SOFTWARE REUSE, REPURPOSING AND REPRODUCIBILITY PHASE II REPORT

Ian Gent, John McDermott, Chi-Jui Wu University of St Andrews;
Catherine Jones, Brian Matthews, Paulina Lach, Steven Lamerton,
STFC and Jonathan Tedds University of Leicester

December 2015

Contents

1. Introduction

This report builds on the work done in Phase I in which we aimed to *"consider issues of software reuse and identification. It will start by considering the issues pertaining to persistent identification of software and how particular pieces of computational research software may not only be identified but kept in a runnable state."*

In the second phase of the Software Reuse, Repurposing and Reproducibility project we aimed to:

- Gather further feedback about the Guidelines for Persistently Identifying Software
- Technical development on
    - Building VMs from GitHub repositories
    - Comparing the use of Docker[1] and Vagrant for scientific software created at STFC
    - Considering how to provision the above containers for wider reuse
- An institutional case study, outside the original partners, to assess the guidance and tools in a wider context and build on their developments in this field.
- A design/prototype implementation of the vision of identifying and linking different aspects of the software lifecycle.

1.1. Project Partner Background and changes

There have been two changes with the project make-up and responsibilities within Phase II. The first is that the University of Leicester has joined the consortium to provide an institutional case study around an individually tailored piece of open source software for the Cancer Research Biobank. The second is that Catherine Jones is now the Group Leader of the Software Engineering Group at STFC. One of the services run by her group is the EPSRC funded Software Engineering Support Centre which provides software management tools and advice. Work in the area of this project is being followed with a view to potential cross-over into future services.

2. Technical Progress

During the summer of 2015, an intern at St Andrews and an Erasmus student at STFC worked on technical developments in support of this project. They were supervised by Ian Gent and Steven Lamerton respectively. The full reports are appendices of this document. Key achievements and findings are described below

2.1. St Andrews

The goal at St Andrews was to see if we could simplify the process of creating Virtual Machines containing research software. After initial investigation we decided that an appropriate route was to try to leverage GitHub and Travis CI. Our intern Chi-Jui Wu created a tool called "Recompute".[2]

[1] https://www.docker.com/

[2] https://github.com/cjw-charleswu/Recompute

From the client side this is a simple web form in which the GitHub URL of a software project is submitted. On successful completion, two new links are provided. One is a download for a vagrant box which should run in any OS with Vagrant[3] and VirtualBox[4] installed, and which contains the software. The other is a "play button", which opens a terminal window into that machine running on our server: using this no software at the client side is necessary apart from the browser, and using a javascript terminal emulator, the user can interact with the software through their browser window.

From the server side, the software is written in Python and in most cases works for GitHub[5] projects which have a Travis CI configuration file available. Travis CI[6] is a tool which does free test builds of GitHub projects, so is widely adopted. However it does not supply users the capability to download the resulting VMs. Accordingly Wu's software scrapes the configuration file provided by the user to work out which packages and dependencies need to be installed and how the software should be built. From that a VM is created using Vagrant and VirtualBox, and a build attempted. If it succeeds the resulting VM can be created for download by the user and/or playing in the browser.

This works for a limited set of languages (including Python, node.js, C++) and is likely to work only if the GitHub contains a Travis configuration file. As a proof-of-concept however, it shows that we can go direct from GitHub to executable and preservable VMs which can even be played in the browser.

2.2. STFC

The goal of this task was to see if the recomputation ideas produced by St Andrews could be transferred to a different organisation and a different scientific domain. In addition we investigated two technical solutions: Docker and Vagrant.

The software chosen is Mantid [7] which is an Open Source project which STFC is one of the main contributors. It has a strong software development process. It is used to manipulate and analyse neutron scattering and muon spectroscopy data generated by ISIS[8] the neutron spallation source at RAL.

The student was able to use both Docker and Vagrant to package the software, which is an achievement as she has not been involved in the development process.

Her main conclusions about these solutions were:

- Both are open source projects with good-quality documentation and large communities
- Docker does not virtualise an entire operating system, unlike Vagrant, so it is generally more lightweight in terms of system requirements and quicker to start

[3] https://www.vagrantup.com/
[4] https://www.virtualbox.org/
[5] https://github.com/
[6] https://travis-ci.org/
[7] http://www.mantidproject.org/Main_Page
[8] www.isis.stfc.ac.uk/

- Vagrant works well on both Windows and Linux, whereas Docker runs only on Linux and cannot easily run graphical applications.
- It is hard to decide explicitly which approach is better and the choice depends on a number of requirements. In my opinion, if no graphical items should be preserved, it is better to use Docker, because it is lighter and may be faster. Nevertheless, if there is a need to maintain graphical applications, Vagrant should be used.

3. Community Building Activities

The team attended a number of events to increase the visibility of the project and the guidelines in particular.

- **Software Sustainability Institute Workshop on Giving Credit to Software.**
  - Ian Gent spoke about the project
  - In the workshop sessions afterwards the model of software entities described in the guidelines were discussed with Martin Fenner of DataCite and Geoffrey Bild from CrossRef.
  - We agreed to write a SSI blog in this area (work in progress)
- **Poster at iPres 2015**
  - The project had a poster at iPres2015 on persistently identifying software. There was much interest generate and the team are following up contacts with MIT and John Hopkins University
- **Outreach within the EPSRC funded Collaborative Computational Projects**
  - In Catherine's new role as leader of the EPSRC funded Software Engineering Support Centre, she has been meeting research groups involved in CCPs and bringing up issues of persistent identification of software and reproducibility to gauge interest in the computational science community.
- **Discussions with British Library/DataCite re implementation of DOIs for Software**
  - There has been a technical meeting between Rachael Kotarski and the Software Engineering Support Centre about the practicalities of providing DOIs for software and the potential interactions/synergies with outputs from Continuous Integration.
- **Discussion with Microsoft Azure for Research on software container hosting**
  - A meeting took place with Kenji Takeda, academic engagement lead for MS Azure for Research to obtain agreement in principle to provide free software container hosting for UK based academics via the Azure for Research Platform, building on previous work by the BRISSKit project deploying Puppet provisioning.
- **Described as part of the UK landscape in the Knowledge Exchange workshop on Software Sustainability**
  - Brian Matthews was one of the participants in a Knowledge Exchange workshop and highlighted this work as part of the UK efforts in this area

4. Case Study

The Cancer Research Biobank at the University of Leicester - http://www2.le.ac.uk/partnership/lcrc/facilities/cancer-biobank - worked with the BRISSKit project – http://www.brisskit.le.ac.uk to professionally tailor and document an instance of the OpenSpecimen http://www.openspecimen.org/ software for active sample data management during 2013-14. The Biobank plans to continue to use the software in a range of future research projects. Rather than rely on individual(s) project funded researchers, having varying technical experience, to curate and update the relevant software, the Biobank needs to preserve the professionally tailored instance of the previous open source build in order to reuse in a range of future projects.

This use case highlights the key importance for the many researchers who need to reuse existing research software that they may not have written or adjusted. Instances of such software are not normally preserved within a group or at institutional level between projects without a unique business case to do so at institutional level. Hence making an instance available via a neutral platform such as Microsoft Azure for Research takes on critical importance if researchers are to be able to sustain, reuse and gain continued credit for modified open source software. It may not be enough to simply deposit versioned code in a Github repository (although this was also done here) since this does not allow simple redeployment as will often be desired without the involvement of a more technically proficient researcher or support resource.

For the use case we have made the tailored instance of caTissue available via Azure for Research as a proof of concept as well as pointing to the github deposit. We also provided detailed description of the provenance, licensing and context for the software according to the project guidelines and this was found to be a very useful resource in guiding researchers in making the code available as desired. It is important to note that the use case also highlighted a limitation with this approach in that while other groups would be able to benefit, the cancer group itself desired to use a new set of functionality included in the latest world release of OpenSpecimen and had decided to proceed in configuring their requirements further against the new instance even if it required reworking some of their previous configuration work.

5. Observations

5.1. Stakeholder views

In the interactions and outreach the project has undertaken, it is obvious that there are three main stakeholders in this area and they take slightly different stances on persistent identification and the benefits of preserving versions of software in a runnable form.

- Research software engineers
- Computational scientists who write code
- Digital Preservation experts

The following observations are based on the interactions undertaken and my not be representative of the views of all members of the stakeholder groups.

On the whole Research Software Engineers, who follow good software engineering practice, feel that there is no need for the long term maintenance or preservation of runnable versions of

software as these can be generated again from the existing code repository and they aren't convinced of the benefits of putting in place further systems. This may be because, on the whole, they are professional software developers and may not use their software and so are unlikely to need the provenance trail. It may also be that they are pragmatic people and believe if the code is important enough, then it will be re-implemented following technology shifts.

There has been a mixed reaction from computational scientists. The idea that persistently identifying code would lead to better acknowledgement was well received by some we spoke to. The importance of reproducibility of the scientific results from software was noted by a couple of people.

Digital Preservation experts are interested in the methods and issues around preserving computer software, both as an artefact in its own right and as a tool to enable the preservation and use of other digital objects. It was identified as a trend for the coming year at iPres 2015.

So to summarise, the further from the creation of the code, the greater the interest in preserving it is. The idea of being able to prove reproducibility of results from software analysis is gaining traction but this is still an area which is novel to many we spoke to.

5.2. Technical Overlaps

There are overlaps with building vagrant or docker containers from a software repository and techniques used in Continuous Integration tools, such as Jenkins or deployment processes such as Puppet (as used at Leicester) or Chef. Ways of linking CI outputs and the reproducibility activity is an area SESC is considering as it rolls out the SESC Build Service. However there are differences between testing and production running, both in scale and in software license conditions.

6. Next steps

We propose three strands of activity in the final phase:

- **Code in active development:** Work on leveraging the software management tools in use in good software management practice to enable the persistent identification of executable code in furtherance of reproducible science
    - STFC will prototype the integration of persistent identifiers of software and the production of appropriate containers such as Docker and/or Vagrant virtual machines into the build process. This will leverage existing tools such as Jenkins and Travis continuous integration. This will be prototyped within the Software Engineering Support Centre's evolving Build Service infrastructure, based on Jenkins.
    - St Andrews will lead a case study of using the above for research code under active development by an individual researcher or small group
    - Leicester will advance generic hosting container build (using Puppet) on MS Azure for Research Platform so that resources are available to any UK based academic.

- **Preservation of code which has ceased active development:** Work on how these techniques can be used for preserving code which is no longer being worked on to ensure the software has a useful life beyond the point development ceases.
    - Leicester will utilise these techniques for 3 bespoke, tailored applications of BRISSKit software components including Cancer Biobank (use of OpenSpecimen); NIHR Leicester-Loughborough Physical Activity BRU (CiviCRM); 100k Genomes (CiviCRM / OpenClinica) and make code available via nationally available repositories.
- **General outreach within the community**, including continued activities within Force11 working group on software citation.

7. Conclusions

Technical progress was made in this phase of the project and the output from the first phase has been publicised. Outreach activity, both directly related to this project, or aligned to it by the project member has demonstrated that this activity is of interest but not well enough defined for a service to be ready to be established although in certain stakeholder groups the importance of this work, and the wider area in general, is acknowledged.