Carnegie Mellon

isr institute for SOFTWARE RESEARCH

Scientific Software Ecosystems

James Howison and Jim Herbsleb
Carnegie Mellon University School of Computer Science

Research supported by the NSF Office of Cyberinfrastructure through the Virtual Organizations as Sociotechnical Systems program
NSF Grant #0943168

Our project

- A *socio-technical* investigation of the scientific software ecosystem
- Independently supported by the NSF OCI
    - Three year project begun in November 2009
- Open Science Grid and its VOs providing a scientific context

Our work on Open Source ecosystems

- Thinking above the project level
- Ecosystem metaphors
  - Evolution through variation, selection and retention
  - Niches, Food-chains/feeding hierarchies
  - Not unplanned: e.g., The Apache Software Incubator
- Primary findings in open source:
  - Diverse sources of resources/motivations
  - Components/tasks typically undertaken by individual companies or individuals
  - Governance structures are lightweight

http://floss.syr.edu/Presentations/oscon2006/

CMU/OSG VOSS Workshop

Funded by our NSF grant, held at CalTech February 16/17 (thanks to Kent Blackburn and LIGO)

VO	Participants
SBGrid	Ian Stokes Rees
STAR	Jerome Lauret
Engage	John McGee and Mats Rynge
OSG	Ruth Pordes, Jim Weichel and Miron Livny
IceCube	Greg Sullivan and Erik Blaufiss
LIGO	Kent Blackburn and Chad Hanna
CMS	Liz Sexton-Kennedy
ATLAS	Rob Gardner
UK eScience	David De Roure
EGEE	Charles Loomis

conway2.isri.cmu.edu/scisoft-ecosystem-workshop/

Outcomes

1. Software reuse
   - Why don't we do more?
   - Reuse isn't free
2. Sustaining quality software over long horizons
3. Innovation vs Stability
4. Software and reproducibility
5. Concerns about funding agency policies

Why not always reusing

- Ease and comfort with “blank page” implementation
  - More fun than
- “My requirements aren’t so complex”
  - Not at the start but eventually; need simple routes into complex stacks
- More reputation rewards for project initiators than later contributors

Time Frame mismatches

- Sustaining high-quality software over long time frames
  - Publishing papers
  - Software work as early career “dues paying” – need long-term career path
  - Project-based funding
    - Chunky funding; how to ensure projects properly “spin off”

Innovation vs. Stability

- Clear understanding:
  - Two types of software work: experimentation and production
  - Migration as an important time for review
- How to communicate this to funding agencies and domain science leaders?

Reproducibility

- Reframing software as part of scientific method
  - Understanding variation from software in same way as radiation in experiments
- Understanding that including code binds one its source (firm, community)
- Virtualization as potential
  - But is this just a “once-removed” recursive issue?

Future plans

- Intensive study of a small number of scientific workflows
  - Working back from published paper
  - Identify components, who wrote it, how funded?
- Work to understand extent of software work in science
  - Do funding agencies realize how crucial software is? How much they spend?
- Explore automated methods for assessing impact of individual scientific software components.
- Potentially introduce OSG people to Open Source foundation people (e.g., Apache, Eclipse) perhaps Workshop?