Introduction to the Minitrack Software Survivability: Strategies for Long-Lasting and Usable Software

Maytal Dahan
Texas Advanced Computing Center,
The University of Texas at Austin
maytal@tacc.utexas.edu

Joe Stubbs
Texas Advanced Computing Center,
The University of Texas at Austin
jstubbs@tacc.utexas.edu

Sandra Gesing
Center for Research Computing
University of Notre Dame
sandra.gesing@nd.edu

Abstract

The focus on software usability, long-lasting and reproducible software is a timely one that spans various domains of science and significant investment of research funding both in the US, Europe, U.K, and elsewhere. Software has become a major driver for research with over 90% of researchers answering surveys that they use software for their research and over 65% expressing that they could not even do their research without software. The computational landscape has evolved from system-centered design focusing on training users to user-centered design delivering solutions that are intuitive and/or self explanatory. The prominence of research software creates challenges in the following areas - usability and ease of use, survivability, and reproducibility. Thus, the concept of long-lasting, easy to use software accelerating science is a major concern for researchers. Additionally, researchers would like to be able to re-use software technologies to be able to analyze further data with established and verified methods, which is part of reproducibility approaches.

1. Introduction

The three concepts usability, survivability and reproducibility are interconnected with each other and cover a wide range of application areas. They affect all layers of the software process - from enabling reproducing experiments via an easy user interface to using containerization for application portability. Such concepts are also relevant in the building of Science Gateways (also known as virtual laboratories or virtual research environments), which by definition serve communities with end-to-end solutions tailored specifically to their needs. Software survivability involves a wide scope that can potentially include the following topics:

- Web-based solutions (web sites, science gateways, virtual labs, etc.)
- Application Programming Interfaces (APIs)
- Computational and Data-Intensive Workflows
- Novel approaches in containerization
- Survivability practices in software development
- System architectures for testing and continuous integration
- Emerging best practices in Machine Learning software
- Best practices and Key Success Factors for usability, survivability and reproducibility

This mini-track, Software Survivability: Strategies for Long-Lasting and Usable Software, introduces the wide variety of accepted papers to HICSS-54. It focused on the broad spectrum of submissions that deal with complex scenarios such as containerization, strategies for long-lasting software, usability and user interface issues, handling data curation and provenance and more.

2. Accepted Papers

The three papers accepted to this track introduce the following three topics:

- Sustainability in regards to software-as-a-service
- Research Software Sustainability and
- Usability, version control, archiving and reproducibility

One paper selected for this minitrack, “Sustainability in the Tapis Framework” delves into the Tapis framework and argues as research depends fundamentally on software, sustainability becomes increasingly critical. Nevertheless, despite valiant efforts from a growing number of researchers and practitioners, a basic understanding of best-practices for sustainable software remains elusive. In this paper, we review the specific practices and strategies that have helped to sustain Tapis, a cyberinfastructure project that has been in use for over a decade. The Tapis framework is an open-source, software-as-a-service Application Programming Interface (API) for collaborative, automated, reproducible computational research which began as the Foundation API for the iPlant Collaborative Project in 2008, and today is used by tens of thousands of individuals across more than a dozen active projects. This paper describes our multi-faceted approach to sustaining an increasingly complex ecosystem of software, documentation and other digital assets, including both technical and organizational strategies for minimizing the cost of sustainment while maximizing available resources for sustainment activities.

Further diving into research software sustainability, the second paper, “Research Software Sustainability: Lessons Learned at NCSA” , discusses why research software is important, and what sustainability means in this context. It then talks about how research software sustainability can be achieved, and what the experiences at NCSA have been using specific examples, what the authors have learned from this, and how they think these lessons can help others.

Lastly, authors of the third paper dive into version control and software sustainability. The paper “A Behavioral Approach to Understanding the Git Experience” details the Investigating and Archiving the Scholarly Git Experience (IASGE) project is multi-track study focused on understanding the uses of Git by students, faculty, and staff working in academic research institutions as well as the ways source code repositories and their associated contextual ephemera can be better preserved. This research, in turn, has implications regarding how to support Git in the scholarly process, how version control systems contribute to reproducibility, and how Library and Information Science (LIS) professionals can support Git through instruction and sustainability efforts. In this paper, we focus on a subset of our larger project and take a deep look at what code hosting platforms offer researchers in terms of productivity and collaboration. For this portion, a survey, focus groups, and user experience interviews were conducted to gain an understanding of how and why scholarly researchers use Version Control Systems (VCS) as well as some of the pain points in learning and using VCS for daily work.

4. Conclusion

These papers show a wide range of applications and impact of software survivability in research software. They cover crucial aspects such as reproducibility and cultural approaches, We hope you will join us for interesting presentations and lively discussions on software sustainability, reproducibility, challenges, and solutions for our evolving landscape.