Research Computing RSE for the Humanities and Social Sciences

Steve Ranford, Senior Research Software Engineer, Research Computing

31/10/2023

Updates

- We've moved from IDG to the RTP department as part of Research Computing as of 1st August 2023.

- Currently recruiting replacement for James Tripp who left in October.
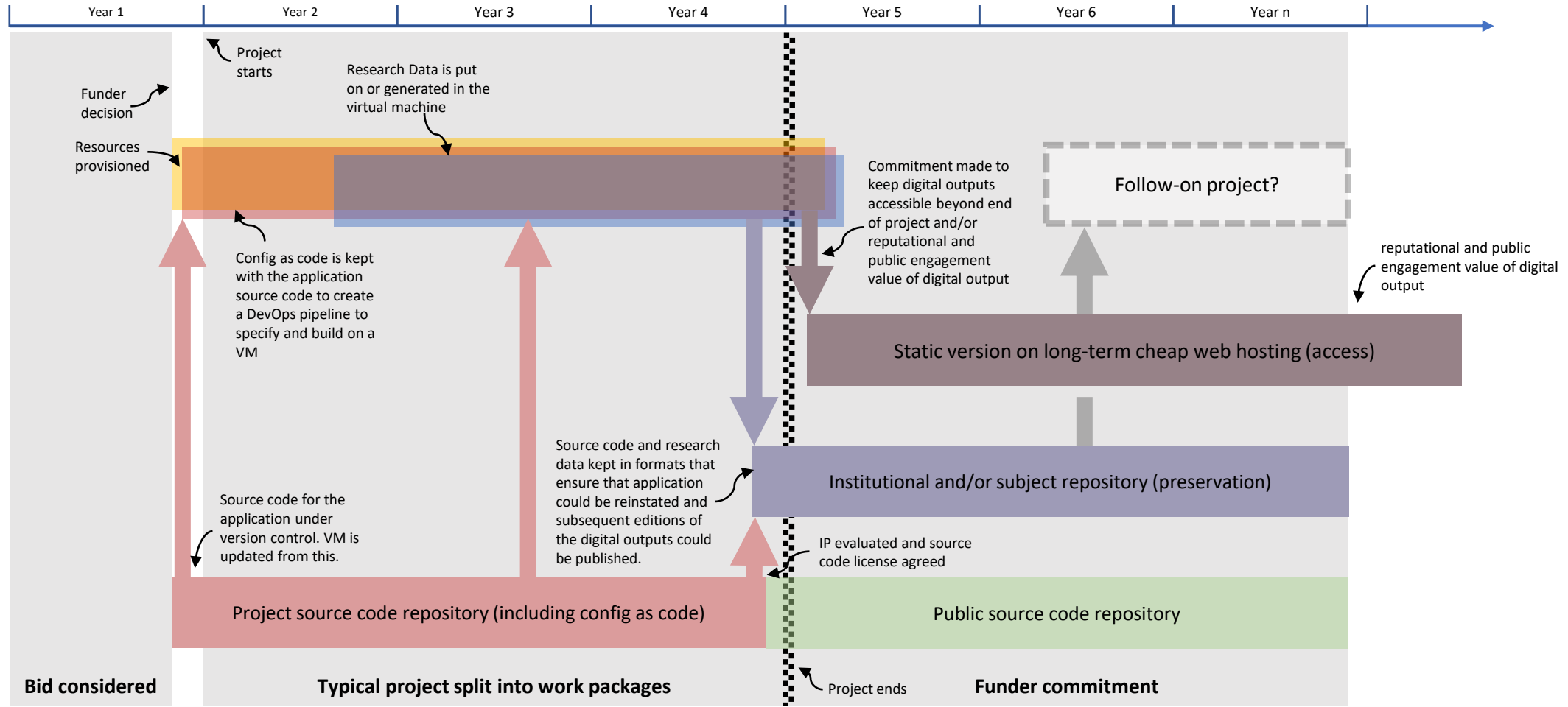
Some Context

- 14 live instances of Omeka Classic (LAMP) with custom theming
- 1 instance of Omeka Classic turned into a served static site
- All instances working with open research data
- Source code between gitlab/github
- VMs managed via Ansible
- A trend towards static sites from the start or static ASAP.

Humanities Research Software Lifecycle

Virtual Machine

Source code

Research Data

Year 1 | Year 2 | Year 3 | Year 4 | Year 5 | Year 6 | Year n

Project starts

Funder decision

Resources provisioned

Research Data is put on or generated in the virtual machine

Config as code is kept with the application source code to create a DevOps pipeline to specify and build on a VM

Source code for the application under version control. VM is updated from this.

Source code and research data kept in formats that ensure that application could be reinstated and subsequent editions of the digital outputs could be published.

Commitment made to keep digital outputs accessible beyond end of project and/or reputational and public engagement value of digital output

Follow-on project?

reputational and public engagement value of digital output

Static version on long-term cheap web hosting (access)

Institutional and/or subject repository (preservation)

IP evaluated and source code license agreed

Project source code repository (including config as code)

Public source code repository

Bid considered

Typical project split into work packages

Project ends

Funder commitment

Current focus

Containerisation first (docker-compose)

- Better application patching and update management
- Improved reproducibility and quality of archive package
- Flexibility development environments
- Integrated with DevOps pipelines with Github Actions

Create a Codespace per project

- Low barrier to entry to web development environments for the non-docker using Humanities scholar
- Can publish to a publically accessible URL for on-demand testing which reduces our VM count and environmental footprint
- Collaboration opportunities both external and remote pair coding via 'live share'

Use of Codespaces

- Using it with open research data/test data
- Integration into VS Code is very pleasing.
- Can be slow unless using prebuild
- A route for allowing people to play with code rejected by the ITS Linux Hosting
- Spent time on configuring the codespace via devcontainer.json to bring the application up on-load.
- Education license gives you access to Github Pro.
- Codespaces is expensive. Pro account doubles the free tier provision. Not enough to use it as primary development environment.

Codespaces (github.com)