In: Julie M. Esanu and Paul F. Uhlir, Eds, Open Access and the Public Domain in Digital Data and Information for Science: Proceedings of an International Symposium. Washington, The National Academies Press, 2004. (ISBN 0-309-09145-4).

Open Source GIS Software: Myths and Realities

Gilberto Câmara[1], Harlan Onsrud[2]

[1]Director for Earth Observation, National Institute for Space Research (INPE), Brazil
Web: http://www.dpi.inpe.br/gilberto

[2] National Center for Geographic Information and Analysis
and Department of Spatial Information Science and Engineering
University of Maine, Orono, ME 04469-5711, USA
onsrud@spatial.maine.edu

Introduction

The development of open source software (OSS) has received a substantial attention recently. Following the successful examples of projects such as Linux, Apache and Perl, there has been a substantial interest by policy-makers and researchers on the dynamics of the production of open source software (Benkler 2003). A topic of particular interest is the adoption of open source software systems in developing nations, as a means of reducing licensing costs and of promoting indigenous technological development, by having access to the source code of these systems. A recent survey on intellectual property rights and international development commissioned by the government of the United Kingdom underpins such policies with an explicit recommendation:

> *"Developing countries and their donor partners should review policies for procurement of computer software, with a view to ensuring that options for using low-cost and/or open-source software products are properly considered and their costs and benefits carefully evaluated" (Barton, Alexander et al. 2002).*

Many studies that discuss the development of open source software portray an idealized view, which considers such software to be a product of a committed group of individuals. These individuals would operate on a distributed network, where each programmer works on a small but meaningful module. The programmers are isolated, communicating by means of a central repository and mailing lists. The incentives to participate operate on an individual level (Weber 2002). Some authors go as far as identifying in open source software a new mode of organizational structure, denoted by *commons-based peer production* (Benkler 2003). Others claim that the globally distributed skill induced by open source will loosen the grip of the richest countries on innovation (Kogut and Metiu 2001).

This article analyses in detail one segment of open source software market, in an attempt to find out the true extent of such claims and to establish the basis for a realistic view of the open source movement. We will focus on geoinformation technology (GI), that includes geographical information systems (GIS), location-based services, and remote sensing image processing. We have chosen the GI market for two main reasons. First, GI is a key technology for developing nations, given its vast range of applications in areas such as environmental protection, urban management, agricultural production, deforestation mapping, public health assessment, crime fighting, and socioeconomic measurements. Additionally, the authors are experts on the area, with a substantial experience on GI software development, and are in a qualified position to assess the different products.

We consider the following questions: (a) what are the conditions of OSS development? (b) Who builds GI OSS products? (c) Is there a need for innovative OSS applications in geoinformation applications? (d) How can developing countries obtain GI OSS to meet their national needs?

Our survey indicates that the view of open source software as a product of a team of committed individuals is not realistic, at least for the geoinformation market. Most products are built either by a very small team of individuals or by corporations, and that large collaborative networked teams are responsible for a small number of products. Additionally, most projects aim at reverse-engineering existing designs or at complying with standards, and few products are innovative. Therefore, there is much scope for new ideas, especially considering recent advances in geographical information science and spatial databases and the much-increased availability of earth observation satellites. Given the constraints in open source software production, such advances will not happen spontaneously and will require public intervention to fund innovation.

In order to support our claims, we first examine the need for innovative GI tools in Section 2. We consider different models of open source software production from an intellectual property viewpoint in Section 3. Then, we review the process of open source GI software production in Section 4. Finally, in section 5 we propose a model for open source projects in the developing world, based on networks of government-financed institutions.

The Need for Innovation on Geoinformation Technology

One of the motivations for our survey on open source GIS software is to identify the extent of innovation in the community. There are three main drivers for innovation in GI technology: (a) the evolution of database management systems (DBMS) to handle spatio-temporal data types; (b) the availability of a new generation of earth observation satellites; (c) the recent advances in geographical information science.

The complete integration of spatial data types in DBMS is bound to change completely the development of GIS technology, enabling a transition from the monolithic systems of today (that contain hundreds of functions) to a generation of *spatial information appliances*, small systems tailored to specific user needs (Egenhofer 1999). Coupled with the data handling capabilities of new generation of DBMS, rapid application development environments will enable the construction of "vertically-integrated" solutions, directly tailored to the users' needs. Therefore, an important challenge for the GIS community is finding ways of taking advantage of the new generation of spatially enabled database systems to build "faster, cheaper, smaller" GIS technology.

A second important reason for developing open-source spatial analysis tools is the need to resolve the "knowledge gap" in the process of deriving information from images and digital maps. This "knowledge gap" has arisen because our capacity to build sophisticated data collecting instruments (such as remote sensing satellites, digital cameras, and GPS) is not matched by our means of producing information from these data sources (MacDonald 2002). To a significant extent, we are failing to exploit the potential of the spatial data we collect. For example, there are currently very few techniques for image data mining in remote sensing archives, and thus we are failing to use the information available in our large earth observation data archives. Much of this "knowledge gap" has resulted from a substantial imbalance in public expenditure in geoinformation technology. Major earth observation satellites programs such as ENVISAT and EOS have budgets on the billion-dollar range, where the vast majority of the money is spent in building and operating the satellites and sensors.

An additional challenge is how to incorporate recent advances from geographical information science into mainstream GIS. A number of important results have been produced in research areas such as spatio-temporal data models(Erwig, Güting et al. 1999), geographical ontologies(Fonseca, Egenhofer et al. 2002), spatial statistics and spatial econometrics(Anselin 1999), cellular automata(Batty 2000), and environmental modeling (Burrough 1998). These results have largely been outside of the reach of the user community, for lack of widely available tools and systems that support them.

Models of Information Production in Open Source Software

From an intellectual property viewpoint, we distinguish three models of information production for OSS: (a) the *post-mature* model; (b) the *standards-led* model; (c) the *innovation-led* model.

The *post-mature* model arises in strongly consolidated markets. In many cases, one proprietary product has a very large market share. As this product becomes popular, its functionality and conceptual model becomes well established, and it becomes part of the "*public commons*". Switching costs will prevent a new commercial product from capturing market share, even if sold at lower prices. In this case, there is a strong incentive for newcomers to license their products as open source. Many users will consider that the perceived benefits of open source will outweigh the cost of switching from the commercial product they might be using. One example is the *Open Office* productivity suite. Alternatively, a private corporation may decide to license a product previously associated to private intellectual property rights as open source software. Such is the case for the *Mozilla* browser.

The *standards-led* model arises when the establishment of standards consolidates a technology and allows compatible solutions from different producers to compete in the marketplace, thus opening an opportunity for open source products. Newcomers can benefit from the substantial intellectual effort that goes into establishing a standard. An example is the SQL database standard, which has motivated products such as mySQL. Another example is the POSIX standard for operating system interfaces, which has reduced switching costs from other UNIX-based environments to *Linux*.

The *innovation-led* model results when universities, public institutions and corporations produce work that has no direct equivalent in the commercial sector. As we shall see later, innovation is the product of the private sector, either directly (e.g., the *Qt* multi-platform interface system) or by a spin-off of a successful research project. As an example of the latter, the University of California developed the Postgres database management system as a research project (Stonebraker and Rowe 1986). After an unsuccessful commercialization attempt, a private company took over the development of Postgres, added SQL support, named the resulting product PostgreSQL, and made it available as open source.

Who Builds Open Source GIS Software?

In order to conduct a more detailed analysis of the GIS open source software developers, we have conducted a survey of 70 GIS open source projects, mainly using a listing provided by the *freegis.org* site, a repository for open source software. Based on size, geographical distribution and affiliation, we distinguished three categories of OSS development teams:

- *Individual-size projects*: the project team consists of 1-3 individuals, usually from the same location and working in their spare time. The software products usually are small-sized specialized applications, who address specific requirements. In general, the developer of the software is also its first user. Examples include the *Vis5D* visualization tool (Hibbard, Paul et al. 1994), the *shapelib* library for reading ArcView® shapefiles, and the *Gstat* geostatistical package (Pebesma and Wesseling 1998).

- *Collaborative networks*: the project core team consists of a team of 15-30 individuals, geographically distributed. The developers will usually have a separate job, and do their work in their "spare" time, or in part time allocated in agreement with their employer. Examples include the GRASS spatial analysis toolkit and the R collection of statistical functions.
- *Corporation-based:* the project core team is part of an institution and is usually a set of 3-8 programmers. There can be outside collaborators, but the main design decisions are made within the institution and in some cases should also address the commercial objectives of these corporations. Examples include the *PostGIS* extension to the PostgreSQL DBMS, and the *TerraVision* systems for terrain visualization on the Internet.

Table 1 – Characterization and Intellectual Property associated to GIS OSS

	Total	Post-mature	Standards-led	Innovation-led
Individual-based	37 (53%)	12	19	6
Networked Team	4 (6%)	1	1	2
Corporation-based	29 (41%)	6	18	5
	70	19 (27%)	38 (54%)	13 (19%)

We have characterized each product according to its IP model and its development team. The results, shown in Table 1, contradict the naïve view of open source projects as a product of committed teams, based on peer-pressure. More than half of the projects are led by individuals, and only four (6%) are based on a loose network of collaborators. The presence of corporation-based projects is very strong, with 41% of all cases examined. The results are further proof that all software, either open or closed source, is constrained by the *essential* properties of its development process: conceptual design, program granularity, cohesion of the programming team and dissemination strategy.

The relatively small proportion of innovative projects (19%) indicates that the design of most open source software products is based on the *post-mature* and *standards-led* production model, where the main aim is not directly to produce innovation, but to lower licensing costs and to break commercial monopolies. The strong presence of *standards-led* products is also a direct reflection of the influence of the OpenGIS consortium in the developer's community. This result further illustrates the notion that the hardest part of software development is the *conceptual design* of the intended product (Brooks 1982). The two innovative projects developed by a networked team of programmers are GRASS and R. Both products have a simple and well-understood conceptual design, and their innovative contribution lies not in their design, but on the analysis functions that scientists develop using these environments.

Out of the 29 corporations involved in developing open source GIS, 17 are private companies, eight are government institutions, and only four are universities. This result indicates that the research community is usually not interested in a direct involvement in long-term open source projects. Maintaining and supporting an open source software project requires considerable resources, beyond the reach of most university groups. For a research prototype to evolve into an open-source product, a team of developers must take over from the original research team and establish a support and maintenance infrastructure for the product.

Problem granularity is another important factor for open source projects, and each type of software induces a different breakdown strategy. In most cases, there is a strong limit on module size, which forces successful open source products to be the products of small teams. The fact that GRASS consists of a set of independent executables is evidence that open source development by distributed teams requires a software structure that can be broken into small, manageable parts.

Our survey of the open source GIS projects also considered the maturity, support and functionality of each product. We measured the maturity of a project by three factors: (a) the number of software releases; (b) the amount of changes in each release; (c) the achievement of the project's stated goals. For assessment of support, we investigated whether the project has an established maintenance team, and evaluated the mailing lists, bug indicators and improvement requests. Evaluation of the concept of 'functionality' considered on the number of modules and by the difficulty of the algorithms involved. We graded each project on a scale from one to five, where five is best. Table 2 presents the average for each type of project team structure.

TABLE 2 – MATURITY, SUPPORT AND FUNCTIONALITY OF OSS GI

	Maturity	Support	Functionality
Individual-led	2.3	1.7	1.8
Networked team	3.7	3.7	3.7
Corporation-based	3.2	3.1	3.0

The results indicate a significant difference in all three aspects (*maturity, support and functionality*) between individual-led products and corporation-based ones. This indicates that the corporative environment is much better suited for long-term software development than an individual's perspective. Individuals are constrained by their duties, which very rarely include a full-time support for OSS development, whereas many corporations rely on earning indirect revenues (e.g., consultancy fees) from their open source products. In many cases, the corporation might be a public service or develop the product based on public funding. The results also indicate that the difference between a corporation and a networked team is much smaller. This is consistent with the overall picture on the open source world, that a committed team of individuals can produce results which are comparable (or better) than that produced by corporations.

Using and Producing Open Source Software in Developing Nations

The preceding sections have examined the nature of open source software development and outlined the main characteristics of its production. We have argued that most mature and successful OSS products require the establishment of organizational structures dedicated to their production. The consequences for developing nations are significant. Many developing nations are currently actively considering policies to support or enforce the adoption of OSS by public institutions (Dravis 2002). The arguments in favor of OSS adoption by public institutions include (Ghosh, Krieger et al. 2002):

- *Lower cost:* adoption of personal computers based on OSS for public use can reduce initial entry cost by as much as 50%. Easier replication of solutions is also possible. Large-scale public projects can greatly benefit from having a prototype developed and tested, that can then be replicated across the country with no additional software costs.
- *Independence from proprietary technology*: many governments are increasingly concerned with over-dependence in some important markets to a small number of vendors.
- *Security*: governments and governmental agencies are becoming aware of the risks they are subject to when adopting proprietary software solutions, in sensitive areas such as e-government, e-procurement, elections, and public finance.
- *Availability of efficient and low-cost software:* the virtuous examples of some products (such as Linux and Apache) have encouraged statements about the widespread availability of OSS software for public use.
- *Ability to develop custom applications and to redistribute the improved products*: Given the "open" nature of OSS, skilled local programmers could adapt the software to fit local needs, and thus increase the efficiency of the services provided by the improved products.

While the authors consider that there is enough empirical evidence to support the first three claims, the issues regarding "software availability" and "ease of customization" are far more problematic and require a much closer examination. Most successful open source software tools are infra-structural products, such as operating systems, programming languages and web servers. By contrast, the number of mature open source end-user applications is much smaller (Schmidt and Schnitzer 2002). Operating systems, compilers and Web servers respond to the needs of technically qualified IT professionals, who can more easily adapt to the demands of products where support might only be available on the Internet, and requires expertise in the English language.

There is a huge demand for end-user applications in developing nations, especially in the public sector. However, our survey indicates that corporations dominate the development of open source software. These corporations will develop software based on their strategic interests, which are unlikely to include the full range of end-user applications needed by developing countries. Therefore, if governments in developing nations aim to profit from the potential benefits of open source, they must intervene and dedicate a substantial amount of public funds to support the establishment and long-term maintenance of open source software projects. The benefits of this strategy could be substantial. Consider, for example, the case of urban cadastral systems based on a spatial database for middle-sized cities. The typical base cost of a commercial spatial database solution for one city is US$ 100,000 (one hundred thousand dollars). Should 10 cities adopt such solution in a given year, there is a saving of US$ 1 million/year on licensing fees, which can finance local development and local adaptation.

There is also a substantial additional benefit of investing on qualified human resources. Government strategies for supporting indigenous open source software development and adaptation would result in a "learning-by-doing" process. Such processes, as opposed to "learning-by-using", are credited with fostering innovation in the developed world (Landes 1999) and the same lessons would appear to apply for those nations supporting emerging economies.

As an example of government-funded projects, a group of R&D institutions in Brazil is currently developing *TerraLib*, an open-source GIS library that enables quick development of custom-built applications for spatial data analysis (available at www.terralib.org). As a research tool, TerraLib aims to enable the development of GIS prototypes that would include recent advances in GIScience. On a practical side, TerraLib enables quick development of custom-built applications using spatial databases. We believe that projects such as TerraLib show that open source GIS projects can make substantial contributions to the spatial information community, by providing a platform for innovation and collaborative development(Câmara, Souza et al. 2000).

Conclusions

This work examines the nature of open source software development, by looking in detail on the application area of geoinformation technology. We surveyed 70 open source GIS software projects and concluded that the “Linux paradigm” is the exception rather than the rule, and that corporations are the main developers of successful open source products. Since networked teams develop only 6% of the all open source GIS products, our result refutes the view that open source software development defines a new “mode of production”. As established by extensive research, good software design and development are the products of qualified teams that operate at a high level of interaction. Developing software in a decentralized manner requires a modular design, which is difficult to achieve for most applications, since few software products can be broken in very small parts without a substantial increase in interaction costs.

The direct participation of universities in open source software is limited, due to the conflict between the generation of new research ideas and the need for long-term software maintenance and upgrades. As a result, innovative projects account for less than 20% of the total and a large proportion of the projects (53%) simply aim to provide standardized components for spatial data processing. Individuals or small teams develop more than half of the products surveyed, and their best results are specialized applications aimed at conversion and visualization of data in established formats. Corporations account for 41% of all products and have a much better quality than individual-led software. This demonstrates that the impetus behind open source software is not coming from altruistic individuals working in the midnight hour, but from professional programmers.

These results have important consequences for public policy guidance. First, good open source software is the product of corporations, which will build them based on their strategic intents. Therefore, governments worldwide who try to benefit from the open source software model by simply establishing legislation that mandates its use could be frustrated in their objectives, because of the lack of suitable public-sector applications. In order to create the software they need, governments need to establish public-funded projects for open source development and adaptation to local needs. Failure to understand the open source development model will result in a lost opportunity for the developing world to reduce the current technological gap between the rich and poor nations.

Acknowledgements

The first author would like to thank Antonio Miguel Monteiro, Ricardo Cartaxo Modesto de Souza, Ubirajara Moura de Freitas and the rest of INPE's staff at the Image Processing Division for the great enthusiasm and support during a trajectory of 25 years of free software development.

References

Anselin, L. (1999). Interactive techniques and Exploratory Spatial Data Analysis. Geographical Information Systems: principles, techniques, management and applications. P. Longley, M. Goodchild, D. Maguire and D. Rhind. Cambridge, Geoinformation International.

Barton, J., D. Alexander, et al. (2002). Integrating Intellectual Property Rights and Development Policy. London, UK Department for International Development.

Batty, M. (2000). GeoComputation Using Cellular Automata. GeoComputation. S. Openshaw and R. J. Abrahart. London, Taylor&Francis**:** 95-126.

Benkler, I. (2003). "Coase's Penguin, or, Linux and The Nature of the Firm." Yale Law Journal **112**(Winter 2002-03).

Brooks, F. (1982). "No Silver Bullet: Essence and Accidents of Software Engineering." IEEE Computer.

Burrough, P. (1998). Dynamic Modelling and Geocomputation. Geocomputation: A Primer. P. Longley, S. Brooks, R. McDonnell and B. Macmillan. New York, John Wiley.

Câmara, G., R. Souza, et al. (2000). TerraLib: Technology in Support of GIS Innovation. II Workshop Brasileiro de Geoinformática, GeoInfo2000, São Paulo.

Dravis, P. (2002). A Survey on Open Source Software. San Francisco, CA, The Dravis Group.

Egenhofer, M. (1999). Spatial Information Appliances: A Next Generation of Geographic Information Systems. First Brazilian Workshop on GeoInformatics, Campinas, Brazil.

Erwig, M., R. H. Güting, et al. (1999). "Spatio-Temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases." GeoInformatica **3**(3): 269-296.

Fonseca, F., M. Egenhofer, et al. (2002). "Using Ontologies for Integrated Geographic Information Systems." Transactions in GIS **6**(3): 231-257.

Ghosh, R. A., B. Krieger, et al. (2002). Open Source Software in the Public Sector: Policy within the European Union. Maastricht, International Institute of Infonomics, University of Maastricht, The Netherlands.

Hibbard, W., B. Paul, et al. (1994). "Interactive Visualization of Earth and Space Science Computations." Computer **27**(7): 65-72.

Kogut, B. and A. Metiu (2001). Distributed Knowledge and the Global Organization of Software Development. Boston, MA, Massachusetts Institute of Technology.

Landes, D. S. (1999). The Wealth and Poverty of Nations. New York, W. W. Norton & Co.

MacDonald, J. (2002). The Earth Observation Business and the Forces that Impact It. Earth Observation Business Network 2002, Vancouver, CA, MacDonald Dettwiler.

Pebesma, E. J. and C. G. Wesseling (1998). "Gstat: a program for geostatistical modelling, prediction and simulation." Computers & Geosciences **24**(1): 17-31.

Schmidt, K. M. and M. Schnitzer (2002). Public Subsidies for Open Source? Some Economic Policy Issues of the Software Market. Munich, Seminar for Economic Theory, Ludwig Maximilian University.

Stonebraker, M. and L. A. Rowe (1986). The Design of POSTGRES. ACM-SIGMOD International Conference on the Management of Data. Washington, D.C.**:** 340-355.

Weber, S. (2002). The Political Economy of Open Source. Berkeley, CA, University of California.

TABLE 3 – INDIVIDUAL PROJECTS

Software	**IP**	**Mat**	**Sup**	**Funct**	**Webpage**
CAVOR	Post-mature	2	2	2	www.cavor.org/
DEM Tools	Standards	2	1	2	www.arq.net/~kasten/demtools/
DEM Viewer	Standards	2	1	2	www.geogr.uni-jena.de/~p6taug/demviewer/
DLGVU	Standards	3	1	1	www.cs.arizona.edu/icon/oddsends/dlgvu/
e00compr	Post-mature	3	2	2	pages.infinit.net/danmo/e00/
Efoto	Post-mature	1	1	1	e-foto.sourceforge.net/
FMAPS	Standards	1	1	1	FMaps.sourceforge.net/
G3DGMV	Standards	2	1	2	g3dgmv.sourceforge.net/index.html
GAMA	Standards	3	2	2	www.gnu.org/software/gama/
GDAL	Standards	3	3	3	www.remotesensing.org/gdal/
GeoServer	Standards	2	2	2	geoserver.sourceforge.net/
GISToolkit	Post-mature	3	2	2	gistoolkit.sourceforge.net/
GMT	Post-mature	4	3	2	gmt.soest.hawaii.edu/
gstat	Innovation	4	3	4	www.gstat.org/
HADES	Innovation	2	1	1	www.siliconcavings.org/
InlineWMS	Standards	1	1	1	sourceforge.net/projects/inlinewms/
IVICS	Post-mature	3	1	2	www.nsstc.uah.edu/ivics/
KDEM	Standards	2	1	2	www.mindspring.com/~jamoyers/kdem/
libgeotiff	Standards	3	3	2	www.remotesensing.org
MapEditor	Standards	1	1	1	sxpert.esitcom.org/projects/mapeditor/
MB	Innovation	4	3	3	www.ldeo.columbia.edu/MB-System/
mitab	Post-mature	3	3	2	mitab.maptools.org/
NCVIew	Standards	3	3	2	meteora.ucsd.edu/~pierce/
OpenSVG	Post-mature	2	1	1	www.carto.net/projects
PopMap	Post-mature	1	1	1	popmap.sourceforge.net/
QuantumGIS	Post-mature	1	1	1	qgis.sourceforge.net/
RMAP	Standards	2	2	1	www.reza.net/rmap/
RoadMap	Post-mature	2	2	2	roadmap.saignon.net/
shapelib	Post-mature	3	3	2	gdal.velocet.ca/projects/shapelib
TerraForm	Standards	3	2	2	terraform.sourceforge.net/
Therion	Innovation	2	2	2	therion.speleo.sk/
TkGeomap	Standards	2	2	2	tkgeomap.sourceforge.net
vhclmaps	Standards	1	1	1	www.ivtools.org/vhclmaps/
Vis5D	Innovation	4	2	3	www.ssec.wisc.edu/~billh/vis5d.html
VTP	Innovation	2	1	2	vterrain.org/
WKB4J	Standards	1	1	1	wkb4j.sourceforge.net/
XRMAP	Standards	3	1	1	frmas.free.fr/li_1.htm

TABLE 4 – CORPORATE PROJECTS

Software	**IP**	**Mat**	**Sup**	**Func**	**Status**	
3MAP	Innovation	3	4	4	Private	www.ping.com.au/3map/
DEGREE	Standards	3	3	3	Private	deegree.sourceforge.net/
DEM3D	Standards	2	2	2	Gov	craterlake.wr.usgs.gov/dem3d.html
GeoTrans	Standards	4	3	3	Gov	164.214.2.59/GandG/geotrans/geotrans.html
GeoVRML	Standards	4	4	3	Private	www.geovrml.org/
GeoVista	Innovation	4	4	4	Univ	www.geovista.psu.edu/
GISViewer	Standards	3	3	2	Univ	elib.cs.berkeley.edu/gis/
GSLIB	Innovation	4	2	4	Private	www.gslib.com
IDV	Standards	3	3	3	Univ	my.unidata.ucar.edu/content/software/IDV/
ImageMagick	Post-mature	4	4	3	Private	www.imagemagick.org
JTS	Standards	3	3	3	Private	www.vividsolutions.com/jts/
LAS/ADAPS	Standards	4	2	4	Gov	edc.usgs.gov/programs/sddm/lasdist/
MapIt	Post-mature	2	2	2	Private	www.mapit.de/index.en.html
MapLab	Standards	3	3	2	Private	www2.dmsolutions.ca/webtools/maplab/
MITOrtho	Standards	3	3	2	Univ	tull.mit.edu/orthoserver/
mySQL	Standards	4	4	5	Private	www.mysql.com
NCAR	Post-mature	4	4	3	Gov	ngwww.ucar.edu/ng4.2/
NRDB	Post-mature	3	2	3	Gov	www.nrdb.co.uk/nrdbview.html
OpenEV	Standards	2	2	2	Private	openev.sourceforge.net/
OpenMap	Standards	4	3	2	Private	openmap.bbn.com/
OSSIM	Post-mature	3	3	3	Private	www.ossim.org/
PDFMap	Standards	2	2	2	Private	pdfmap.sourceforge.net
PostGIS	Standards	3	3	3	Private	postgis.refractions.net/
PostgreSQL	Standards	4	4	5	Private	www.postgresql.com
STDS++	Standards	3	3	2	Gov	mcmcweb.er.usgs.gov/sdts/sdtsxx/
TerraLib	Innovation	3	4	4	Gov	www.terralib.org/
TerraView	Post-mature	3	4	4	Gov	www.dpi.inpe.br/terraview/
TerraVision	Innovation	5	5	5	Private	www.tvgeo.com/
Thuban	Standards	2	3	2	Private	thuban.intevation.org/

TABLE 5 – TEAM PROJECTS

Software	**IP**	**Mat**	**Supp**	**Funct**	**Webpage**
GRASS	Innovation	5	5	5	grass.itc.it/
GeoTools	Post-mature	2	2	2	geotools.sourceforge.net/
MapServer	Standards	4	4	3	mapserver.gis.umn.edu
R	Innovation	4	4	5	www.r-project.org/