

IN THIS ISSUE:

- Software Sustainability Maturity Model
- Surviving initial project funding
- Top tips for a successful open source project
- FAQs

Online newsletter available at
▶ <http://www.oss-watch.ac.uk/newsletters/september2010.pdf>

Welcome to the first OSS Watch newsletter of the new academic year. Inevitably, things have been a little quieter over the Summer but we still have plenty to tell you about. In this month's featured article, Ross Gardler introduces us to a new Software Sustainability Maturity Model. Ross then goes on to give us a real life example of a project surviving beyond its initial funding in his blog piece. To wrap up, Elizabeth Tatham draws our attention to Damien Katz's blog to provide us with some top tips for a successful open source project.

We hope you enjoy reading our newsletter; as ever please do [get in touch](#) with any comments or questions.

Elena Blanco, Content Editor, OSS Watch ▶ info@oss-watch.ac.uk

News from OSS Watch



Open source projects compete in Demo Cup

The world's best open source projects will come under close scrutiny in an effort to determine which one is the biggest game changer in the market. In a competition held on 1 October, the 13 finalists in the Demo Cup, organised by the Open World Forum, will be assessed on their innovative and open nature and the impact they might have on their respective markets. Projects will also have eight minutes to persuade the jury of their worthiness. The finalists are ActiveEon; Disruptive Innovations; Conecta Research; Hedera Technology; iceScrum; Jaspersoft; Mozilla; Obeo; Pentaho; O Engine; Pilot Systems; Talend and XWiki.

▶ <http://www.networkworld.com/news/2010/090610-open-source-projects-under.html?hpg1=bn>

Solar-powered open-source cell phone system tested at festival

The open source project OpenBTS has been used in combination with Asterisk to build a low-cost, solar-powered cell phone network. Tested at the recent Burning Man festival in the Black Rock Desert of Nevada, the network could be deployed and operated at substantially lower cost than existing technologies in greenfields in the developing world.

▶ <http://www.networkworld.com/news/2010/083010-open-source-voip-cell-phones-at-burning-man.html>

Google announces 'Wave in a box' project

Google has announced the 'Wave in a box' project, a stand-alone and functional open source version of its recently cancelled Wave real-time collaboration project. The company says that it has already open sourced 200,000 lines of code as part of the development of the Wave protocol, and plans to expand on the example server that it released to create a 'more complete' application. The move came, in part, in response to Wave developer community requests in the Wave protocol forums.

▶ <http://www.h-online.com/open/news/item/Google-Wave-s-open-source-future-in-a-Box-1071917.html>

Investment in open source software set to rise, Accenture survey finds

The open source software market has reached a turning point, with organisations in the US, UK and Ireland now committing to clear strategies and policies for open source software development. According to the findings of a survey released by Accenture, more than two-thirds of organisations anticipate increased investment in 2010, with more than a third expecting to migrate mission-critical software to open source in the next 12 months.

▶ http://www.marketwatch.com/story/investment-in-open-source-software-set-to-rise-accenture-survey-finds-2010-08-05?reflink=MW_news_stmp

Breaking down barriers for women in open source

The Free Software Foundation (FSF) is acting to implement strategies to broaden the participation of women in the FOSS community. Recommendations for removing barriers and broadening membership among women in open source projects were published recently by the FSF's Women's Caucus, tasked to devise solutions to address the problem.

▶ http://www.sdtimes.com/BREAKING_DOWN_BARRIERS_FOR_WOMEN_IN_OPEN_SOURCE/By_David_Worthington/About_OPENSOURCE/34544

Software Sustainability Maturity Model

When choosing software for procurement or development reuse you need to consider the future. While a software product may satisfy today's needs, will it satisfy tomorrow's needs? Will the supplier still be around in five years' time? Ross Gardler presents an outline of a new 'Software Sustainability Maturity Model', which can be used to formally evaluate both open and closed source software with respect to its sustainability.

▶ <http://www.oss-watch.ac.uk/resources/ssmm.xml>

▶ OSS Watch online

▶ OSS Watch blog

▶ Contact OSS Watch

▶ OSS Watch twitter

OSS Watch is funded by the
Joint Information
Systems Committee

Software Sustainability Maturity Model

Full article can be found at <http://www.oss-watch.ac.uk/resources/ssmm.xml>

When choosing software for procurement or development reuse - regardless of the licence and development model you will use - you need to consider the future. While a software product may satisfy today's needs, will it satisfy tomorrow's needs? Will the supplier still be around in five years' time? Will the supplier still care for all its customers in five years' time? Will the supplier be responsive to bug reports and feature requests? In other words, is the software sustainable?

When evaluating closed source software, much of this is guesswork. Open source software, on the other hand, presents a number of [advantages](#) over closed source, such as options for the ongoing maintenance and development of local software configurations. This can reduce the risks presented by supplier failure. But it can also complicate product evaluation, since newcomers to [open source development methodologies](#) may find it difficult to evaluate the software in terms of its sustainability.

This document presents an outline of a new 'Software Sustainability Maturity Model' (SSMM), which can be used to formally evaluate both open and closed source software with respect to its sustainability. The model provides a means of estimating the risks associated with adopting a given solution. It is useful for those procuring software solutions for implementation and/or customisation, as well as for reuse in new software products. It is also useful for project leaders and developers, as it enables them to identify areas of concern, with respect to sustainability, within their projects.

1. Some useful definitions

Before we examine existing techniques for measuring software sustainability, let's define some important terms as they are used in this context.

Software reuse is the re-application of knowledge encapsulated in software code in order to reduce the effort of developing and maintaining a new software system. Even when a complete software application cannot be reused individual components, data formats, high-level designs, algorithms or other items may still be reusable.

Black-box reuse is the reuse of a component without the need to understand its inner workings. That is, one simply makes calls on the component and receives the results. Black-box reuse can be achieved only when the component performs exactly as required, as it is not possible to modify the component's behaviour.

White-box reuse is the reuse of a component after some internal modifications have been carried out on the component. For white-box reuse, it is not necessary for the component to perform precisely as required in the new environment. However, the component must be well designed and documented, and licensed to allow for modification.

2. Factors affecting software sustainability

The sustainability of software is affected by both technical and non-technical issues. Technical issues tend to focus on how reusable the software is - i.e. its potential for adaptation - while non-technical issues include how a project is [governed](#) and funded. Often it is not possible to cleanly separate the technical issues from the non-technical issues.

To be sustained over time, software needs to be both useful and adaptable. It also needs to evolve as the users' needs evolve. The potential for reuse is therefore a key factor in the sustainability of software. Reuse can save

time and money, and increase the reliability of resulting products. However, an attempt to reuse software that is not easily reusable can have the reverse effect. The goal of the Software Sustainability Maturity Model is to provide a means of evaluating the risk factors in reusing software.

Software sustainability is also affected by the number of environments in which the software is likely to be used and reused. For example, if a tool is applicable to only a small number of low-value users, it is less likely that sufficient resources will be available for the ongoing support and development of the product.

It is not normally good practice to attempt to create a software solution that seeks to satisfy too many disparate groups of users. This is because it is difficult to address the needs of all potential users without having to make significant compromises. It is therefore becoming increasingly common for suppliers to share core components of a software system and then focus on a specific group of user needs at higher levels of abstraction, such as the user interfaces. For example, consider that numerous Internet search engines, social networks and micro-blogging services all collaborate on the same core data management frameworks. In these cases organisations that often compete in the marketplace are able to collaborate on overlapping technology, while still developing competitive advantages in their specific market. Often called [open innovation](#), this approach is discussed in more detail in [this video presentation](#).

There are many examples of software code being reused, especially if we look to open source software for inspiration. Reusable software components range from small libraries, such as those in the Apache Commons project, through to frameworks for building complete applications, such as the Eclipse Rich Client Platform. There are also complete applications that can be extended through plug-in systems, such as Moodle, Drupal or the Mozilla Firefox web browser

The sustainability of software is affected by both technical and non-technical issues.

▶ OSS Watch online

▶ OSS Watch blog

▶ Contact OSS Watch

▶ OSS Watch twitter

OSS Watch is funded by the
Joint Information
Systems Committee

JISC

▶ Article continues at: <http://www.oss-watch.ac.uk/resources/ssmm.xml>

▶ OSS Watch online
<http://www.oss-watch.ac.uk>

▶ OSS Watch blog
<http://osswatch.jiscinvolve.org>

▶ Contact OSS Watch
info@oss-watch.ac.uk

▶ OSS Watch twitter
<http://twitter.com/osswatch>



Surviving initial project funding

Published by Ross Gardler on September 6, 2010

STAY UP-TO-DATE

▶ OSS Watch blog feed

▶ <http://osswatch.jiscinvolve.org/feed>

We've often claimed that [opening up development](#) of a project can help in its [long term sustainability](#). By allowing new funders and participants to take an active role, even leadership, in a project it is possible to survive the natural coming and going of project participants.

Today I added the following update to the OSS Watch [sustainability case study on Apache Cocoon](#):

“Activity on the project has slowed considerably since its heyday. However, development continues despite the departure of a significant number of community leaders. It can therefore be argued that Cocoon validates the community model of software development as described in this document.”

“**Nobody can call it perfect but it is certainly useful.**”

Whilst Apache Cocoon is considered to have [changed computing](#) it is not necessary to have such a profile in order to take advantage of sustainability through openness. Even niche projects from the academic sector can be valuable case studies, for example, lets consider the JISC TechDis Toolbar.

Steve Lee, our accessibility expert, has been working with the [team](#) at the University of Southampton to open up a cross browser [ToolBar](#) designed to help make the web more accessible. It's a great project that allows users to control the way a page is displayed, invoke a text to speech reader, spell check editable content, look up dictionary definitions and extract reference information (amongst other things). Although the tool is an accessibility tool many of its features are of much more general use, [Lifehacker](#) said the work brought *“something*

long overdue for web users.”

Steve helped the team open source the project and tried to work with TechDis to explain the benefits of collaborative development, in particular the ability to spread the cost (and risk) of development across multiple partners. Steve spoke about this with the [H Online](#) at our [TransferSummit](#) back in June:

“Lee told The H that the tool, developed as an open collaboration between JISC TechDis and University of Southampton's School of Electronics and Computer Science, was created to replace a previous toolbar... Lee said the open development process... has allowed the project to be more sustainable.”

With the support of both TechDis and Southampton the Toolbar has been getting plenty of attention and [use](#). Nobody can call it perfect but it is certainly useful. Furthermore, since it is open source others can help improve it.

Despite the success of the Toolbar in terms of raw use figures Sal Cooke, Director of Techdis, recently [announced](#) the demise of the Toolbar. She said that TechDis were *“delighted by the response and the positive feedback we've had from users”* and that the *“number of downloads has surpassed all expectations.”* So why kill the project?

[Ed: In the full post Ross goes on to explain that although TechDis have canned their Toolbar project the software lives on thanks to the Southampton team]

▶ <http://osswatch.jiscinvolve.org/wp/2010/09/06/surviving-initial-project-funding/>



Top tips for a successful open source project

Published by Elizabeth Tatham on August 15, 2010

Damien Katz, whose Apache CouchDB recently hit 1.0, provides some excellent tips on creating a successful open source project in his blog [Getting your open source project to 1.0](#). Drawn from five years' experience, the tips include general advice interwoven with examples from the project. He begins with the fundamental question, Why?, explaining that a successful project needs a reason for being – a clear idea of what problem it solves – and you need to figure this out and explain it.

Almost as important as knowing what you are is knowing what you're not: *‘Stating clearly what your project isn't trying to do or be helps make it much easier to explain what you can't implement or change... and to focus on what you actually are.’* Next, he advises, *‘don't expect to attract anyone to your project until you have a substantial amount of working code that isn't a big ball of spaghetti’*. Code comes first, but don't try to do everything (well), as you'll probably never actually release anything: *‘You'll need to pick a few things that you do really well and execute on those things.’*

“**We can help you create a successful open source project by providing advice every step of the way.**”

On the subject of community, Katz encourages you to *‘make sure the people who show a strong desire to contribute aren't ignored, and feel like their efforts will eventually amount to something’*. But bear in mind, he warns, that community is often incompetent. You will sometimes need to hurt people's feelings for the sake of the project because *‘the quality of the community is more important than its absolute size’*. *‘Our committers,’* he stresses, *‘are our first line of defence against poor code and design.’*

In the end, though, it's up to you to use your brain and *‘figure out what's actually important to you, your project and its community... Projects can't follow cookie cutter rules.’*

One tip that could be added to this list is to [contact OSS Watch](#). We can help you create a successful open source project by providing advice every step of the way. In addition, our briefing documents offer invaluable information on everything a new project needs to consider, including [governance models](#), [sustainability](#), [how to build an open source community](#) and [licensing](#).

▶ <http://osswatch.jiscinvolve.org/wp/2010/08/15/top-tips-for-a-successful-open-source-project/>

▶ OSS Watch online
<http://www.osswatch.ac.uk>

▶ OSS Watch blog
<http://osswatch.jiscinvolve.org>

▶ Contact OSS Watch
info@osswatch.ac.uk

▶ OSS Watch twitter
<http://twitter.com/osswatch>

Events



Sept-Oct **30-1** Open World Forum, Paris, 30 Sep - 1 Oct 2010

The Open World Forum will take place in Paris 30 Sep - 1 Oct 2010 with a theme of 'Open is the Future: Open Innovation, Open Enterprise, Open Society'. The conference promises to be a worldwide hub for open innovation with 36 keynotes, workshops and think-tanks. The event is free to free and open source stakeholders and registration is now open.

▶ <http://www.openworldforum.org/>

Oct **19** JISC event: The future of Research? - London, 19 October 2010

JISC is running a one day conference entitled 'The future of Research?' at the Congress Centre in London on 19 October 2010. The event will examine three major challenges facing education institutions involved in research today: institutional reputation; efficiency and effectiveness; and collaboration in a competitive environment. The programme is aimed at vice chancellors, pro vice chancellors and senior managers in education institutions involved in research today.

▶ <http://www.jisc.ac.uk/Home/events/2010/10/futureofresearch.aspx>

Nov **10-12** NGS Innovation Forum '10, 23-24 November 2010: call for abstracts

The NGS Innovation Forum will be held at STFC Rutherford Appleton Laboratory on 23-24 November 2010. The two-day event will showcase the impact that the NGS has had on research in the UK, allow delegates to find out more about using the NGS in applied research and enable IT staff to find out how their institution can benefit from the NGS. There will also be a poster session featuring peer-reviewed posters from NGS users. The deadline for abstracts is 10 September 2010.

▶ <http://www.ngs.ac.uk/content/ngs-innovation-forum-10-poster-submission>

Frequently Asked Questions

▶ OSS Watch online

▶ OSS Watch blog

▶ Contact OSS Watch

▶ OSS Watch twitter

OSS Watch is funded by the
Joint Information
Systems Committee

JISC

Q Would you publicise my event or news about my project?

A Provided your news or event is relevant to open source software and the educational sector in the UK, we are delighted to add your event or a link to your news item on our [news](#) or [events](#) page. Please [contact us](#) with your request.

Q Can I use an open content licence on my code?

A We would advise against it very strongly. Open content licences applied to executable software make no requirement for the corresponding source code to be made available. Open content licences applied to source code do not require any executables built from the source to have their source published. All in all open content licences are poorly adapted to the special circumstances that surround the making and changing of computer software.

Find answers to your questions at: <http://www.oss-watch.ac.uk/about/faq.xml>

▶ OSS Watch online
<http://www.oss-watch.ac.uk>

▶ OSS Watch blog
<http://osswatch.jiscinvolve.org>

▶ Contact OSS Watch
info@oss-watch.ac.uk

▶ OSS Watch twitter
<http://twitter.com/osswatch>