



# A Perspective on Open Source Software Sustainability

Craig A. Stewart, Julie A. Wernert, Eric A. Wernert, William K. Barnett

Indiana University Pervasive Technology Institute

## Summary

Within the niche of software funded in part by the NSF or used by NSF-funded researchers, we have identified characteristics that well-sustained projects have in common:

- A utility and/or flexibility of use sufficiently broad that the software remains relevant over many years
- A strong core of committed leaders, most often co-located in one geographic area
- Control over the definitive software versions and effective test and build processes
- Deep and effective engagement with users
- Use of business models that evolve over time

## Reference

Stewart, C.A., Barnett, W.K., Wernert, E.A., Wernert, J.A., Welch, V., Knepper, R. 2015. "Sustained software for cyberinfrastructure – analyses of successful efforts with a focus on NSF-funded software." Indiana University, Bloomington, IN. Available at: <http://hdl.handle.net/2022/19807>

## Acknowledgements

This research is based upon work supported in part by the National Science Foundation under Award No. 1147606 and builds on work supported by NSF awards 1002526 & 0829462. This work was also supported by the Indiana University Pervasive Technology Institute, which was initiated with major funding from the Lilly Endowment, Inc. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF) or other supporting organizations.

## Contact

William K. Barnett  
Indiana University Pervasive Technology Institute  
Email: [barnettw@iu.edu](mailto:barnettw@iu.edu)  
Phone: (812) 856-3038  
Website: <http://pti.iu.edu>

## Motivation & Objectives

- Software is a core element of cyberinfrastructure and clearly essential for an effective distributed cyberinfrastructure that supports science and engineering research.
- We wanted to understand what factors lead to software projects being **well sustained** over the long run, ultimately lessening barriers for use and increasing research effectiveness and efficiency.

## Benefits of Well-sustained Software

Well-sustained, open-source software:

- Enhances the chances that a given analysis or cyberinfrastructure experiment can be replicated.
- Reduces the vulnerabilities eroding the cybersecurity of the systems in which they are deployed.
- Provides students with resources for learning the craft of research using the same software used by leading US scientists.
- Enables transformative research.
  - For example, verifying of the existence of the Higgs boson; HTCondor and the Open Science Grid enabled the data analysis that confirmed the existence of the Higgs boson.

## Methods

- Survey of 5,000 NSF principal investigators; 685 respondents (17% response rate)
- In-depth case study interviews with 9 well-sustained cyberinfrastructure software projects

## Survey Results – Criteria for Adoption

- The five most important adoption criteria ranked by average importance score in a range of 1 to 5, were:
  1. Capabilities and features: 4.54 ( $\pm 0.05$ )
  2. Total cost of ownership: 4.22 ( $\pm 0.06$ )
  3. Long-term availability: 4.18 ( $\pm 0.06$ )
  4. Reliability/maturity: 4.16 ( $\pm 0.05$ )
  5. Initial purchase cost: 4.00 ( $\pm 0.06$ )
- Respondents overwhelmingly (94%) reported capabilities and features as the most important factor to consider when adopting a software product
- Availability under an open-source license *per se* was far less a concern for most respondents than were its capabilities, cost, and reliability. However, three of the top five most important criteria identified related to characteristics of open source software – total cost of ownership, long-term availability, and initial purchase cost.
- Factors that make a software product **sustainable**, contrasted to those **required for adoption**, with compatibility, availability of support resources, and an active development process cited most often.
- Only 18% of respondents identified software capabilities as key to sustainability.
- Cost factors ranked near the bottom when considering **sustainability** factors.
- Most commonly identified well-sustained software products were as follows (open source products marked with asterix)
  - MATLAB, Microsoft Office, R-project\*, TeX & La TeX\*, Mathematica, SPSS, Adobe Acrobat, Linux\*, Python\* and EndNote.
- Of the top 50 most-cited, commercial products were mentioned roughly twice as often as their open-source counterparts.
- The most-cited open-source projects include R, TeX/LaTeX, Linux, and Python.

## Survey Results – Governance Models

- The five most frequently indicated items, ranked by average importance score in a range of 1 to 5, were:
  1. Hybrid license (commercial/noncommercial users pay different prices), 3.78 ( $\pm 0.10$ )
  2. Contributed effort, organizationally supported model (often a corporation supporting an open-source software tool), 3.65 ( $\pm 0.10$ )
  3. Meritocracy/volunteer-driven model, 3.41 ( $\pm 0.11$ )
  4. Membership/foundation model, 3.35 ( $\pm 0.12$ )
  5. Benevolent/enlightened dictator model, 3.29 ( $\pm 0.12$ )
- Examples cited with governance models that aid the sustainability include Linux, R-project, Apache, Mozilla, TeX & LaTeX, Python, GNU, Eclipse, OpenOffice, Image.

## Case Study Projects

Project	Governance	# Users	Annual user meeting?
<i>Primarily end user applications</i>			
VTK, ITK, CMake, ParaView (Kitware)	Company leads, with stakeholder and advisory committee input on the four products	>100,000	Yes
R project	Foundation with governing board, hosted by university	>1,000,000	Yes
Galaxy	Two PIs as leads (one biologist, one computer scientist)	30,000	Yes
LAPACK	3 person mgmt. team – PI and Co-PIs	Tens of thousands	Yes
Unidata	Director with strategic advisory committee and user committee	55,000	Yes
<i>Primarily middleware (may deliver end-user applications)</i>			
HUBzero	Foundation with board oversees; CEO provides operational leadership	>1,000,000	Yes
Kuali	Foundation with governing board	> 140 higher ed institutions	Yes
Globus Online	University of Chicago owns non-profit company led by PI, with advisory committee	> 14,000 registered users	Yes
HTCondor	Principal Investigator, with input from key stakeholders	> 100,000	Yes

## Case Study Results – Governance & Administration

- Strong, committed, visionary leadership
- Trusted cohort of operational leadership supporting the vision
- A geographically centralized core leadership team
- Strong preference for a permissive, open-source license, but with control of official code releases.
  - *Project leaders who control the official code release are adamant that their software, and thus reputation and future funding success, will not be hindered by badly functioning code.*
- All projects have access to good facilities for testing software, high-quality web pages, and high-quality online documentation or help, including web-based self-help.
- Many provide a mechanism for web-based, community-mediated assistance. This was a factor that was common across software that can be viewed as end-user software or software that provides frameworks accessed directly by end-users

## Case Study Results – Licensing

- Some projects argued against open source as a basic principle for sustained CI software, highlighting the importance of keeping some core components of a software package under some other sort of license as a way to ensure the overall sustainability.

## Case Study Results – Community Engagement

- **Developers engaged with users**
  - A centralized development team with regular contact with users and leadership
  - Developers integrated into support mechanisms, responding to help-desk inquiries; monitoring and participating in listserv, wiki, or blog discussions; and presenting workshops or training classes.
- **Domain experts engaged with developers**
  - Subject-matter (domain) and technological and engineering expertise have an increased probability of broad adoption and sustainability. Agile in adapting to changing domain needs, technologies, and trends, and their robust yet flexible products can be expanded or modified, and possibly adopted outside the original project or domain.
- **Conferences or user meetings**
  - Conferences and user meetings seem to be particularly important for software that has a strong end-user component, and that is focused on end-users, such as R and Galaxy.

## Case Study Results – Financial & Business Models

- Most successful projects studied benefitted from some amount of NSF funding
- NSF investment is seen as essential in "critical-path" scientific software, i.e., utilities, end-user applications, and/or middleware essential to research, discovery, and innovation
- Often noted was the fact that CI software has a lifecycle many times longer than that of the individual hardware systems on which it runs
- Software project leaders: Software should be viewed and funded as "infrastructure" critical to research, discovery, and innovation as more visible investments in supercomputers.
- Large projects with 100K+ users (e.g. Globus Online, HUBzero, and HTCondor) have found ways to turn utilization into cash flow.
- Several projects (e.g., HTCondor, Globus Online) weave together funding from multiple federal agencies.
- Filling a unique needs of a particular domain can generate funding and allow for sustained growth in capabilities and user base.
- Public/private partnerships (e.g., Kitware)
- Private foundation funding has been essential to projects (e.g., Kuali) aiming to solve a significant shared challenge – e.g., the cost of commercial enterprise software for HIGHER ED
- On the horizon: Internet2 NET+ factor
  - A fundamental shift in supporting activities of the US research community.
  - In the NET+ model, the US research community pays directly for services offered to support its activities (rather than NSF paying institutions to offer services to the US research community)
  - InCommon membership is required in adopting NET+ tools, a hindrance especially for smaller schools, compared to open-source software. So far, Globus remains in the "service validation" phase of Internet2 NET+