

# Can we trust what we get for free?

Chris George

Software Engineer

formerly

McMaster Centre for Software Certification

United Nations University – International Institute for Software

Technology

# Can we trust what we get for free?

You run your own small business. You need an accounting program, and find:

- **FreeAccount** An open source program hosted on SourceForge.
- **InstaCount** A freeware program, free to download a trial version, \$50 for the full version.
- **ExcelAccount** From a well-known software company, based on Microsoft Excel, \$200 for the basic single user *premium* version, \$400 for the multi-user *professional* version.

All run on Windows, the operating system you use. They all have the functions you need.

Which would you choose?

# Can we trust what we get for free?

You want to build a SWAT model. SWAT is free. You need a model preparation application. You find:

- **QSWAT**, which needs QGIS, which is free and open source.
- **ArcSWAT**, which needs ArcGIS with Spatial Analyst. For ArcSWAT it is possible to get:
  - an unofficial licence for free on the internet;
  - an official licence for personal use from ESRI for \$100 each year.

All run on Windows, the operating system you use. They all have the functions you need.

Which would you choose?

## Is 'free' worthless – or worse?

- Some common sayings:
  - There's no such thing as a free lunch.
  - You get what you pay for.
  - If you're not paying, you're the product.
- “Cheap” usually means poor quality (or poor behaviour: “that was a cheap trick”)
- But what about Wikipedia?
- Two meanings of “free”, as in:
  - free beer
  - free speech, or freedom(Richard Stallman)

# What is Open Source Software?

- Computer software
  - Free (as in free beer)
  - Source code available
  - Licence (as in freedom) to
    - study
    - change
    - distribute
  - Obligation to make changes/extensions also open source
- (some open source licences, especially GNU GPL)

## Warranty of quality?

*THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.*

*(GNU GPL)*

## Can you claim damages if it goes wrong?

*IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.*

*(GNU GPL)*

## Better to buy?

*Microsoft warrants that properly licensed software will perform substantially as described in any Microsoft materials that accompany the software.*

*Microsoft excludes all implied warranties, including those of merchantability, fitness for a particular purpose, and non-infringement.*

*If you have any basis for recovering damages from Microsoft, you can recover only direct damages up to the amount that you paid for the software (or up to \$50 USD if you acquired the software for no charge).*

(Microsoft Windows 8.1 retail licence)

This lasts for 1 year from purchase



# What is Open Source Software?

*open source is an intellectual property destroyer. I can't imagine something that could be worse than this for the software business and the intellectual-property business*

Microsoft executive, 2001

# Characteristics of Open Source Software

- Security
- Affordability
- Transparency
- Perpetuity
- Interoperability
- Flexibility
- Localization

# Examples of Open Source Software

- Linux, Android
- Apache web server
- Mozilla, Firefox, Thunderbird
- OpenOffice, LibreOffice
- GIMP
- PHP, Java, Python
- MySQL, SQLite, PostgreSQL
- CVS, SVN, Git
- GRASS, QGIS, MapWindow
- SWAT
- R

# Open Standards

Data cannot be said to be public if it is held in a proprietary format.

**Open standards**, such as the **OpenDocument** standard (ISO/IEC 26300) for spreadsheets, charts, presentations, and word processing documents

- prevent 'lock-in' to a particular tool or provider
- promote interoperability

The **Open Geospatial Consortium (OGC)** is the body that promotes and controls standards for spatially referenced data.

# Top 4 reasons for choosing open source software

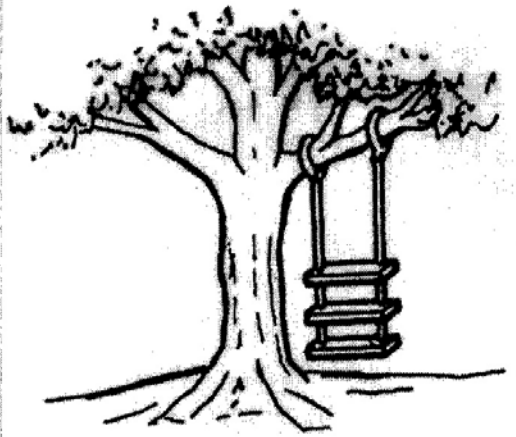
(as provided by the Open Source Business Conference survey)

- Lower cost
- Security
- No vendor 'lock-in'
- Better quality

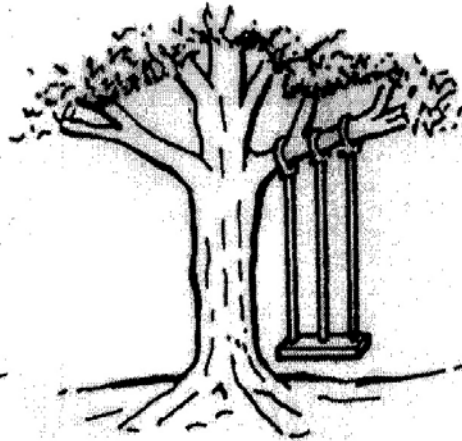
# Risks of open source

- Lack of support
- Poor documentation
- Poor quality code – written by amateurs
- Casual development practices
- Unsustainable

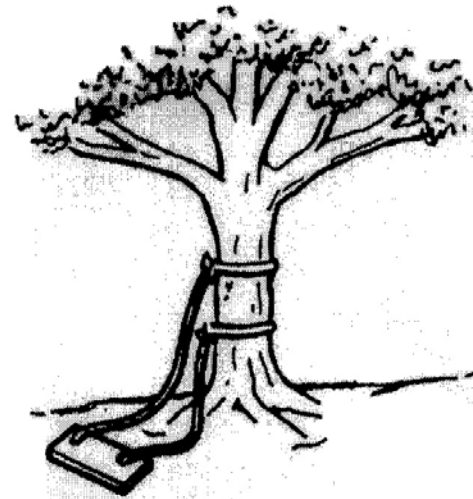
Most software projects fail. Unfortunately, most open source projects fail in public.



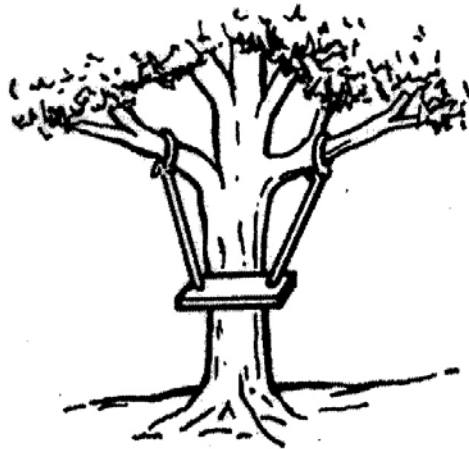
**AS PROPOSED BY THE  
PROJECT SPONSOR**



**AS SPECIFIED IN THE  
PROJECT REQUEST**



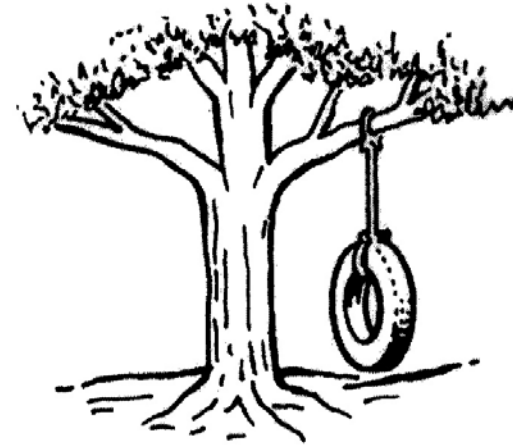
**AS DESIGNED BY THE  
SENIOR ANALYST**



**AS PRODUCED BY  
THE PROGRAMMERS**



**AS INSTALLED AT  
THE USER'S SITE**



**WHAT THE USER WANTED**

7/27

# CRAPL

## Community Research and Academic Programming License

- Created for people who publish science using their own software, who might be embarrassed to show the source.
- *Any appearance of design in the Program is purely coincidental and should not in any way be mistaken for evidence of thoughtful software construction.*
- *"The Author" probably refers to the caffeine-addled graduate student that got the Program to work moments before a submission deadline.*
- *You agree to hold the Author free from shame, embarrassment or ridicule for any hacks, kludges or leaps of faith found within the Program.*
- An open source licence intended to encourage openness and collaboration (with protection unless published).



# How to choose an open source project

- Does it do what you want, on your platform?
- Reputation
- Is it alive?
  - date of last release
  - traffic on user group
- ◆ Documentation

# Open source software in science

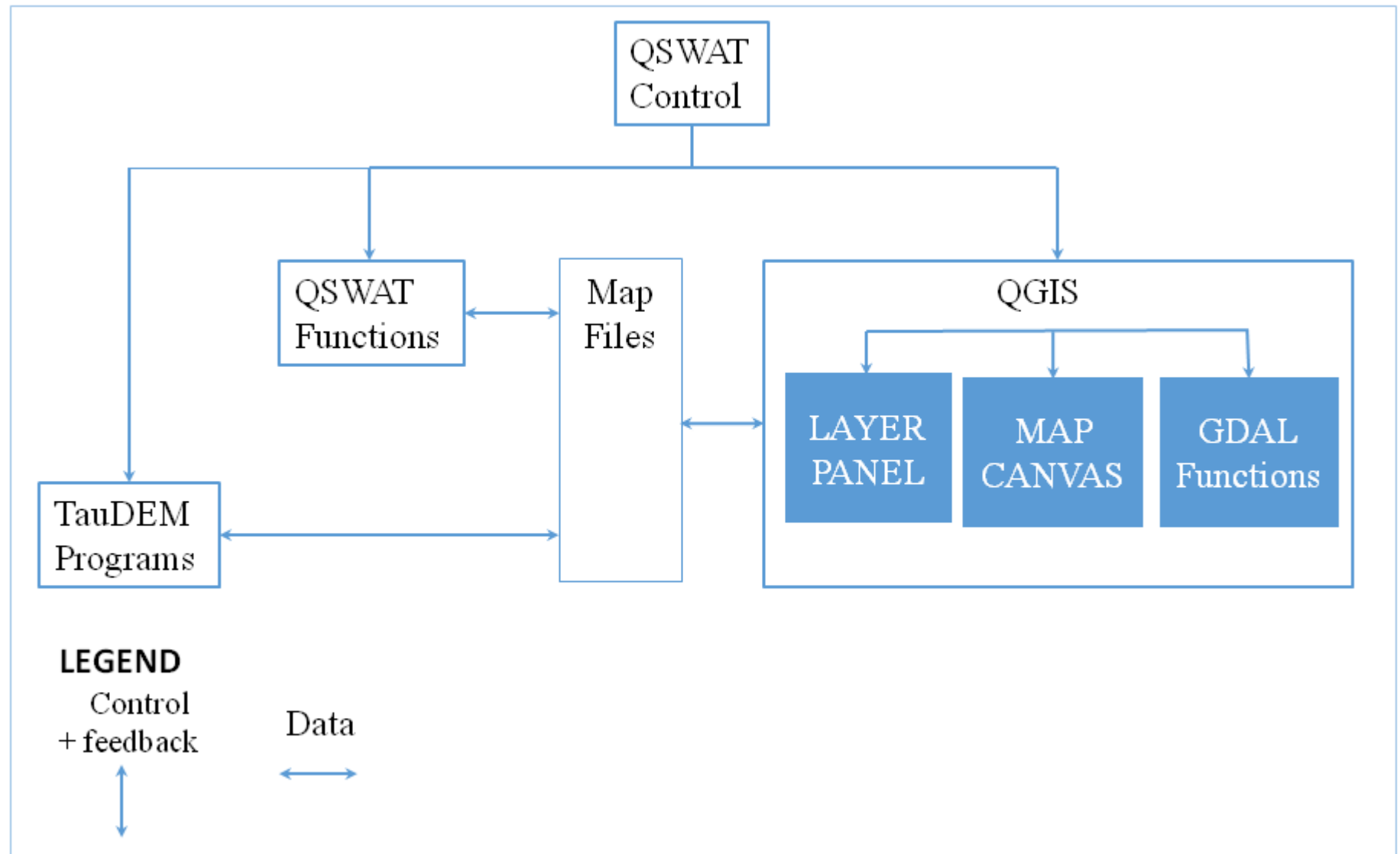
- Scientists (and engineers, and policy makers) depend on software
  - quantity of data
  - analysis of data
  - running models
  - presentation of results
- How do we know the results are correct?
- Normal answer in science is reproducibility. When modelling, calibration+validation is a version of this.
- Common answer in software is multiplicity of users, effectively reputation.

# Reproducibility of results requires the code

Article by Darrel Ince et al in *Nature*, February 2012:

- Computing enables large data sets to be investigated, which makes software of increasing importance.
- Reproducibility requires source code
- Design, algorithm description, pseudocode: insufficient.
- Still problems of hardware, versions of operating system, compiler, and other software.

# QSWAT Structure



# How to develop open source software

- Always copy rather than invent – but acknowledge!
- Use libraries, existing packages.
- Release early, release often.
- Help your users – quickly. Try not to reply *RTFM*.
- Document carefully, and keep it up to date.
- Make code readable rather than clever.
- Good design before speed.
- *Everything should be made as simple as possible, but not simpler* (after Einstein).

# How to complain about open source software

- Can you repeat the problem?
- Has anybody else reported the problem?
- Run the tutorial example, to see if the problem is your example (tutorial works) or your installation (tutorial fails).
- It presumably works for other people. What is different about what you are doing?
- Explain what you were doing, as well as what happened.
- Quote *all* error messages *exactly*.
- If you use a screen image (good idea) try not to obscure other relevant detail!
- Just *It doesn't work* is not helpful.
- *I have to hand my project in tomorrow* suggests a lack of organization

# Where to complain about open source software, or ask for help

- Usually two options:
  - privately:
    - bug reporting page on website
    - email address for reporting bugs or problems
  - publicly: user group
- Is this problem likely to affect other people?
- Will the typical user find it interesting?
- If in doubt, make it private – maintainer can publicise.
- If response is private, asking for information, make your answer private!

## When you get help ...

- Tell the maintainer if it worked! (as well as if it did not work)
- Then the maintainer has more information to improve the software or help other people.
- When others ask for help, and you know the answer

**JOIN IN!!**