The Science of Scientific Research Software

John D. McGregor
johnmc@clemson.edu
The problem
Problem

• The National Science Foundation (NSF) funds research projects that include software development
• Most often, after the specific grant is over, the software is abandoned
• NSF would like to have a method that is effective and efficient for sustaining some of this software.
• Our hypothesis is that a healthy ecosystem around the research will promote sustainability
NSF’s Goal

• Support the creation and maintenance of an innovative, integrated, reliable, sustainable and accessible ecosystem of software and services that advances scientific inquiry and application at unprecedented complexity and scale.
Scope

- Local - A professor and his students or multiple researchers within the same division of a research institution.
- Institutional - Collaboration between different departments at research institution.
- National - Joint projects between multiple research institutions within a country.
- International - Research networks between multiple institutions across continents.
- Global - Institutions funded by world organizations to tackle the Grand Challenges of science.
Risk

• When a software tool becomes popular outside the research group of developers, the continued use of the software system is a point of risk for any team adopting that software.

• Science outcomes are dependent not only on the continued support of the funded software package, but also on the continued maintenance of the software packages upon which the package depends.

• To gauge the amount of risk, the research group should consider the quality and health of the ecosystem surrounding the software.
Sustainability - 1

• Business and funding models
  – Government funding is in discrete bundles
  – Administrations change and priorities shift
  – National Institutes of Health has given software maintenance only grants

• Reproducibility
  – Research software and data should be sufficiently open to allow replication
  – Central, open data repositories are needed
Sustainability - 2

• Attribution and data curation
  – Ensuring credit and data correctness
  – Code that is integrated into the core may be difficult to cite or to give proper attribution
  – Software and data sets as first class publishing citizens
  – Git now provides a way to store, retrieve, and cite data sets

• Openness of research results
  – Trust among research collaborators
  – Risks associated with software reuse
Facets of Scientific Research Software Development - People

• Usually science “or” computing; rarely science “and” computing; computational scientists are still rare

• Good people are in high demand

Facets of Scientific Research Software Development - Technology

• Software engineering skills are undervalued
• Not on equal footing
• Software engineering is confused with computer science

Facets of Scientific Research Software Development - Software development

– Process must be flexible to address emergent requirements

– Configuration management, issue tracking, etc. are often ignored

http://programmers.stackexchange.com/questions/130850/difference-between-devops-and-software-configuration-management
Facets of Scientific Research Software Development - Science

- Software/hardware differences can inhibit reproducibility.
- Configuration management is needed to build variants quickly.

http://experimentalmath.info/blog/2013/01/set-the-default-to-open-reproducible-science-in-the-computer-age/
Socio-technical ecosystems

• A socio-technical ecosystem is a collection of organizations, people, and technologies related to each other in multiple ways.

• The ecosystem surrounding a software system is a context that includes the influences of collaborating and competing organizations, users, developers, and the domain.
Ecosystem Strategy for Scientific Research Software - 1

• Gatekeepers
  – Ensures integrity of the code and data
  – Open source projects use this approach
  – The Theoretical and Computational Biophysics Group (TCBG) hires gatekeepers to manage the core and the graduate students build extensions
  – Has multiple revenue streams including grants, licenses, and course revenue.
  – “[Stable code] is exactly the opposite of what you call graduate student legacy code.”
Ecosystem Strategy for Scientific Research Software - 2

- Roadmaps
  - Research projects usually require roadmaps
  - The Eclipse Science Working Group (SWG) works to solve the problems of making science software inter-operable and interchangeable.
  - Eclipse projects
    - Transparent decision making about the priorities
    - Maintain roadmaps and have clear life cycles for projects
- Projects
  - The Eclipse Integrated Computational Environment
  - DAWNSci
  - Proposing new project in measurement
  - Clemson University is a founding member
  - https://science.eclipse.org
Ecosystem Strategy for Scientific Research Software - 3

• Visionary Leadership
  – Usually a scientist who recognizes the critical role that software plays
  – Kitware leads by building and hosting the ecosystems for VTK, ITK, and their participation on XDATA.
  – According to Andrew Ross, Director of Ecosystem for the Eclipse Foundation, “Collaboration requires trust. An important part of building trust is enabling a sense of community identity"
Ecosystem Strategy for Scientific Research Software - 4

• Business models
  – Too many research business models begin and end in government funding
  – Kitware uses multiple business models including
    • Research lab – to do experiments
    • Software development organization
    • Ecosystem developer
  – Science Exchange uses a multi-sided market approach to act as a matchmaker between scientific research projects and labs which conduct analyses
  – Consortia/foundations resolve issues of IP ownership and sustainability
Trust

• Clear governance
Value Blueprint for small team scientific research

Lead professor
PhD students
Lab technicians
Collaborating professors

Research results
Peer reviewers
open publishing funders

Published research results

Citers
Free riders
Commercial venture

Spin-off project

Ron Adner’s The Wide Lens
Ron Adner’s The Wide Lens
5 levers applied to scientific software engineering ecosystems

- Relocate – move responsibility for software development to university level
- Separate – common services from domain-specific services
- Combine – identify common services needed and develop as a group
- Add – a lightweight but comprehensive process
- Subtract – remove the reliance on poorly tested software for producing critical scientific results
Future work

• Suppose the cloud is the platform. How would that affect extensions and derivations?
• What does critical mass look like for a scientific research ecosystem?
• What are the basic elements that promote success?
• What information modeling techniques will help?
  – Trust models
  – Collaboration diagrams