Models for Sustainability for Robust Cyberinfrastructure Software - Software Sustainability Survey

Julie Wernert
Eric A. Wernert
Craig A. Stewart

Indiana University
PTI Technical Report PTI-TR13-007
January 2013

Citation:

Wernert, J., Wernert, E. A., and Stewart, C. A. “Models for Sustainability for Robust Cyberinfrastructure Software - Software Sustainability Survey,” Indiana University, Bloomington, IN. PTI Technical Report PTI-TR13-007

Table of Contents

A. Document History

Relevant Sections	Version	Date	Changes	Author
Entire document	0.9	August 2012	Initial document	J. Wernert
Entire document	1.0	Jan 2013	Editing throughout	C. Stewart
Entire document	1.02	February 2013	Editing throughout, added tables numbers, data file URL, etc.	J. Wernert

B. Acknowledgements, disclaimers, copyright, and license

This research has been supported, in part, by National Science Foundation (NSF) grant award NSF:1147606 "EAGER: Best Practices and Models for Sustainability for Robust Cyberinfrastructure Software." Craig A. Stewart is the principal investigator (PI) of this award, with Richard Knepper, Von Welch, and Eric A. Wernert as co-principal investigators (Co-PI). This work was also supported with general support from the Lilly Endowment, Inc. to Indiana University in support of the IU Pervasive Technology Institute. Any opinions expressed here are those of the author and to not necessarily represent the opinions of the NSF, the Lilly Endowment, Inc., or Indiana University.

C. Document Body – Software Sustainability Survey Results

C.1. Executive Summary

We conducted a survey of named investigators or personnel on projects funded by the NSF during the five-year period from 2007-2011. A random sample of 5,000 individuals, drawn from a list of 34,901 PIs and Co-PIs, was invited to take this survey; 685 individuals, or 17% of the invitees, completed the full survey.

The first question on the survey asked what factors were important to researchers in selecting a software package. The most important factors were:

1) Capabilities and features of a software product are the most important factors to consider when adopting a software package, with a mean score of 4.54 (on a scale of 5). Respondents overwhelmingly (94%) reported identifying this factor as "important" or "very important."
2) Total cost of ownership (4.22),
3) Long-term availability (4.18),
4) Reliability/maturity (4.16),
5) Initial purchase cost (4.0).

Whether or not a software product was available under an open-source license was far less of a concern for most respondents than were its capabilities, cost, and reliability, which may underscore the fact that most respondents identified their primary role as "software user," rather than a "software developer" or other technical role.

When asked to evaluate the factors required for a software product to be considered sustainable, responses contrasted to those required for adoption, with compatibility, availability of support resources, and an active development process cited most often. Capabilities of a software product were mentioned by only 18% of respondents as key to sustainability, and cost factors ranked near the bottom.

When respondents were asked to identify products that met the requirements for sustainability they had just described, a majority cited commercial products. The ten products listed most often were: MATLAB, Microsoft Office, R-project*, TeX & La TeX*, Mathematica, SPSS, Adobe Acrobat, Linux*, Python*, and EndNote. (An asterisk indicates the four open-source products). Of the top 50 most-cited products, 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.

Respondents were asked to consider the relative success of some common governance models in open software initiatives in creating an environment for long-term sustainability. There was no clear single frontrunner. The five most frequently indicated items, ranked by average importance score in a range of 1 to 5, were:

- Hybrid license (commercial/noncommercial users pay different prices) – 441 responses, 3.78 mean score
- Contributed effort, organizationally supported model (often a corporation supporting an open-source software tool) – 422 responses, 3.65 mean score
- Meritocracy/volunteer-driven model – 388 responses, 3.41 mean score
- Membership/foundation model – 355 responses, 3.35 mean score
- Benevolent/enlightened dictator model – 417 responses, 3.29 mean score.

When asked to cite examples of open software products (or associated companies/consortia/organizations) with governance models that aid the sustainability of their software products, respondents cited a wide range of products with varying governance models. The top 11 tools identified were:

1) Linux
2) R-project
3) Apache
4) Mozilla
5) TeX & LaTeX
6) Python
7) GNU
8) Eclipse
9) OpenOffice
10) ImageJ
11) Google

Google was mentioned by eight respondents. Other projects that were mentioned more than three times each were Java, Kuali, PetSC, ABINIT, LAMMPS, Mathworks/MATLAB, and MySql.

A wide diversity of opinions was expressed in the free-text comments. Common themes include:

- Complaints about pricing and licensing fees for commercial software
- Comments that leadership in software projects is very important
- Comments expanding on support for a particular governance model. No single governance model was mentioned the most.

C.2. Introduction

This document summarizes the responses to the 2012 Software Sustainability Survey, conducted as part of the National Science Foundation EAGER grant titled "Best Practices and Models for Sustainability for Robust Cyberinfrastructure Software." The survey aimed to identify the best practices and models required for developing, deploying, and supporting robust, sustainable cyberinfrastructure software, and to identify key factors users consider in software adoption. Further, the survey attempted to aggregate attitudes about the importance of sustained software in scientific research. The Indiana University Center for Survey Research (CSR) fielded this study for Principal Investigator Craig Stewart, Indiana University Office of the Vice President for Information Technology and CIO, and Co-Investigators Richard Knepper, Von Welch, Eric Wernert, and Grant Analyst/Information Manager Julie Wernert, all of University Information Technology Services, Indiana University.

C.3. Data Collection Methodology

C.3.1. Sample Design

The target population for this study was named investigators or personnel on projects funded by the National Science Foundation during the five-year period from 2007-2011, inclusive. The population was not narrowed to those with knowledge in scientific software or with an interest in developing sustainable software. The intent was to engage a broad spectrum of researchers who develop and maintain software as their primary role, and to gain perspective from users of scientific software. Potential respondents were identified from the NSF Awards database, which is available to the public. The list contained names and email addresses for 34,901 researchers, from which 5,000 were randomly sampled for this study.

As with any survey instrument or resulting report, it is best to avoid reading too much into specific results, either positive or negative. The authors made every effort to accurately summarize and convey the survey results so as to avoid any bias. Readers are encouraged to pay attention to the survey methodology detailed in Section C.2., especially sources of survey error described in Section C.2.6, and to frame any interpretation of responses in the context of the respondent demographics detailed in Section C.4.

C.3.2. Questionnaire

The web questionnaire was developed by Julie Wernert in consultation with the project's principal investigators and the IU Center for Survey Research (CSR). It was programmed using Qualtrics Web Survey Software. The web questionnaire was piloted between February 27 and April 6, 2012, using a small number of known persons from Indiana University with an interest in software sustainability. Based on comments made during the pilot deployment, changes were made prior to launching the actual web survey. The production web survey launched April 20, 2012, and closed June 11, 2012.

Appendix 1 contains the final questionnaire in Word document format.

C.3.3. Data Collection

An email invitation was sent to those included in the pilot sample list on February 27, 2012. The invitation included a summary of the project and the link to the survey's website. Instructions for manually accessing the survey were included at the end of the email message in the event the recipient was unable to use the direct link. One additional email message was sent to those in the pilot sample who had not already responded, refused to participate, or indicated ineligibility at the time the first message was sent.

During the production period, the CSR sent an email invitation to those included in the NSF sample list on April 20, 2012. The invitation included a summary of the project and the link to the survey's website. Instructions for manually accessing the survey were included at the end of the email message in the event the recipient was unable to use the direct link. Two additional reminder messages were sent to those in the NSF sample list who had not already responded, refused to participate, or indicated ineligibility at the time the first message was sent. The final

message alerted the recipients that the study was closing soon and this was their last opportunity to participate. The number of messages sent on each date is noted below for both pilot and production fielding.

Sample Type	Message Type	Date Sent	Number Sent
Pilot	Invitation	February 27, 2012	34
Pilot	Reminder 1	March 13, 2012	21
Production	Invitation	April 20, 2012	4999
Production	Reminder 1	May 8, 2012	4699
Production	Reminder 2	May 31, 2012	4401

Table 1. Data collection schedule and number of messages sent for the EAGER Software Sustainability Survey

Appendix 2 contains the text of the email invitation and reminder messages.

C.3.4. Final Dispositions and Response Rates

The following table classifies every case according to its final disposition. These dispositions are based on the guidelines for final disposition codes established by the American Association for Public Opinion Research (AAPOR) Standard Definitions for Final Dispositions of Case Codes, 2010.

Dispositions	Pilot Sample	Production Sample	Total
Completion (I)	27	658	**685**
Partial completion (P)	6	124	**130**
Refused (R)	0	33	**33**
No response (UH)	1	3905	**3906**
Mailing returned/Undeliverable (UO)	0	113	**113**
Ineligible	0	166	**166**
Total	**34**	**4999**	**5033**

Table 2: AAPOR codes and dispositions for the EAGER Software Sustainability Survey

Using the above disposition codes, the response rate is calculated as follows:

(I+P)/ (I+P) + (R+IR) + (UH+UO) = **.167 or 17%**

For a survey of this sort, a 17% response rate is reasonable.

C.3.5. Post-survey Data Processing and Analysis

The final dataset for the Software Sustainability Survey was prepared in June 2012. First, survey data from the web survey were imported into SPSS software for data cleaning and analysis. The main data cleaning and editing steps for numeric items were as follows. The coding of skipped items was reviewed in SPSS to ensure appropriate assignment of missing values. Data were checked for inconsistencies such as illogical values or inappropriate missing data. Preparation of open-ended items for analysis involved removing any identifying information and coding the additional comment in text responses.

C.3.6. Information Regarding Sources of Survey Error

Surveys of this kind are sometimes subject to types of inaccuracies for which precise estimates cannot be calculated. For example, findings may be influenced by events that take place while the survey is in the field. Events occurring since the time the surveys were completed could have changed the opinions reported here. Sometimes questions are inadvertently biased or misleading. The views of people who responded to the survey may not necessarily replicate the views of those who refused to fill out their questionnaires.

C.3.7. Final Data Set Available for Additional Analysis

The survey data are available online at: http://hdl.handle.net/2022/17312

C.4. Results

C.4.1. Importance of Factors in Software Adoption

With a mean score of 4.54 (on a scale of 5), respondents overwhelmingly reported that the capabilities and features of a software product are the most important factors to consider when adopting a software package, with nearly 94% of respondents identifying this factor as "important" or "very important." According to respondents, other critical factors to consider are total cost of ownership (4.22), long-term availability (4.18), reliability/maturity (4.16), and initial purchase cost (4.0). Whether or not a software product was available under an open-source license was far less of a concern for most respondents than were its capabilities, cost, and reliability, which may underscore the fact that most respondents identified their primary role related to software as that of a "software user," rather than a "software developer" or other technical role.

How important is each of the following factors when deciding whether or not to adopt a particular software package, whether it be commercial, off the shelf, scientific, open source, etc.? Please use a 1 to 5 scale where 1 means not at all important and 5 means critically important.

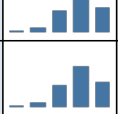
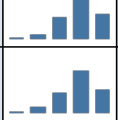
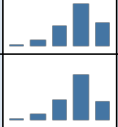
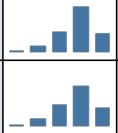
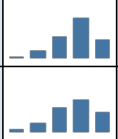
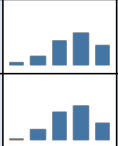
	Number of Responses	Distribution (1 = not at all important, 5 = critically important)					Histogram	mean	std dev	median	mode
		1	2	3	4	5					
Capabilities/Features	810	0.1%	1.1%	5.1%	33.3%	60.4%		**4.53**	0.66	5	5
Total cost of ownership (e.g., renewals, support, hardware/system requirements, etc.)	813	0.4%	2.5%	15.7%	37.9%	43.5%		**4.22**	0.82	4	5
Long-term availability (7)	811	0.6%	3.3%	14.2%	41.4%	40.4%		**4.18**	0.84	4	4
Reliability/Maturity (11)	809	0.2%	2.1%	14.0%	48.9%	34.7%		**4.16**	0.75	4	4
Initial purchase cost (4)	813	0.1%	5.5%	22.0%	38.9%	33.5%		**4.00**	0.89	4	4
Documentation (e.g., manuals, instructions, annotations within the code, online tutorials) (3)	811	0.6%	4.8%	22.9%	44.0%	27.6%		**3.93**	0.87	4	4
Long-term maintenance (8)	810	1.4%	5.2%	24.0%	43.0%	26.5%		**3.88**	0.91	4	4
Interoperability with other tools (5)	811	1.6%	6.5%	21.3%	44.6%	25.9%		**3.87**	0.93	4	4
What software others in my field/industry are using (15)	808	2.7%	6.9%	21.9%	47.5%	20.9%		**3.77**	0.95	4	4
Prior experiences (negative or positive) with company or developers (10)	807	2.9%	9.0%	24.4%	43.5%	20.2%		**3.69**	0.99	4	4
Licensing terms (e.g., redistribution terms, open source terms, etc.) (6)	813	3.6%	11.2%	27.6%	35.1%	22.6%		**3.62**	1.06	4	4
Availability of technical support (1)	810	2.6%	12.2%	29.9%	35.9%	19.4%		**3.57**	1.02	4	4
Strong user community (13)	810	3.2%	10.6%	31.7%	36.4%	18.0%		**3.55**	1.01	4	4
Open source (9)	810	9.4%	15.1%	29.9%	29.9%	15.8%		**3.28**	1.18	3	3
Security features (12)	810	7.2%	18.4%	34.6%	26.0%	13.8%		**3.21**	1.11	3	3
Other 1	83	15.7%	1.2%	6.0%	36.1%	41.0%		**3.86**	1.39	4	5
Other 2	44	29.5%	2.3%	9.1%	25.0%	34.1%		**3.32**	1.67	4	5

Table 3: Importance of factors in software adoption

"Other" factors reported as important in adopting software products by survey participants:

Category	Frequency	Average Importance	Sub-Category	Sub-Cat Frequency	Sub-Cat Avg Import.
User Ease	23	4.57	user - ease of use	11	4.82
			user - ease of learning	7	4.5
			user - easy user interface	3	4
			user - ease of use by students	2	4
Functionality	18	4.44	functionality - ability to customize interface or program	5	4.25
			functionality - file formats, raw data, export	3	4.33
			functionality - supported workflows	2	5
			functionality - extensibility	2	4.5
			functionality - interoperability with other software	2	4.5
			functionality - batch mode or automation	2	4
			functionality - accuracy	1	5
			functionality - functional output	1	5
Systems	18	4.24	systems - OS support	10	4.4
			systems - cloud hosting or compatibility	2	4
			systems - cross-platform	2	4
			systems - HPC ready	2	3.5
			systems - hardware compatibility	1	5
			systems - backward compatibility w/ older hardware	1	4
Development	10	4.8	development - available source code (not nec. Open source)	1	5
			development - developed by peers	1	5
			development - design practices	1	5
			development - governance	1	5
			development - long-term viability of programming language	1	5
			development - long-term viability of supporting libraries	1	5
			development - responsiveness to feedback	1	5
			development - vulnerability to corruption	1	5
			development - API connectivity	1	4
			development - published foundations	1	4
Support Ease	9	4.44	support - ease of support	4	4.25
			support - ease of install	3	5
			support - ease of licensing implementing and compliance	1	4
			support - vendor service	1	4
Quality	8	4.29	quality - speed	3	4
			quality - code quality & integrity	2	4.67
			quality - overall quality	1	5
			quality - stability	1	5
			quality - accuracy	1	3
Documentation	5	4.6	documentation - algorithms (no black boxes)	1	5
			documentation - examples provided	1	5
			documentation - understandable to non-programmers	1	5
			documentation - available 3rd party manuals	1	4
			documentation - textbook available	1	4
Community	4	4.75	community - collaborators use	1	5
			community - employer supplied	1	5
			community - input on releases	1	5
			community - user's conference	1	4
Cost	2	4.5	cost - recurring cost	1	5
			cost	1	4
Product Legacy	2	4	legacy of product	2	4

Table 3a: Other factors important in software adoption

C.4.2. Tolerance for "Immature" Software

Despite some 83% of respondents identifying "maturity and reliability" as key factors in adopting a software product, nearly 75% of respondents also said they were neutral-to-extremely tolerant of software still considered immature, or software that had not evolved into a hardened, robust state.

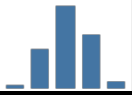
On a scale of 1-5, with 1 being not tolerant at all and 5 being extremely tolerant, describe your tolerance for using software that is still considered immature. That is, software that has not yet evolved into a hardened, robust product. (Again, this includes software considered to be commercial, off the shelf, scientific, open source, etc.)											
	Number of Responses	Distribution (1 = not at all tolerant, 5 = extremely tolerant)					Histogram	average	std dev	median	mode
		1	2	3	4	5					
Tolerance for immature software	811	4.4%	21.1%	41.2%	27.4%	5.9%		3.09	0.94	3	3

Table 4: Tolerance for "immature" software

C.4.3. Requirements for Sustainability

When asked to evaluate what factors were required for a software product to be considered sustainable, responses were in stark contrast to those required for adoption, with compatibility, availability of support resources, and an active development process cited most often. Capabilities of a software product were mentioned by only 18% of respondents as a key factor in achieving sustainability, and cost factors ranked near the bottom of factors mentioned as key to sustainability.

Given that sustained software is, at minimum, software that a user community can expect to be available for the foreseeable future (3 to 5 years), briefly describe what additionally in your view is required for a software package to be considered sustained.					
Response Category	Total Mentions (942)	% of 501 Respondents Mentioning	Common sub-categories & number of mentions		Histogram (max=57)
Compatibility	221	44.1%	compatibility - with OS upgrades	54	
			compatibility - all OS's / platforms	54	
			compatibility - file formats	26	
			compatibility - with other tools	26	
			compatibility - backwards	25	
			compatibility - backwards and forwards	11	
			compatibility - with new hardware (incl. GPUs)	7	
			compatibility - updates to underlying tools and languages	4	
			compatibility - OTHER	14	
Support	183	36.5%	support - bug tracking and fixes	57	
			support - (non-specific)	34	
			support - online community/user forums	20	
			support - easy to install / installation support	8	
			support - access to developers	6	
			support - tech support	6	
			support - by community of developers	4	
			support - easy to update/upgrade	4	
			support - OTHER	44	
Development	135	26.9%	development - active development & frequent updates	29	
			development - updates	19	
			development - regular / periodic updates	18	
			development - new features	8	
			development - open to user input	6	
			development - more than 1 or 2 developers	5	
			development - roadmap	5	
			development - OTHER	45	
Capability	92	18.4%	capability - functionality / fits user needs	15	
			capability - stable / robust / reliable	15	
			capability - correctness	13	
			capability - bug-free	9	
			capability - expandable / extensible	6	
			capability - programmable / flexible / customizeable	6	
			capability - scalability	5	
			capability - efficiency / performance	4	
			capability - includes new algorithms & science methods	4	
			capability - OTHER	15	
Community	77	15.4%	community - large user base / widely adopted	32	
			community - large, active community	21	
			community - adopted as industry/scientific standard	4	
			community - growing and communicative	3	
			community - OTHER	17	
Documentation	63	12.6%	documentation - (non-specific)	29	
			documentation - tutorials	7	
			documentation - manual	5	
			documentation - up to date	4	
			documentation - use cases	4	
			documentation - OTHER	14	

Table 5: Requirements for sustainability

Usability	45	9.0%	usability - ease-of-use / user-friendly	16
			usability - consistency of UI (across versions)	14
			usability - easy to learn	4
			usability - OTHER	11
Duration	33	6.6%	duration - longer than 5 years (10 years)	16
			duration - longer than 5 years (10-20 years)	12
			duration - longer than 5 years (20-50 years)	4
			duration - much less than 5 years	1
Open Source	25	5.0%	open source - (non-specific)	19
			open source - availability of source code	5
			open source - more stable than tech start-ups	1
Funding	24	4.8%	funding - long-term commitment for funding or effort (by NSF, company, community, or developer)	8
			funding - stable funding or fiscal health of company	7
			funding - for developers and support	4
			funding - history / track record of company	3
			funding - OTHER	2
Cost	24	4.8%	cost - low or free (to acquire)	12
			cost - low/free upgrade costs or annual fees	8
			cost - OTHER	4
Licensing	20	4.0%	license - perpetual / no annual renewal fees	8
			license - flexible use / allocation	5
			license - transition to open source if discontinued	3
			license - OTHER	4

Table 5: Requirements for sustainability (continued)

C.4.4. Software Products Meeting Sustainability Requirements

When respondents were asked to identify specific products that met the requirements for sustainability they had just described, a majority cited commercial products. In looking at the top 49 most-cited products, commercial products were mentioned twice as often as their open-source counterparts. The most-cited open-source projects include R, TeX/LaTeX, Linux, and Python.

Please name up to four software products that you have adopted that meet (or have met) the requirements for sustainability just described.	
Respondents	635
Total # of Packages Identified	1950
# of Unique Packages Identified	855
Avg. packages per respondent	3.07

Sustainable software identified by more than 1% of respondents		
Title	Frequency	% out of 635
MATLAB	145	22.8%
Microsoft Office	92	14.5%
R-project	65	10.2%
TeX & LaTeX	47	7.4%
Mathematica	39	6.1%
SPSS	37	5.8%
Adobe Acrobat	33	5.2%
Linux	31	4.9%
Python	29	4.6%
EndNote	25	3.9%
ArcGIS	24	3.8%
IDL	23	3.6%
Microsoft Word	20	3.1%
Microsoft Excel	19	3.0%
LabView	19	3.0%
Adobe Photoshop	19	3.0%
Firefox	18	2.8%
Java	18	2.8%
Maple	17	2.7%
Stata	16	2.5%
SigmaPlot	15	2.4%
Adobe Creative Suite	14	2.2%
GAUSS	14	2.2%
MySQL	14	2.2%
Apache	13	2.0%
Gaussian	13	2.0%
JMP	12	1.9%
Origin	12	1.9%
SAS	12	1.9%
MPI	11	1.7%
Octave	11	1.7%
OpenOffice	11	1.7%
Adobe Illustrator	10	1.6%
ChemDraw	10	1.6%
ImageJ	10	1.6%
Microsoft Powerpoint	10	1.6%
Chrome	9	1.4%
Dropbox	9	1.4%
emacs	9	1.4%
gcc	9	1.4%
geneious	9	1.4%
Igor / Igor Pro	9	1.4%
KaleidaGraph	9	1.4%
Fortran	8	1.3%
Mac OS	8	1.3%
NX	8	1.3%
Systat	8	1.3%
Postgres	7	1.1%
Thunderbird	7	1.1%

Licensing of Top 49 Identified Sustainable Packages		
# commercial	32	65.3%
# open source	16	32.7%
# variable licensing	1	2.0%

Table 6: Software products meeting sustainability requirements

Additional Software Titles Identified as Sustained	
Number of times identified	Titles (in alphabetical order)
6	ANSYS, asp, Clustal, drupal, Eclipse, Mesquite, PETSc, php, ROMS, sage, Sequencher, tecplot, TeXshop, Ubuntu
5	Access, Apache Web Server, ENVI, Gimp, iWork, LAPACK, PAUP/PAUP*, perl, Primer, VisIt, Windows
4	ACT, ATLAS, BLAST, comsol, DreamWeaver, git, google chrome, Google Earth, HDF, IRAF, LibreOffice, Microsoft Powerpoint, MrBayes, Muscle, ncl, NetCDF, NetLogo, Numpy, LibreOffice, Oracle, Paraview, RAxML, SciPy, Statistica, Tomcat, trac, visual studio, Weka, WRF
3	Apple iWork, AutoCAD, Autodesk, BEAST, Canvas, Chemstation, chroma, CLC Workbench, ClustalX, Condor, CPLEX, Debian linux, DNAStar, ELLE, FFTW, Filemaker Pro, GMT, gnuplot, Google Docs, GRASS, Haskell, HDF5, Imagine, iTunes, Lie, LPILE, Lucene, MathCad, Mathtype, mega, Microsoft Visual Studio, Microsoft Windows, MikTex, MOTHUR, MPICH, NetBeans, NING, Papers, pdf, PYMOL, quickbooks, Safari, SHELX, Skype, svn, TurboTax, UCSF Chimera, Unity, VASP, Vernier, vim, xpp, Zemax, Simulink
2	ABINIT, Adobe connect, Adobe Professional, Agilent Chemstation, Alfresco, amanda backup software, Amira, Atlas Ti, Bioconductor, BioEdit, biopython, Blackboard, Blender, Bowtie, C-language, Cactus Computational Toolkit, Cadence, camtasia, CCP4, ChromaTof, Cold Fusion, CUDA, Cygwin, Elmer, ERDAS IMAGINE, Fedora, Fit2D, GAMESS, GEANT, GEMPAK, GeoStudio, Gfortran, gfortran, Globus, Google Apps, Grapher, GraphPad Prism, HYPRE, IGV, ImagePro, iOS, Jmol, jquery, LAMMPS, LS-Dyna, Maxima, Microsoft products, Minitab, Mr Bayes, NAMD, neuron, NIH Image, NVivo, NWChem, OCaml, opencl, OPENFOAM, OpenMRS, OpenSees, ORCA, Outlook, Oxygen, PHENIX, Phylip, PostgreSQL, PostgresSQL, processing, REDCap, redhat linux, Regional Ocean Modeling System (ROMS), ROOT, Sakai, Seaview, Seismic Analysis Code, Serial Cloner, SOLR, Sparky, Spartan, SQL Server, Surfer, Tau, Tecplot 360, TNT, UCINET, UDT, vicon, WIEN2K, WinEdt, WordPerfect, WordPress, XMGRACE

Table 6a: Additional software products meeting sustainability requirements

C.4.5. Software Products Rejected Due to Lack of Sustainability Requirements

Fewer than one in three respondents reported they rejected a software package because it did not meet their sustainability requirements. But those who so reported mention some of the same commercial products previously cited as exemplars of sustainability. Further, while cost and other financial issues were not broadly considered factors essential to achieving sustainability, they again weigh heavily (from a user/customer perspective) when adopting or rejecting a product.

Are there software products that you have wanted to adopt but did not adopt because they did not meet your requirements for sustainability?		
Number of Responses	Distribution	
	1 = yes	2 = no
791	246 *31.1%*	545 *68.9%*

Please cite specific examples of software products that did not meet your requirements for sustainability.			
Number of Software Title responses	320	Number of Reasons Cited	80
Most Frequent Titles Identified		Most Frequent Reasons Identified	
Title	Number of times referenced	Reason	Number of times
MATLAB	9 (4 cost related)	High licensing costs	13
Bibliographic/Citation software	4	changes too frequently	5
PAUP	4	Stop and go development / unclear future	4
Globus	3	single-developer application	3
Mathematica	3 (2 cost related)	poor/unresponsive user support	3
Microsoft Excel	3	immature / wait and see	3
Microsoft Windows	3	underlying algorithm cannot be verified (not open src)	3
NS2/NS3	3	Buggy / inconsistent output	3
Open Office	3	Poorly maintained	3
SPSS	3	non-professional development (by academic unit)	2
StatView	3	Anything not free	2
TreeView	3	Unreliable community add-ons (e.g. for MATLAB, R)	2
VisTrails	3	Audience too broad/too specific	2
CricketGraph	2	Lack of stable funding / young company	2
CUDA	2	Does not meet requirements	2
HyperChem	2		
IDL	2		
Internet Explorer	2		
LaTeX editors	2		
Microsoft Word	2		
NCSS stats	2		
OmniPlan	2		
Scientific Workplace	2		
SigmaPlot	2		

Table 7: Software products rejected due to lack of sustainability requirements

C.4.6. Best Practices for Software Sustainability

When respondents were asked to consider the best software engineering practices essential for sustainable design, over 50% of respondents identified aspects of the development process, including testing, version control, code documentation, and frequent updates. Following closely are practices related to compatibility, chiefly interoperability with other hardware, software, and libraries, and backwards and forwards compatibility, which were noted by nearly 45% of respondents. Practices related to usability, documentation, and design were also citied with some frequency, each being mentioned by some 30% of respondents.

Thinking of best practices in software engineering and/or "software carpentry," what practices, in your view, are most essential for software to be designed or built for sustainability? Please enter up to four different practices.					
Response Category	Total Mentions (1236)	% of 434 Respondents Mentioning	Common sub-categories & # of mentions		Histogram (max=100)
Development	223	51.4%	development - testing / automated testing / test suite	50	
			development - choice of language/library	23	
			development - version control system	19	
			development - frequent updates, new features, regular release schedule	17	
			development - code documentation	14	
			development - OTHER	100	
Compatibility	193	44.5%	compatibility - OS - hardware and device independent / runs on multiple platform	71	
			compatibility - backwards & forwards	37	
			compatibility - interoperability with other packages	37	
			compatibility - file formats / file interchange	23	
			compatibility - other OS issues	9	
			compatibility - interoperability with other libraries & support tools	7	
			compatibility - OTHER (non-OS related)	9	
Usability	143	32.9%	usability - intuitive / good / effective / consistent / simple interface	65	
			usability - ease of use	39	
			usability - ease of learning	6	
			usability - OTHER	33	
Documentation	134	30.9%	documentation - (non-specific)	43	
			documentation - good quality / clear / complete	29	
			documentation - tutorials	11	
			documentation - help files	5	
			documentation - OTHER	46	
Capability	130	30.0%	capability - robust / reliable / stable	22	
			capability - broad functionality / application	12	
			capability - scalability	10	
			capability - flexibility	9	
			capability - functionality for domain / appropriate set of features	7	
			capability - limited set of features done well	4	
			capability - customization	3	
			capability - performance	3	
			capability - OTHER	60	
Design	126	29.0%	design - modularity	37	
			design - extensibility	9	
			design - consult with actual users, also in testing	8	
			design - transparency	4	
			design - requirements analysis	3	
			design - OTHER	65	
Support	103	23.7%	support - (non-specific)	83	
			support - bug reporting / resolution	20	
Community	55	12.7%	community - online user groups	20	
			community - OTHER	35	
Open Source	42	9.7%		42	
Cost	19	4.4%		19	
Licensing	8	1.8%		8	
Other	60	13.8%		60	

Table 8: Best practices for software sustainability

C.4.7. Software Products that Embody Best Practices for Sustainability

When respondents were asked to identify products that employ the software engineering practices that support sustainability they had just described, again a significant majority cited commercial products. In looking at the top 46 most-cited products, commercial products were mentioned 50% more often than their open-source counterparts. The most-cited open-source projects remained consistent, with R, TeX/LaTeX, Linux, and Python most often mentioned as having adopted software engineering processes that aid long-term sustainability.

Please list up to four software products that, in your view, embody the best practices just identified (in Q6)	
Respondents	349
Total # of Packages Identified	817
# of Unique Packages Identified	369
Avg. packages per respondent	2.34

Sustainable software identified by more than 1% of respondent		
Title	Frequency	% out of 349
Matlab	61	17.5%
R-project	35	10.0%
Microsoft Office	21	6.0%
LaTeX	20	5.7%
Python	18	5.2%
Linux	15	4.3%
Adobe Acrobat	14	4.0%
Stata	11	3.2%
Mathematica	10	2.9%
Mozilla FireFox	10	2.9%
Excel	9	2.6%
SPSS	9	2.6%
Adobe Photoshop	7	2.0%
Eclipse	7	2.0%
GCC	7	2.0%
Maple	7	2.0%
EndNote	6	1.7%
java	6	1.7%
LabView	6	1.7%
Mac OS	6	1.7%
Microsoft Word	6	1.7%
SAS	6	1.7%
Adobe Creative Suite	5	1.4%
Apache Software	5	1.4%
ArcGIS	5	1.4%
Chrome	5	1.4%
Emacs	5	1.4%
GRASS	5	1.4%
IDL	5	1.4%
ImageJ	5	1.4%
KaleidaGraph	5	1.4%
Mesquite	5	1.4%
Microsoft PowerPoint	5	1.4%
origin	5	1.4%
Postgres	5	1.4%
sigmaplot	5	1.4%
Adobe products	4	1.1%
Apache web server (Tomcat)	4	1.1%
ChemDraw	4	1.1%
Galaxy	4	1.1%
GAUSSIAN	4	1.1%
Geneious	4	1.1%
iTunes	4	1.1%
JMP	4	1.1%
Numpy/Scipy	4	1.1%
Tecplot / Techplot 3560	4	1.1%

Licensing of Top 46 Identified Sustainable Packages		
# commercial	29	63.0%
# open source	17	37.0%

Table 9: Software Products that Embody Best Practices for Sustainability

Additional Software Titles Identified	
# of times identified	Titles (in alphabetical order)
3	ABINIT; Drupal; ENVI; enzo; FLASH; Miktex; Mozilla software; MPICH; MySQL; octave; OpenMPI; OpenOffice; paraview; PHP; Sequencher; Systat; unix; vim; VisIT; Visual Studio; yt
2	Adobe Flash; Android; ANSYS; BioEdit; biopython; Blackboard; BLAST; cactus; CLC Bio Genomics Workbench; FileMaker; gMail; GMT; GNU/Linux; IDRISI; Igor Pro; iLife; imagemagick; IOS - Apple mobile software; LAMMPS; LAPACK; Lucene; MestReNova; Microsoft products; Microsoft Outlook; Mozilla Thunderbird; MrBayes; NetCDF; OPENFOAM; OpenMRS; perl; PETSc; Red Hat Linux; ROMS; Ruby; scipy; SolidWorks; Spartan; SQLite; Tcl/Tk; texlive; Trilinos; UCINet; VASP; VMD; VTK; WordPerfect; Wordpress

Table 9a: Additional Software Products that Embody Best Practices for Sustainability

C.4.8. Governance Models

When respondents were asked to consider the relative success of the most common governance models used in open software initiatives in creating an environment for long-term sustainability, there was no clear frontrunner.

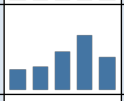
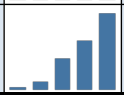
Consider the open software initiative governance models listed below, and indicate how successful you think each is in creating an environment for long-term sustainability. Please rate each model independently using a scale of 1-5, with 1 being least successful and 5 being most successful in terms of creating an environment for sustainability. (If you are unfamiliar with a particular model, please select "Don't know".)												
	Number of Responses	Distribution (1 = least successful, 5 = most successful)					Histogram	mean	std dev	median	mode	Number Unsure
		1	2	3	4	5						
Hybrid License Commercial model (software made openly available for non-commercial use; requires fees for commercial use license or more in-depth support) (3)	441	4.1%	10.7%	18.1%	37.2%	29.9%		**3.78**	1.11	4	4	231
Contributed-effort, Organizationally-supported model (corporation takes open source code and adds features, fixes bugs, offers support, etc.) (4)	422	4.0%	11.6%	24.4%	35.3%	24.6%		**3.65**	1.09	4	4	247
Meritocracy/Volunteer-driven model (distributed control awarded in recognition of technical contributions) (2)	388	4.6%	16.8%	27.6%	34.8%	16.2%		**3.41**	1.09	4	4	282
Membership/Foundation model (organizations contribute resources and/or fees that enable development in exchange for some control over the decision-making process) (1)	355	8.2%	15.8%	26.5%	32.4%	17.2%		**3.35**	1.17	3	4	320
Benevolent/Enlightened dictator model (centralized control by a single individual or small group with contributions by others) (5)	417	12.9%	14.4%	22.3%	31.2%	19.2%		**3.29**	1.29	4	4	245
Other, please specify: (9)	32	3.1%	6.3%	18.8%	28.1%	43.8%		**4.03**	1.09	4	5	115

Table 10: Governance models for sustainability

Specific Comments/Details from 'Other' (only 13 out of 32 provided details)
Assure long term survival somehow (rating: 5)
At the end of the day the software in itself is going to be judged (no rating)
collaborative development (control is not collaboration) (rating: 5)
Combination of meritocracy and dictator (no rating)
I do not see the point of these questions since we already have models that work such as BLAST. I would consult those. (rating: 6)
I don't quite understand these models (rating: 5)
I have seen all of these models work. (rating: 5)
Linux is a combo of method 4 (contributed effort) and 5 (enlightened/benevolent…) I think (rating: 5)
Look at Firefox (rating: 5)
Program Exchange for scientific subgroups (rating: 5)
Services Driven Vendor Model/Open Source Code (rating: 5)
Sponsorship by national laboratories (rating: 4)
they all have problems, but there are also successes based on them as well (no rating)

Table 10a: Comments about governance models for sustainability

C.4.9. Software Products with Governance Models that Aid Sustainability

When asked to cite examples of open software products (or associated companies/consortia/organizations) with governance models that aid (or inhibit) the sustainability of their software products, respondents cited a wide range of products with varying governance models..

List up to four examples of open software products (or associated companies/consortia/organizations) with governance models that, in your opinion, aid the sustainability of their software products.		
Respondents	278	
Total # of Packages Identified	570	
# of Unique Packages/Groups identified	290	
Avg. packages per respondent	2.05	
Sustainability aiding software identified by 4 or more respondents		
Title	Frequency	% of respondents identifying (out of 278)
Linux includes: Ubuntu (9), Red Hat (7), Linux kernel (5), GNU/Linux (3), Linux Foundation (2)	68	24.5%
R-project	45	16.2%
Apache includes: Apache Software Foundation (20), Apache Web Server (5)	25	9.0%
Mozilla inculdes: Firefox (12), Thunderbird (4)	24	8.6%
TeX & LaTeX	22	7.9%
Python & Python Software Foundation	21	7.6%
GNU includes: Emacs (5), gcc (4), Linux (3), other compilers (2), GIMP (2)	21	7.6%
Eclipse	12	4.3%
OpenOffice	12	4.3%
ImageJ	9	3.2%
Google includes: Google Android (4), Google Earth (1)	8	2.9%
java	5	1.8%
Kuali & Kuali Foundation	5	1.8%
PetSC	5	1.8%
ABINIT	4	1.4%
LAMMPS	4	1.4%
Mathworks & MATLAB	4	1.4%
MySql	4	1.4%

Table 11: Software Products with Governance Models that Aid Sustainability

Additional Software Titles Identified as Aiding Sustainaibility	
Number of times id	Titles (in alphabetical order)
3	Drupal, Free Software Foundation, GAMESS, GRASS, Microsoft, Moodle, openfoam, Postgres
2	Adobe Acrobat Reader, BEAST, Blender Foundation, cactus, Canvas, CCP, Computational Infrastructure for Geodynamics, Condor, GALAXY, GIMP, git, GNU/emacs, Gromacs, IETF, Joomla, jQuery, libreoffice, Mathematica, Mesquite, Micromanager, MikTex, NCAR, NS3, Octave, OpenMPI, OpenMRS, opensees, pdf, Phenix, PhP, Python Software Foundation, SAGE, Sakai, SCO, sourceforge, Stata, UNIX, VMD, VTK, WeBWorK, wikipedia, Wordpress, WWW Consortium

Table 11a: Additional software products with governance models that aid sustainability

C.4.10. Software Products with Governance Models that Inhibit Sustainability

List up to four examples of open software products (or associated companies/consortia/organizations) with governance models that, in your opinion, inhibit or interfere with the sustainability of their software products.		
Respondents	155	
Total # of Packages Identified	212	
# of Unique Packages/Groups identified	139	
Avg. packages per respondent	1.37	
Inhibiting/interfering software identified by 2 or more respondents		
Title	Frequency	% of respondents identifying (out of 155)
Microsoft includes Windows (5), Office (4), all products (3)	24	15.5%
Java	15	9.7%
Apple includes iPhone (1), App Store (1)	9	5.8%
MySQL	7	4.5%
OpenOffice	7	4.5%
Adobe includes Framemaker (1), Photoshop (1), Reader/Acrobat (1)	5	3.2%
Google includes Google Apps (1), Maps API (1), pay for service products (1)	5	3.2%
GNU incl. GNU libc (dictator model) (1), licensing restricts corporate contributions (1)	4	2.6%
Oracle	4	2.6%
Perl	4	2.6%
SAS	4	2.6%
ESRI (ArcGIS)	3	1.9%
GAUSSIAN	3	1.9%
Globus	3	1.9%
MATLAB	3	1.9%
Hudson	2	1.3%
Linux	2	1.3%
NCBI	2	1.3%
OpenCL	2	1.3%
Python	2	1.3%
SOAP	2	1.3%
VASP	2	1.3%
Wolfram	2	1.3%

Table 12: Software products with governance models that inhibit sustainability

C.4.11. Additional Comments Regarding Governance

Please offer any comments you may have about the effective governance of open software initiatives:
Total number of responses: 93
Direct Responses (57)
A good proejct must have leadership with clarity of vision and honesty with the user base. Communities go sour when led by people or organizations that say nice things about open source, but then act in ways that are counter to the community's interest.
Any system that is going to try and decide what I need is going to be ineffective and unsustainable. My needs are dynamic and I need to be flexible. Also, the software that tries to do everything is generally not good and doing many things.
As a long term open software user, I am open to switching applications every few years, but I don't tolerate attempts attempts to lock me in via proprietary formats.
As a user (not a contributor/modifier) the most important thing is update manuals. Mr Bayes is still missing two chapters in its manual -- I can't get functionality to work that was purported in its 2001 paper. An annual republishing would be in order fr ALL sustainable software programs. The manual updates can incorporate the questions/comments from the user lists so that the answer to all questions isn't "well, have you checked the mailing list archives?" Each edition of the manual should have a dat at which all the comments prior to that date were incorporated into the manual, so that if one reads the manual, one only has to read the mailing list archive from that date forward. search capabilities for the mailing list archives must also be top ntch. SourceForge's interface doesn't cut it.
Bad soft ware is a disease that corrupts any academic that relies on developing it. The development of lousy code leads to a perversion of the mission of scientific literature.
Behind every tool, there has to be someone who thinks of that tool as their baby and wants to see it grow and prosper.
Best if led by a scientist who cares about others using his/her code. Gaussian has done a few horrible things (including insisting that anyone at a University where Martin Head-Gordon is employed can't use their software - fights over turf)
Biggest problem I have with open source is others are not using software. Until they reach a critical mass of user base, collaboration is difficult. It doesn't do me any good if I can't share my work.
Community-based governance
Dictatorship model only works when truly benevolent. Periodically, revolutions may be necessary.
Effective governance for us equates to 'less is better.' Separate websites for each version of open software that contains voluntary support is a better example of effective governance of open software.
Effective governance may be difficupt to generalize. The personalities of the principal players may be more important than anything else. Personalities determine how the community will rally round an open source software, how it will improve and be adoped.
For volunteer efforts with a benevolent dictatorship, tremendous respect, i.e. a cult of personality, for the benevolent dictator seems to be required.
GAUSSIAN, by its restrictions and practices, inhibits the enhancement of the computational sciences.
Have observed that good will and good expertise are not enough...collapse of consortia due to strong personalities coupled with "rules" that all must agree.
I don't think governance of open source initiatives is a one-size-fits-all proposition. What works will depend on many factors including the problem domain, culture, founders, and the size of the community. Sometimes the best option isn't great, just thatall other options are worse.
documentation to a wiki-style format where a large group of dedicated users can contribute and monitor.If the group at the helm is too small, they will often prioritize other matters over the tedious task of documentation, or else they will simply not realize what issues users are encountering that could be ironed out with better documentation. This can beespecially true of a small company -- it's only as a for-profit company gets larger that they may find documentation an important component of selling their product and building up strong word-of-mouth.
I have seen all different sorts of models work. It really depends on the personalities and skills of the people and/or the commitments of the organizations involved. Just like great companies, it is all about good leadership, good governance, the ability o motivate people, aligned goals, and the ability to get groups to all pull in the same direction. It should be evaluated the same as any other team being given a task.
I think that bad models can work with the right people. But that points to it being less a question of model, than of people. My gut says that it's a matter of matching a model to the community.
I think there's a deep question about scale; the point should be to find points of intervention that, through modularity, etc., allow for small and agile development within a larger initiative.
I think you should have a better target audience for your surveys.
I'm in favor of it. When the dictator is truly benevolent, the dictator model is very effective. But if the software is successful, it's a lifetime commitment! Corporate sponsorship is all very well (Ubuntu, Apache), but one always has questions abut whose interests are being served. A corporation's obligation is to its shareholders, not to the software. My experience with consortia has been uniformly disappointing. Democratic governance sounds nice, but it's hard to get things done and it'seasy to slide into oligarchy. Debian is the only democratically governed project whose operation really impresses me.
I'm not really sure how important governance is. It seems like the quality of the product, devotion of the user base, and quality support are much more important.
I've become a bit more leery of open software governance structures involving corporate control after Sun/Oracle craziness with Java and OpenOffice, though this may be irrational on my part.
In my experience in academia, the biggest issue with open software initiatives is that the recognition of contributions to an open software initiative does not align with recognition within the academic community. For instance, development of sustainablesoftware is often seen at odds with producing rapid, high-impact publications without developing a basis for sustainable software.
in our communituy, finding good developers who can also collaborate is the key to effective 'governance' Small to medium sized research open source projects dont have the funding or the luxury to support governance as an activity, much beyond having and nlightened PI, agreeable proposal reviewers, and clients willing ot use the software. Governance is more of a collaboration without a strong top down benevolent administrative superstructure.
In practice, it seems that many governance structures do not fit cleanly into the options offered - gray areas and hybrid structures are common.
interesting article here: http://randyfay.com/topics/governance
It's really frustrating to have expensive software that becomes obsolete and is no longer supported by the company
LaTeX may be a little too open which has led to some forking

Table 13: Additional comments regarding governance models

Listening to user feedback is critical. Whether the project uses the dictator or meritocracy models, the users should have a large voice in the direction and contect of the product. Again, having a robust and availalbe product does no good, unless a thrving community and economic ecosystem has also been built up. Enthusiastic involvement is the best ingredient for sustainability.
Many companies make the software so expensive that it is not possible for univeristies to purcase and then pay the fees for annual maintenance. Many of these software programs are only used for a few class periods each year and it is not reasonable to bu them all. We have had very good experience with GRL WEAP. They allow us to download the software and use it for a limited amount of time or for a limited number of uses. This makes it easy to use the software for classroom instruction.
Many intentionally wish less sustainability to sell a newer version
MSW is not attentive enough of the needs of the scientific community and their help system is cumbersome in current versions.
Must offer some form of software support, either as a FAC sheet or a contact person
Open source software initiatives that have small communities need to be pragmatic about what they want to accomplish - they have a limited set of participants (highered or academic software) and so they must look hard up front at the overall costs needed o accomplish goals in limited timelines while still seeking community input on the software. Key to this is getting code out and getting an integration process that is open for all to participate.
Peer evaluation based on knowledge seems the most optimal way
Really depends on how widely used the software is - different types of software may require different goverance.
Reliability in access and performance is paramount; For example, too often phylogenetic programs at San Diego's Supercomputing Center were inaccessible, due to viruses, or whatever, that seriously compromised my progress and caused me to abandon their sie
Reliability. One bankruptcy/etc should not derail a program or its development in the future. Transparancy.
Some projects that 'fork' show some demonstration of unhappiness with governance ... mysql and openoffice ... for example. Licensing is probably the most significant expression of governance to most users. There are 1,000s of open source offerings, if nt projects, and a lot of them do not self-sustain.
Structured meritocracy with controls over version submits and full archiving work best
Sustainability by utlitizing Creative Common Licensing models
Sustainability is just not how I think about software.
The biggest issue in my experience with open source software in science is that without strict control one can get multiple diverging branches form with limited to no validation of the actual science being implemented which can errode trust in the softwar and thus the user base.
The model where academic software is ported to a commercial company certainly increase sustainability; however, one drawback is the lack of support for academic users. Two examples are the Antelope (seismic) and Poly3D (rock mechanics) packages. Somehowthe academic users need to be supported.
The openness of eclipse plugins and apache modules as well as good documentation and on-going support yield sustainability. Adobe's handing of FrameMaker (i.e. killing linux/solaris/mac versions, slow update process, poor user communication), inhibits ts use and sustainability.
The patchwork nature of iit makes it hard to discern the ways in which the software will develop and be maintained over time. A matrurity model for governance might be a good idea.
The W3C is a complex case that doesn't fall into "more sustainable" easily ↟ they move incredibly slowly, partially because they're trying to mediate very disparate companies and market forces. I believe that web browsers are less forked than they woul otherwise be without them, however.
There are a lot of projects that do not have effective governance, but you don't hear about them because they don't succeed.
There has to be "someone in charge" but that person or group has to listen broadly to the user community, has to develop and expand the user community, and respect how users implement the product. Too many projects are made to meet the developers' biases and there is a tendency for developers to belittle users' needs.
There ought to be regulations on operating instructions and access to technical assistance.
This is a bit of a "can of worms" at this point as there are so many players. This results in it is often being difficult to determine the quality of the software and/or locate useful "products." It is also problematic for the end user when IS/IT departmnts are the gate-keepers.
This is like a religious debate. Modest fees for open software to academics and higher for commercial users is a reasonable way to achieve sustainability. This all depends on the size of the user community.
Very few effective efforts are built top down, they begin with users/developers who want to work together for a goal that is broader than their individual needs
While I have made contributions to existing open source projects, these projects were small and I simply sent in patches. Whether those changes were ever incorporated was a decision I left to the project lead. I have never interacted in such a way as to b noticeably impacted by a governance policy.
While some sort of imposed standard sounds attractive, such a mechanism also tends to stifle innovation and will have to deal with the Mac versus PC issue. The communities invested in each platform have different tolerances and points of view based upon he characteristics of each platform (example, PC users are function key oriented, Mac users are mouse/menu oriented.).
"I don't know, but..." Responses (9)
I am not closely familiar with governance models for open-source products, and so I cannot judge which methods work better or worse.
I am really rather unfamiliar with the subject. - As a consumer of software, I just want stuff that works and doesn't change all the time in unintuitive ways or inefficient ways. I understand that this desire for stability collides at some time with creatvity and progress. I have a low opioning about software in general. I see at lot of changes (website, editing/graphing software, bank statements, credit card stattements....) that are driven by sytem and software changes, but rarely add any real new featues or new benefits.
I do not know much about governance approaches. Certainly the approaches used by emacs or latex approach deliver good results.
I don't know much about this. It seems to me that the models mentioned earlier could all be successful if well-organized and run. If a project can become well-established, the key to long-term sustainability would depend on the ability to continue after te first flush of excitment and founding contributors and leaders fade away. I don't know which of the models demonstrate or promote this trait.
I don't really know much about this, but I like the R model and also the way that the developers of Qiime are working to help users by high throughput DNA sequencing analysis.
I don't really know what you mean by sustainability or what this survey is about. I can't imagine how you got my name. I use ImageJ, Chimera, and 4peaks but I am by no means a heavy user of these sorts of things and have no opinion or knowledge of how the are maintained/sustained.
I don't understand this topic. I am a Director of Academic Sustainability Programs, in that I manage a minor in sustainability studies. Sustainability is defined for my job as the use of resources so that they can also be used for future generations, and ntegration of social, economic, and environmental considerations. Your study is using the term sustainable in a different context, in terms of how well-established and long-term software use is. I think you have arrived to me as a survey taker by mistake.Be careful next time how you locate your survey takers. Don't use only sustainability as a keyword, but also software, computers, engineering, etc. Most of the content of this survey is outside my area of expertise and the terms and concepts are not cler to me.
I really do not use open software or open software platforms.I prefer to purchase software so that it comes with some type of instruction manual and/or help. Open software to me requires me to learn additional things that I normally do not have time to do therefore it takes away from other things that I should be doing.
Sorry, but I am very ignorant on this subject and I believe that many engineers and educators are as well. I wished I could of been of more help in your survey.

Table 13: Additional comments regarding governance models (continued)

"I don't know" / "I don't use" Responses (27)
I'm not into open source software
You got past me later in the survey
I'm not sure why I was recommended to take this survey, since I know essentially nothing about computer science.
As mentioned previously, I don't have much experience with specialty software beyond that which provides basic functions on my desktop office computer or operates instruments in my lab.
Can't really comment on this as I don't use that much software.
don't know much about this topic
I do not feel qualified to answer this question.
I don't feel I have great insights on this topic.
I don't have enough experience with this to comment.
I don't know much about this.
I don't really have much input/knowledge about this.
I don't work with open code software much, so can't answer the questions above.
I have never thought about this before so don't feel like I can comment.
i have never used open software
I have no experience with open software.
I have no idea about this
I just don't know enough about this topic to comment.
I lack expertise to answer these questions
I still have no idea what are you talking about. Probably I am a wrong addressee, or the survey author was incompetent.
I'm not knowledgeable about this entire area
I'm not qualified to answer these questions.
I'm really not sure I have an educated opinion, one way or the other, regarding the questions on this page.
not my area of expertise
this is not an area that I am really familiar with
This is not really my thing
This is not something I have ever even heard of, let alone have an opinion
Why did I get this survey?

Table 13: Additional comments regarding governance models (continued)

C.5. Respondent Characteristics

The target population for this study was named investigators on projects funded by the National Science Foundation during the five-year period from 2007-2011, inclusive. Potential participants were identified from the National Science Foundation Awards database. The population was not narrowed to those with knowledge in scientific software or with an interest in developing sustainable software, so as to engage a broad spectrum of researchers who develop and maintain software as their primary role, and to gain perspective from users of scientific software. As was anticipated, respondents were overwhelmingly faculty or researchers representing the higher education sector. More than 88% of respondents identified their primary area of study as in the science, technology, and engineering fields. Some 77% of respondents identified government agency funding as their primary source of support, but fewer than 29% had received funding as a principal investigator on a software initiative. Some 67% of the respondents identified themselves as "users" of software products, and another 22% as "developers" of software products, with over 86% percent having decision-making authority (or informing the decision-making process) related to software adoption.

In what industry is your primary professional affiliation?		
Area	Responses	Percent
Higher Education (Postsecondary)	583	84.4%
Research	27	3.9%
Education (Pre-K-12)	23	3.3%
Nonprofit	15	2.2%
Government	11	1.6%
Engineering	8	1.2%
Health Care	6	0.9%
Agriculture/Mining/Construction	4	0.6%
Manufacturing	4	0.6%
Communications/Utilities	3	0.4%
Software Development	3	0.4%
Business or Other Services	2	0.3%
Other	2	0.3%
Total	691	

Table 14: Respondents' primary professional affiliations

Select your primary area of study/research/expertise		
Area	Responses	Percent
Biological and biomedical sciences	122	18.3%
Engineering	88	13.2%
Computer and information sciences	75	11.2%
Mathematics and statistics	62	9.3%
Chemistry	54	8.1%
Geology/earth science	43	6.4%
Social sciences	31	4.6%
Physics	29	4.3%
Education	23	3.4%
Oceanography	21	3.1%
Astronomy and astrophysics, other	18	2.7%
Natural resources and conservation	15	2.2%
Psychology	13	1.9%
Physical sciences, other	11	1.6%
Atmospheric sciences and meteorology	10	1.5%
Foreign languages, literature, and linguistics	9	1.3%
Other, please specify:	7	1.0%
Genetics	5	0.7%
History	5	0.7%
Liberal arts and sciences, general studies and humanities	5	0.7%
Multi/interdisciplinary studies	5	0.7%
Business, management, marketing	4	0.6%
Agriculture and related sciences	3	0.4%
Public administration and social service professions	3	0.4%
Communication, journalism, related programs	2	0.3%
Library science	2	0.3%
Health professions and clinical sciences	1	0.1%
Legal professions and studies	1	0.1%
Total	667	

Table 15: Respondents' primary area of study, research, or expertise

What is your current primary role within your organization?		
Role	Responses	Percent
Faculty member	506	74.0%
Research scientist/Post-Doctoral fellow	78	11.4%
Executive director/administrator	49	7.2%
Project manager	19	2.8%
Graduate student/research assistant	16	2.3%
Analyst/programmer	8	1.2%
Retired Faculty	3	0.4%
System administrator	1	0.1%
Other	4	0.6%
Total	684	

Table 16: Respondents' primary roles

What is the source of your current research funding? (Select all that apply.)		
Role	Frequency	% of Respondents Selecting
Government agency (e.g., NSF, NIH, DoD, DoE, etc.)	633	77.7%
Internal/Institutional (e.g., university, commercial R&D group, etc.)	191	23.4%
Corporate/private sector	96	11.8%
Private foundation	85	10.4%
Sub-contracts	49	6.0%
Business development sources (e.g., SBIR, STTR, etc.)	28	3.4%
Venture capital	7	0.9%
Software consortium or collaborative	2	0.2%
Other	19	2.3%
Total Selections / Total Respondents	1110 / 815	

Table 17: Respondents' sources of funding

What is the primary source of your current research funding? (Select one.)		
Role	Frequency	Percent
Government agency (e.g., NSF, NIH, DoD, DoE, etc.)	218	77.6%
Internal/Institutional (e.g., university, commercial R&D group, etc.)	32	11.4%
Corporate/private sector	10	3.6%
Private foundation	9	3.2%
Business development sources (e.g., SBIR, STTR, etc.)	7	2.5%
Venture capital	3	1.1%
Sub-contracts	1	0.4%
Software consortium or collaborative	0	0.0%
Other	1	0.4%
Total	281	

Table 18: Respondents' primary sources of funding

Are you (or have you ever been) the principal or co-principal investigator on any software development project?		
Number of Responses	Distribution	
	1 = yes	2 = no
683	197 *28.8%*	486 *71.2%*

Table 19: Respondents' Designations as PI or Co-PI

How would you describe your engagement with cyberinfrastructure software? (Select all that apply.)		
Role	Frequency	% of Respondents Selecting
Software user	545	66.9%
Software developer	179	22.0%
Software support (installing, training, trouble-shooting, etc.)	92	11.3%
Software development management & coordination	8	1.0%
Software requirements & design	6	0.7%
Software education & training	5	0.6%
Software selection process	3	0.4%
System administrator	2	0.2%
Other	2	0.2%
Not applicable	87	10.7%
Total Selections / Total Respondents	840 / 815	

Table 20: Respondents' roles in relation to cyberinfrastructure software.

When it comes to the adoption of cyberinfrastructure software for your organization which of the following best describes your role?		
Role	Responses	Percent
You make all technical (and financial) decisions with little input from others on your team / in your organization.	116	18.6%
You make technical (and financial) decisions with a great deal of input from others on your team / in your organization.	309	49.4%
You make technical recommendations, but do not have final decision-making authority.	111	17.8%
You support various software installations, but are not involved in the decision-making or adoption process.	89	14.2%
Total	625	

Table 21: Respondents' roles in software adoption decisions

D. Appendices

Appendix Contents

- Appendix 1: Final Questionnaire
- Appendix 2: Email Invitation and Reminder Messages
- Appendix 3: Data Collection Timing Information

D.1. Appendix 1: Final Questionnaire

EAGER: Software Sustainability Survey

How to complete the survey: Use the buttons on the bottom of each survey page to proceed to the next page or to see the previous page. Your responses to the survey are saved each time you submit a page. Do not use your browser's navigation buttons. You can exit the survey at any point by closing your browser window. To return, access the link in your invitation message.

impadopt

How important is each of the following factors when deciding whether or not to adopt a particular software package, whether it be commercial, off the shelf, scientific, open source, etc.? Please use a 1 to 5 scale where 1 means not at all important and 5 means critically important.

	<b>Not at all important 1 (1)	<b>2 (2)	<b>3 (3)	<b>4 (4)	<b>Critically Important 5 (5)
Availability of technical support (1)	❍	❍	❍	❍	❍
Capabilities/Features (2)	❍	❍	❍	❍	❍
Documentation (e.g., manuals, instructions, annotations within the code, online tutorials) (3)	❍	❍	❍	❍	❍
Initial purchase cost (4)	❍	❍	❍	❍	❍
Interoperability with other tools (5)	❍	❍	❍	❍	❍
Licensing terms (e.g., redistribution terms, open source terms, etc.) (6)	❍	❍	❍	❍	❍
Long-term availability (7)	❍	❍	❍	❍	❍
Long-term maintenance (8)	❍	❍	❍	❍	❍
Open source (9)	❍	❍	❍	❍	❍
Prior experiences (negative or positive) with company or developers (10)	❍	❍	❍	❍	❍
Reliability/Maturity (11)	❍	❍	❍	❍	❍
Security features (12)	❍	❍	❍	❍	❍
Strong user community (13)	❍	❍	❍	❍	❍
Total cost of ownership (e.g., renewals, support, hardware/system requirements, etc.) (14)	❍	❍	❍	❍	❍
What software	❍	❍	❍	❍	❍

others in my field/industry are using (15)					
Other, please specify: (16)	❍	❍	❍	❍	❍
Other, please specify: (17)	❍	❍	❍	❍	❍

tolerance
On a scale of 1-5, with 1 being not tolerant at all and 5 being extremely tolerant, describe your tolerance for using software that is still considered immature. That is, software that has not yet evolved into a hardened, robust product. (Again, this includes software considered to be commercial, off the shelf, scientific, open source, etc.)

- ❍ Not tolerantat all1 (1)
- ❍ 2 (2)
- ❍ 3 (3)
- ❍ 4 (4)
- ❍ Extremelytolerant5 (5)

sustaindef
Given that sustained software is, at minimum, software that a user community can expect to be available for the foreseeable future (3 to 5 years), briefly describe what additionally in your view is required for a software package to be considered sustained.

Please name up to four software products that you have adopted that meet (or have met) the requirements for sustainability just described.

sustprod
Software Product 1: (1)
Software Product 2: (2)
Software Product 3: (3)
Software Product 4: (4)

notsustain
Are there software products that you have wanted to adopt but did not adopt because they did not meet your requirements for sustainability?
- ❍ Yes (1)
- ❍ No (2)

Answer If Are there software products that you have wanted to adopt... Yes Is Selected

notsustain
Please cite specific examples of software products that did not meet your requirements for sustainability.

Example 1: (1)
Example 2: (2)
Example 3: (3)

Thinking of best practices in software engineering and/or "software carpentry," what practices, in your view, are most essential for software to be designed or built for sustainability? Please enter up to four different practices.

softengprac

Practice 1: (1)
Practice 2: (2)
Practice 3: (3)
Practice 4: (4)

Please list up to four software products that, in your view, embody the best practices just identified.

bpracprod

Software product 1: (1)
Software product 2: (2)
Software product 3: (3)
Software product 4: (4)

The rest of the survey will focus on open software initiatives and applications.

socialeng

Consider the open software initiative governance models listed below, and indicate how successful you think each is in creating an environment for long-term sustainability. Please rate each model independently using a scale of 1-5, with 1 being least successful and 5 being most successful in terms of creating an environment for sustainability. (If you are unfamiliar with a particular model, please select "Don't know".)

	Least successful 1 (1)	2 (2)	3 (3)	4 (4)	Most successful 5 (5)	Don't know (6)
Membership/Foundation model (organizations contribute resources and/or fees that enable development in exchange for some control over the decision-making process) (1)	❍	❍	❍	❍	❍	❍
Meritocracy/Volunteer-driven model (distributed control awarded in recognition of technical contributions) (2)	❍	❍	❍	❍	❍	❍
Hybrid License Commercial model (software made openly available for non-commercial use; requires fees for commercial use license or more in-depth support) (3)	❍	❍	❍	❍	❍	❍
Contributed-effort, Organizationally-supported model (corporation takes open source code and adds features, fixes bugs, offers support, etc.) (4)	❍	❍	❍	❍	❍	❍
Benevolent/Enlightened dictator model (centralized control by a single individual or small group with contributions by others) (5)	❍	❍	❍	❍	❍	❍
Other, please specify: (9)	❍	❍	❍	❍	❍	❍

List up to four examples of open software products (or associated companies/consortia/organizations) with governance models that, in your opinion, aid the sustainability of their software products.

openaid

Example 1: (1)
Example 2: (2)
Example 3: (3)
Example 4: (4)

List up to four examples of open software products (or associated companies/consortia/organizations) with governance models that, in your opinion, inhibit or interfere with the sustainability of their software products.

openinhib

Example 1: (1)
Example 2: (2)
Example 3: (3)
Example 4: (4)

governance

Please offer any comments you may have about the effective governance of open software initiatives:

industry

In what industry is your primary professional affiliation?

- ❍ Agriculture/Mining/Construction (1)
- ❍ Banking/Finance/Insurance/Real Estate (2)
- ❍ Business or Other Services (3)
- ❍ Telecommunications/Utilities/Networking (4)
- ❍ Education (Pre-K - 12) (5)
- ❍ Government (6)
- ❍ Health Care (7)
- ❍ Higher Education (Postsecondary) (8)
- ❍ Law (9)
- ❍ Manufacturing (10)
- ❍ Nonprofit (11)
- ❍ Retail/Wholesale Trade (12)
- ❍ Transportation (13)
- ❍ Other, please specify: (14) _____

areastudy

Select your primary area of study/research/expertise?

- ❍ Agriculture and related sciences (1)
- ❍ Architecture and related services (2)
- ❍ Area, ethnic, cultural, and gender studies (3)
- ❍ Astronomy and astrophysics, other (4)
- ❍ Atmospheric sciences and meteorology (5)
- ❍ Biological and biomedical sciences (6)
- ❍ Business, management, marketing (7)
- ❍ Chemistry (8)
- ❍ Communication, journalism, related programs (9)
- ❍ Computer and information sciences (10)
- ❍ Education (11)
- ❍ Engineering (12)
- ❍ English language and literature/letters (13)
- ❍ Foreign languages, literature, and linguistics (14)
- ❍ Family and consumer sciences/human sciences (15)
- ❍ Genetics (16)
- ❍ Geology/earth science (17)
- ❍ Health professions and clinical sciences (18)
- ❍ History (19)
- ❍ Immunology (20)
- ❍ Legal professions and studies (21)
- ❍ Liberal arts and sciences, general studies and humanities (22)
- ❍ Library science (23)
- ❍ Mathematics and statistics (24)
- ❍ Multi/interdisciplinary studies (25)
- ❍ Natural resources and conservation (26)
- ❍ Oceanography (27)
- ❍ Parks, recreation, leisure & fitness studies (28)
- ❍ Pharmacology (29)
- ❍ Philosophy and religious studies (30)
- ❍ Physical sciences, other (31)
- ❍ Physics (32)
- ❍ Psychology (33)
- ❍ Public administration and social service professions (34)
- ❍ Security and protective services (35)
- ❍ Social sciences (36)
- ❍ Visual and performing arts (37)
- ❍ Other, please specify: (38) _____

primerole

What is your current primary role within your organization?

- ❍ Analyst/programmer (1)
- ❍ Executive director/administrator (2)
- ❍ Faculty member (3)
- ❍ Graduate student/research assistant (4)
- ❍ Project manager (5)
- ❍ Research scientist/Post-Doctoral fellow (6)
- ❍ System administrator (7)
- ❍ Other, please specify: (8) _____

fundsource

What is the source of your current research funding? (Select all that apply.)

- ❑ Business development sources (e.g., SBIR, STTR, etc.) (1)
- ❑ Corporate/private sector (1)
- ❑ Government agency (e.g., NSF, NIH, DoD, DoE, etc.) (1)
- ❑ Internal/Institutional (e.g., university, commercial R&D group, etc.) (1)
- ❑ Private foundation (1)
- ❑ Software consortium or collaborative (1)
- ❑ Sub-contracts (1)
- ❑ Venture capital (1)
- ❑ Other: (1) _____

If QID17 (Count) Is Less Than or Equal to 1, Then Skip To Are you (or have you ever been) the p...

primefund

What is the primary source of your current research funding?

If What is the source of your current research funding? (Sel... Business development sources (e.g., SBIR, STTR, etc.) Is Selected

- ❍ Business development sources (e.g., SBIR, STTR, etc.) (1)

If What is the source of your current research funding? (Sel... Corporate/private sector Is Selected

- ❍ Corporate/private sector (2)

If What is the source of your current research funding? (Sel... Government agency (e.g., NSF, NIH, DoD, DoE, etc.) Is Selected

- ❍ Government agency (e.g., NSF, NIH, DoD, DoE, etc.) (3)

If What is the source of your current research funding? (Sel... Internal/Institutional (e.g., university, commercial R&D group, etc.) Is Selected

- ❍ Internal/Institutional (e.g., university, commercial R&D group, etc.) (4)

If What is the source of your current research funding? (Sel... Private foundation Is Selected

- ❍ Private foundation (5)

If What is the source of your current research funding? (Sel... Software consortium or collaborative Is Selected

- ❍ Software consortium or collaborative (6)

If What is the source of your current research funding? (Sel... Sub-contracts Is Selected

- ❍ Sub-contracts (7)

If What is the source of your current research funding? (Sel... Venture capital Is Selected

- ❍ Venture capital (8)

If What is the source of your current research funding? (Sel... Other: Is Selected

- ❍ Other: ${q://QID17/ChoiceTextEntryValue/9} (9)

beenpi

Are you (or have you ever been) the principal or co-principal investigator on any software development project?

- ❍ Yes (1)
- ❍ No (2)

cisoftware
How would you describe your engagement with cyberinfrastructure software? (Select all that apply.)
- ❑ Software user (1)
- ❑ Software developer (1)
- ❑ Software support (installing, training, trouble-shooting, etc.) (1)
- ❑ Other, please specify: (1) _____
- ❑ Not applicable (1)

adoptrole
When it comes to the adoption of cyberinfrastructure software for your organization (e.g., research group, lab, center, etc.), which of the following best describes your role?
- ❍ You make all technical (and financial) decisions with little input from others on your team / in your organization. (1)
- ❍ You make technical (and financial) decisions with a great deal of input from others on your team / in your organization. (2)
- ❍ You make technical recommendations, but do not have final decision-making authority. (3)
- ❍ You support various software installations, but are not involved in the decision-making or adoption process. (4)

participate
Would you be willing to provide further comments for a future, in-depth case study as part of this NSF-funded project?
- ❍ Yes (1)
- ❍ No (2)

Answer If Would you be willing to provide further comments for a fu... Yes Is Selected

Please provide the following information so we may contact you when we begin the next part of this project. (Your contact information will not be associated with your survey responses.)

Answer If Would you be willing to provide further comments for a fu... Yes Is Selected

Name: (1)
Title: (2)
Institution: (3)
Phone: (4)
Email: (5)

Thank you for completing the Software Sustainability Survey! If you would like an electronic copy of the study report sent to you at the conclusion of our research, please check the box below. Once you submit this page, you will not be able to come back to the survey. If you wish to review or change your responses, please do so now.
- ❑ Send me a copy of the report! (1)

Appendix 2: Email Invitation and Reminder Messages

D.1.1. Survey Invitation

From: Craig Stewart [stewart@iu.edu]
Subject: Your Participation in NSF Software Sustainability Study

Dear {FirstName} {Lastname}:

I am writing to ask for your help in a landmark NSF study that is being conducted by Indiana University to identify best practices in the development, deployment, and support of robust cyberinfrastructure software. Data from this study will be used to define software sustainability and to determine specific guidelines for the creation and evaluation of high-quality sustained software. You were suggested to us as someone who is knowledgeable about the use or evolution of sustainable open software and we very much hope to hear your thoughts via this web survey, which averages 20 minutes to complete.

The survey can be accessed here:

https://websurv.indiana.edu/SoftSustain/index.cfm?id={surveyid}{password}

The Indiana University Center for Survey Research (CSR) administers the survey and assures that your responses will remain completely confidential. Neither your name nor your organization will be associated with any data or included in any reports.

In appreciation for your participation, we will send you a copy of the study report at the conclusion of our research. Please specify at the end of the survey that you wish to receive a copy.

If you have any questions about this project or how the results will be used, please feel free to contact Julie Wernert, UITS Grant Information Manager, at jwernert@iu.edu, or call (812) 856-5517.

Thank you for your time and help with this important effort to build a sustainable software infrastructure for 21st century science and engineering in the United States.

Sincerely,

Dr. Craig A. Stewart
Executive Director, Pervasive Technology Institute
Associate Dean, Research Technologies
OVPIT, Indiana University

If you are unable to access the link listed above, please follow these instructions:

- In your Web browser, type: websurv.indiana.edu/Sustain
- In the Survey ID box, enter: {SurveyID}

If you have you have any other difficulties logging in or have questions about the study, please e-mail csr@indiana.edu or call 1-888-226-9234 for assistance.

Reference ID: {popID}

D.1.2. Reminder #1

From: Craig Stewart [stewart@iu.edu]
Subject: Your Participation in NSF Software Sustainability Study

Dear {FirstName} {Lastname}:

Last week I sent a request for your participation in a NSF-funded study which aims to identify best practices in the development, deployment, and support of robust cyberinfrastructure software. Your input will help us create guidelines where there are currently none for the creation and evaluation of high-quality sustained software. It will take about 15 – 20 minutes to complete. I ask that you try to set aside this time in order to help advance the knowledge in this area.

The survey can be accessed here:

https://websurv.indiana.edu/SoftSustain/index.cfm?id={surveyid}{password}

The Indiana University Center for Survey Research (CSR) administers the survey and assures that your responses will remain completely confidential. Neither your name nor your organization will be associated with any data or included in any reports.

If you have any questions about this project or how the results will be used, please feel free to contact Julie Wernert, UITS Grant Information Manager, at jwernert@iu.edu, or call (812) 856-5517.

Thank you for your time and help with this important effort to build a sustainable software infrastructure for 21st century science and engineering in the United States.

Sincerely,

Dr. Craig A. Stewart
Executive Director, Pervasive Technology Institute
Associate Dean, Research Technologies
OVPIT, Indiana University

If you are unable to access the link listed above, please follow these instructions:

- In your Web browser, type: websurv.indiana.edu/Sustain
- In the Survey ID box, enter: {SurveyID}

If you have you have any other difficulties logging in or have questions about the study, please e-mail csr@indiana.edu or call 1-888-226-9234 for assistance.

Reference ID: {popID}

D.1.3. Reminder #2 (Final)

From: Craig Stewart [stewart@iu.edu]
Subject: Software Sustainability Survey Closing

Dear {FirstName} {Lastname}:

Last month I sent a request for your participation in a NSF-funded study which aims to identify best practices in the development, deployment, and support of robust cyberinfrastructure software. Your feedback is very important to us. Your input will help us create guidelines where there are currently none for the creation and evaluation of high-quality sustained software. It will take about 15 – 20 minutes to complete. I ask that you try to set aside this time in order to help advance the knowledge in this area.

The survey can be accessed here:

https://websurv.indiana.edu/SoftSustain/index.cfm?id={surveyid}{password}

The Indiana University Center for Survey Research (CSR) administers the survey and assures that your responses will remain completely confidential. Neither your name nor your organization will be associated with any data or included in any reports.

If you have any questions about this project or how the results will be used, please feel free to contact Julie Wernert, UITS Grant Information Manager, at jwernert@iu.edu, or call (812) 856-5517.

Thank you for your time and help with this important effort to build a sustainable software infrastructure for 21st century science and engineering in the United States.

Sincerely,

Dr. Craig A. Stewart
Executive Director, Pervasive Technology Institute
Associate Dean, Research Technologies
OVPIT, Indiana University

If you are unable to access the link listed above, please follow these instructions:

- In your Web browser, type: websurv.indiana.edu/Sustain
- In the Survey ID box, enter: {SurveyID}

If you have you have any other difficulties logging in or have questions about the study, please e-mail csr@indiana.edu or call 1-888-226-9234 for assistance.

Reference ID: {popID}

D.2. Appendix 3: Data Collection Timing Information

Sample Type	Message Type	Date Sent	Number Sent
Pilot	Invitation	February 27, 2012	34
Pilot	Reminder 1	March 13, 2012	21
Production	Invitation	April 20, 2012	4999
Production	Reminder 1	May 8, 2012	4699
Production	Reminder 2	May 31, 2012	4401

Appendix Table 1. Data collection schedule and number of messages sent for the software sustainability survey