CSESSP Write Up Session C:

Economics of Software Tools

Lead: Ray Idaszak
Presenter: Quincey Koziol
Notes: Ray and Quincey

URL: http://tinyurl.com/csessp-c

C - Economics of Software Tools: Participants over the 2 days

- Ray Idaszak, RENCI - University of North Carolina at Chapel Hill, rayi@renci.org
- Quincey Koziol, The HDF Group, koziol@hdfgroup.org
- Daniel S. Katz, NSF, dkatz@nsf.gov
- Aleksandra Pawlik, Software Sustainability Institute, a.pawlik@software.ac.uk
- Ernie Lucier, NITRD
- Sol Greenspan, NSF
- John McGregor, Clemson
- Thomas Ndousse-Fetter, DOE
- David E. Bernholdt, Oak Ridge National Laboratory, bernholdtde@ornl.gov
- Ronald Boisvert, NIST, boisvert@nist.gov
- Walter Scarborough, TACC - University of Texas, wscarbo@tacc.utexas.edu
- Will Schroeder, Kitware, Inc. will.schroeder@kitware.com
- Costin Iancu, Lawrence Berkeley National Laboratory, cciancu@lbl.gov
- Ira Baxter, Semantic Designs, idbaxter@semanticdesigns.com
- Sandy Landsberg,Sandy.Landsberg@hpc.mil
- Christos Kartsaklis, Oak Ridge National Laboratory, kartsaklisc@ornl.gov

C - Economics of Software Tools

Charge: Explore and focus on the issue of developing and maintaining software tools relevant to this nationally critical, challenging domain that has to compete for scarce resources (e.g., skilled personnel) with a mass market for similar tools.

C - Economics of Software Tools (Summary)

- ***Incentives have to align with desired results***
  - “If you keep doin’ what you’ve done, you’re gonna get what you’ve got”
  - Need holistic view of what constitutes “optimal” economics for software tools, e.g. hardware, software, partnerships, metrics, funding, social, community, incentives
- There are a spectrum of successful models in industry, non-profit, academia, national labs
  - Need more study & development of how to translate these models more pervasively into the community needs to be pursued
- The economics of software tools is improved by more use of software frameworks in support of tool creation (i.e. versus building a tool a time)
- Increasing CSE hardware speed and complexity necessitates increasing developer knowledge
  - This in general decreases the # of developers able to solve the problem
  - Thus increasing the gap between software engineering and domain developers doing coding
  - Need recognition these economics don’t work nor sustain and new model is needed

C - Economics of Software Tools (Summary, continued)

- Identify and develop improved models of CSE software needs that drive hardware architecture and not vice-versa
- Need improved understanding on how to best build CSE community while developing CSE software
- Funding agencies can recognize that better funding long term software maintenance is important
  - e.g. Software like GEANT4 http://www.geant4.org/geant4/ generates more data than the hardware it models
  - Altas probably has a budget for maintenance, so it follows that software should have commensurate budget
  - Also consider dedicated staff for long-term software maintenance
- Need better articulation of incentives to attract and retain top CSE software talent
  - e.g. Prestige, community, freedom and creativity, clearer career path, “wow factor”, sense of service, etc.

C - Economics of Software Tools (Focus)

What industrial and manufacturing partnership models are there?

- Industry can be an outlet for academically developed code as a model for scientific software sustainability
  - Industry can get financial help from funding agencies for scientific software maintenance, e.g. Kitware Inc involved with NIH R01
- Industry-academia consortiums are still relevant for software maintenance
  - e.g. iRODS consortium: http://irods.org/consortium/
  - e.g. Kerberos consortium: http://www.kerberos.org/
- With respect to software development - if it's intended or anticipated to have a community - building the community concurrent with building the software is paramount. This is not always done. Some formalization of how to do this would be of benefit to the CSE community.
  - If the software is developed for the developer's own use, and he/she later decides that a community might be interested, this is a decision point at which the developer then needs to start thinking of building the community concurrently with further software development
  - Recommendations and support for developer(s) taking such leap would be useful

C - Economics of Software Tools (Focus)

What concept-to-supported-product lifecycle models would support tools R&D? Where are the gaps in realizing these models?

- A key recognition in a R&D software tool concept-to-supported-product lifecycle model, don't build a tool at a time, rather build an infrastructure for building tools
- A gap is that tool development infrastructures are not pervasive but need to be
- Related, we look at the R statistical framework: https://www.r-project.org/
  - A given package can serve a group as small as two users or as large as thousands of users.
  - R is one model for how a single software framework, with a strong community, can simultaneously support tools with small and large number of users

C - Economics of Software Tools (Focus)

What university or national lab center and institute models are there?

- University center, institute, national lab models are a spectrum; many models of success
  - Recommend a focus that much more study & development of how to translate these models more pervasively into the community needs to be pursued
  - e.g. HDF and VTK are examples of scientific software successfully spun out into industry
- Government projects that know lifecycle costs (not just software) well can better inform science projects in this area
- Open source
  - Relatively few open source projects work with community contributions like Linux; most lack vibrant user community contributing and supporting it
    - Community building is important at onset when doing open source
    - Successful/useful open source projects will get more community support
  - See community building bullet in Industry slide

C - Economics of Software Tools (Focus)

What role can funding agencies play in fostering efforts?

- Funding agencies can recognize that funding good software engineering during software for science artifact creation is an investment that will save money and increase research productivity in the long term and ultimately produce better software consistently
  - NSF Software Institutes, in part, are on the path to this
- Funding agencies can recognize that funding long term software maintenance is important
  - e.g. specifically dedicated staff for long-term software maintenance
- Software development can be made comparable to purchasing laboratory equipment - there should be some expected maintenance costs involved
- Funding agencies can fund research into software engineering for science and academia
  - e.g. how much infrastructure is appropriate to carry along with any given software project?

C - Economics of Software Tools (Focus)

What are alternatives that can compete with dollars in attracting and retaining skilled software personnel for CSE?

- Prestige; knowing that you are contributing to something that is important at scale (e.g. local, national, global) and that you are publicly recognized for doing so
  - e.g. All US national weather data stored with HDF
- Greater sense of community involvement; ability to work collaborative with a user community
- More freedom and creativity in science and academia
- A more clear career path (but note that we do not have this in academia in general, maybe do in the national labs)
  - Especially with respect to long-term career paths
- Wow factor
  - e.g. same course offered to students garnered different attendance based on description
- A sense of service - doing something for the wider community/nation
- At CPS-PI Meeting 11/6/14, CPS MOOCs - Magnus Ergestedt (GA Tech) advertised a course with three different titles
  - Embedded Software Design - 20 people applied
  - Applied Robotics - 200 people applied
  - IoT - 20,000 people applied