Before starting a new software project

A few things to consider

Your new software project awaits you – it will be momentous! A creation of sheer genius fuelled by your ambition, envisioned by your creativity, crafted by your intellect and worthy of praise from the Gods themselves (well... the research councils)! Before you embark on your historic venture, perhaps you need to consider what will befall your invention in the years to come.

Whether for research, administration, learning or teaching, software is an increasingly valuable research tool and output, and needs to be managed as such.

This briefing paper is for researchers who program and anyone who is starting a software-development project. It will help you to understand the sustainability and preservation requirements of your software, the outputs from the development process and how to plan and undertake a software project.

Why is software sustainability important?

Software development projects take up a lot of time and money, and outputs from the development process such as design documents, code, and binaries are valuable assets in their own right. If software is sustained and used for longer, it can have a far greater impact.

Few people think beyond the immediate use of software to consider how it could be exploited further once the project is over. Decisions need to be made on what is to be done with the software and other outputs from the development process. This will ensure that valuable ideas and investment are not lost.

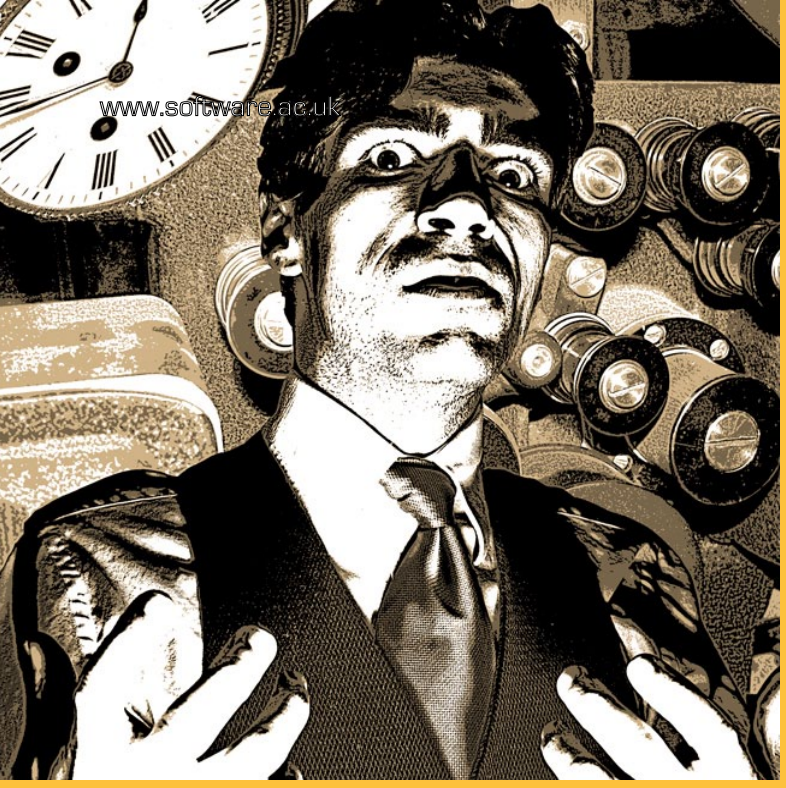
Sometimes you might not need to sustain the software: it might be easily reproducible or the result of a training exercise. Frequently, you will have the opportunity or obligation to sustain or preserve your software for longer.

Why think about sustainability at the beginning of a project?

Once a software-development project get started, it's easy for it to take on the characteristics of supertanker. Changing course can be difficult. It is important to ensure that the destination and direction are right from the off, and that the project isn't trading immediate shortcuts for longer-term expense and hassle. Whilst flexibility during the project is valuable, and agile development methodologies are useful, clarity on what the project is producing and how long the software needs to last for is critical.

The beginning of software development projects is the ideal time to reflect and document sustainability and preservation requirements. Funders of projects increasingly require sustainability and exit plans right from the start of a funding application, and the discipline of considering all potential requirements upfront will help start off the project on the right foot.

Come the end of a project, it is easy to get tied up with other concerns. Unless there is a planned set of activities to sustain the outputs, or the heroic efforts of one committed person, it is common for software to slip into *abandonware*.



It's time to think like a Mad Scientist

When it comes down to it, it turns out that Mad Scientists know a thing or two about bring a project to life.

Your creation will change the world!

Establish expectations. The first thing to consider is what sort of software-development project you will create: it could be a proof-of-concept demonstrator, a pilot or an operational service. This will establish everyone's expectations of robustness and longevity.

Will other aspiring geniuses follow in your pioneering footsteps?

It's time to consider the requirements for sustainability and preservation. Our detailed guidance contains lists of various scenarios where it is advisable to sustain your software, and the benefits of doing so. This guidance can be used to generate requirements:

- If your software will generate publishable research results, you may require greater sustainability so that others can reproduce your results.
- If you're reusing code from external sources, you may need to revert back to earlier versions if IP settlements arise with that code.
- If there is a requirement for an audit function, the software may be required for accessing audit data in the future.

The result will be several clear requirements about the long-term aspects of the desired software, such as 'The software shall be reconstructable by researchers within the field for a period of ten years', 'The software development process shall have a version control system' or 'The audit function shall be accessible by the user for a minimum of five years'.

Probably best listen to those villagers with the pitchforks...

Listen to your requirements and feed them into the design process. Apart from the requirements for sustainability, you will have others for functions, interfaces, performance levels and security/privacy. Prioritisation of these requirements will result in a deeper appreciation of what the project needs to achieve. If you want the software to be around for a long time, then designing for ease of maintenance will be a key objective. Notions of portability, accessibility and testability will also be important.

Maybe it's time to bring in a professional?

Professional developers follow established software development processes: like keeping meticulous documentation, the use of version control and test data. Whilst this appears to a time-consuming overhead at the start of a project, it pays off many times over in the long run and achieves a more reliable outcome. Software engineering is not for everyone, but it should be possible to bring in specialists when required.

Perhaps you can reuse Igor's Solar Powered Pan-combobulation Device™!

The software you want might already exist. Even if it doesn't exist in its entirety, there might be components you could use to save months of development time.

Don't just re-use code, check first that it is reusable. For example, that it has an active support community that can answer your questions and provide updates.

Further information and useful resources

Software Sustainability Institute – resources for software sustainability and preservation:

http://www.software.ac.uk/resources

OSS Watch – many resources for open source software use and development:

http://www.oss-watch.ac.uk

curtis+cartwright

Software Sustainability Institute

JISC

This briefing paper has beevn preprared as part of a JISC-funded project by the Software Sustainability Institute in partnership with Curtis+Cartwright Consulting Ltd. It is part of a series of outputs that have been devveloped to support the HE sector by raising awareness of software sustainability and preservation issues in the software-development process. Images courtesy of blmurch (front) and Glen Edelson (back).