.ac.uk`