Free/Open Source Software Development: Recent Research Results and Emerging Opportunities

Walt Scacchi
Institute for Software Research
University of California, Irvine

ESEC-FSE07 State of the Art Seminar
8 September 2007
http://www.ics.uci.edu/~wscacchi/Presentations/ESEC-FSE07/

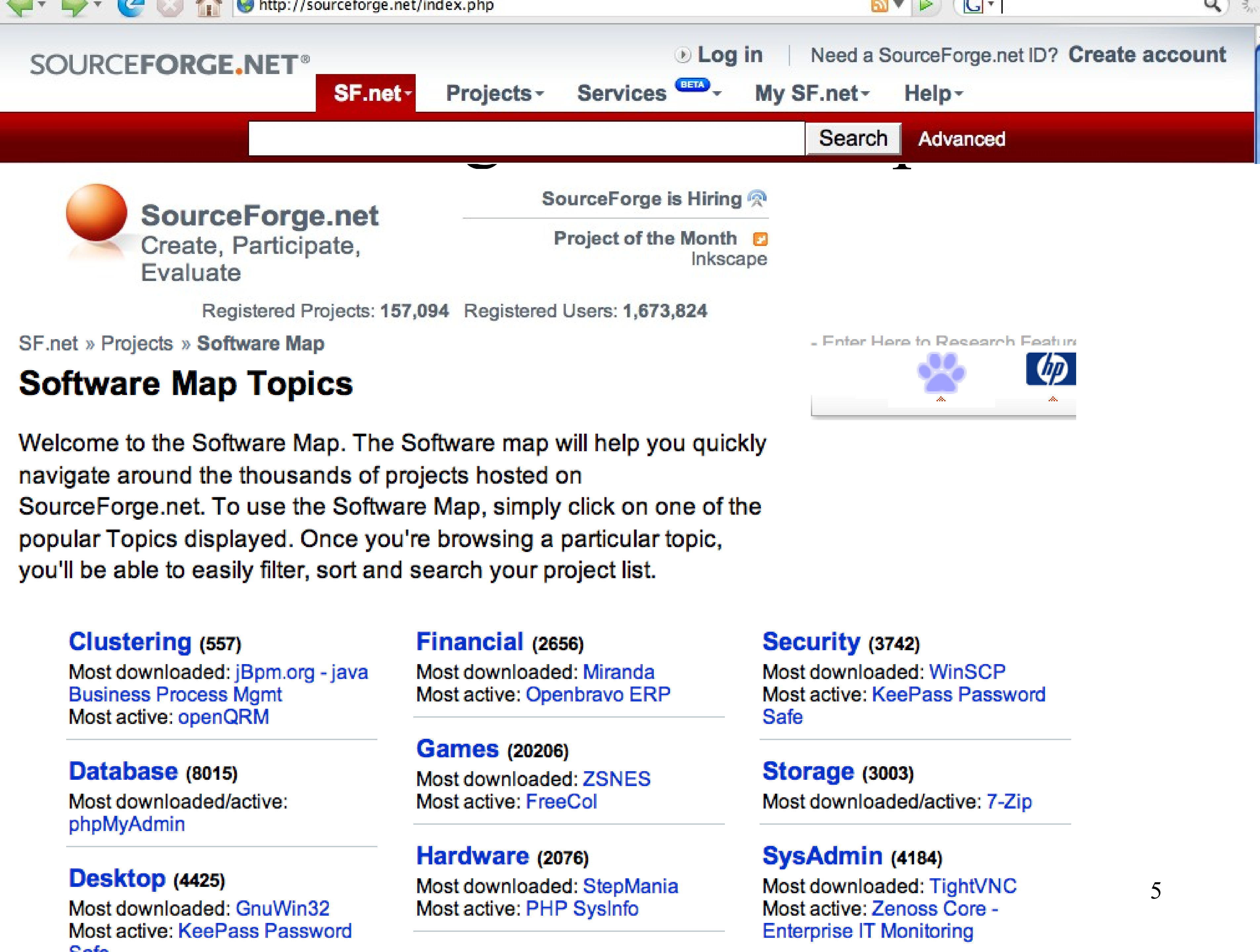
Overview

- Background
- Individual participation
- Resources supporting activities
- Cooperation, coordination and control
- Alliances and social networking across projects
- FOSS as multi-project software ecosystems
- FOSS as social movement
- Discussion and limitations
- Research opportunities

Background

What is free/open source software development?

- Free (as in "freedom" or liberty) vs. open source
  - Freedom to access, browse/view, study, modify and redistribute the source code
  - Free is always open, but open source is not always free
- FOSSD is not "software engineering"
  - *Different*: FOSSD can be faster, better, and cheaper than SE in some circumstances
  - FOSSD teams use 10-50 OSSD tools and communications applications to support their development work



SourceForge.net
Create, Participate, Evaluate

Registered Projects: **157,094** Registered Users: **1,673,824**

SF.net » Projects » **Software Map**

Software Map Topics

Welcome to the Software Map. The Software map will help you quickly navigate around the thousands of projects hosted on SourceForge.net. To use the Software Map, simply click on one of the popular Topics displayed. Once you're browsing a particular topic, you'll be able to easily filter, sort and search your project list.

Clustering (557)
Most downloaded: jBpm.org - java Business Process Mgmt
Most active: openQRM

Database (8015)
Most downloaded/active: phpMyAdmin

Desktop (4425)
Most downloaded: GnuWin32
Most active: KeePass Password Safe

Financial (2656)
Most downloaded: Miranda
Most active: Openbravo ERP

Games (20206)
Most downloaded: ZSNES
Most active: FreeCol

Hardware (2076)
Most downloaded: StepMania
Most active: PHP SysInfo

Security (3742)
Most downloaded: WinSCP
Most active: KeePass Password Safe

Storage (3003)
Most downloaded/active: 7-Zip

SysAdmin (4184)
Most downloaded: TightVNC
Most active: Zenoss Core - Enterprise IT Monitoring

FOSSD Project Characteristics

- Operational code early and often--actively improved and continuously adapted
  - Short-cycle (FOSS) vs. long-cycle (SLC) time processes
- *Post-facto* software system requirements and design
  - FOSSD has its own "-ilities" which differ from those for SE
- *Caution*: the vast majority (>90%) of FOSSD projects fail to grow or to produce a software release.

FOSSD Project Characteristics

- FOSS developers are typically users of what they build, while FOSS users (~1%) are also FOSS developers
- Requires "*critical mass*" of contributors and FOSS components connected through socio-technical interaction networks
- FOSSD projects can emerge/evolve via *bricolage*
  - Unanticipated architectural (de)compositions
  - Multi-project component integrations

OSS Development Models

- Free Software (GPL)
- Open Source (BSD/MIT, Mozilla, Apache)
- Corporate Source (Hewlett-Packard)
- Consortium/Alliance (OSDL, SugarCRM)
- Corporate-Sponsored (IBM-Eclipse, Sun-Netbeans, Sun-OpenOffice, HP-Gelato)
- Community Source (Sakai, Westwood)

- Shared Source with Non-Disclosure (Microsoft)
- Open Systems (open APIs, closed components)

OSS Business Revenue Streams

- *Consulting and Migration* services
  - On-site custom development and support
- *Subscription support* services
  - Ongoing maintenance services 24/7
    - Email or phone help desk
  - Indemnification
  - Access to Operations Network
- *Training* services
  - Web-based how-to’s, tutorials (also retail books)
  - On-site customer training (Direct or via Certified Partners)
- Services sold on a direct basis (e.g., in North America and Europe), and via Certified Partners (globally)

Research methodology

- Early empirical case studies of FOSSD Projects
  - Mockus, Fielding, Herbsleb, 2000, 2002, Apache httpd server
  - Reis and Fortes, 2002, Mozilla Web browser
  - Schach *et al.*, 2002; Holt *et al.*, 2000, Linux Kernel
  - Koch and Schneider 2001; German 2002, GNOME User Interface
  - Jorgensen, 2001, FreeBSD operating system
  - Garg *et al.*, 2002, OSSD (“progressive open source”) within HP
  - Jensen and Scacchi, 2003-04, NetBeans IDE
  - etc.

Research methodology

- Individual case studies
  - significant details, contextualization, and nuance
  - little/no comparative analysis
  - limited (and *premature*) generalization
- Few studies that examine *multiple OSSD projects* in *multiple domains*
  - Such studies offer higher degree of comparative analyses and generalization of results
    - Scacchi 2002: *requirements processes* in FOSSD projects for (a) Internet infrastructure (b) networked computer games, (c) astrophysics, and (d) academic software engineering

Research methodology

- Comparative (case) studies
  - Multiple open software development projects
    - Within and across multiple communities
- Qualitative techniques
- Analyzing and modeling
  - development processes
  - work practices and roles
  - development artifacts and tools
  - community structures and process dynamics

Individual participation

Individual participation in FOSSD projects: motives and consequences

- FOSS developers want to:
  - learn about new tools, techniques, skills, etc.
  - have fun building software
  - exercise their technical skill
  - try out new kinds of systems to develop
  - interconnect multiple FOSSD projects
- FOSS developers frequently:
  - build trust and reputation with one another
  - achieve “geek fame” (for project leaders)
  - spend more time reading online documents and communicating with one another than writing code

Resources supporting FOSS activities

FOSSD resources/capabilities

- Personal software development resources
- Beliefs supporting FOSSD
- FOSSD informalisms
- Skilled, self-organizing developers
- Discretionary time and effort
- Trust and social accountability

Personal software development resources

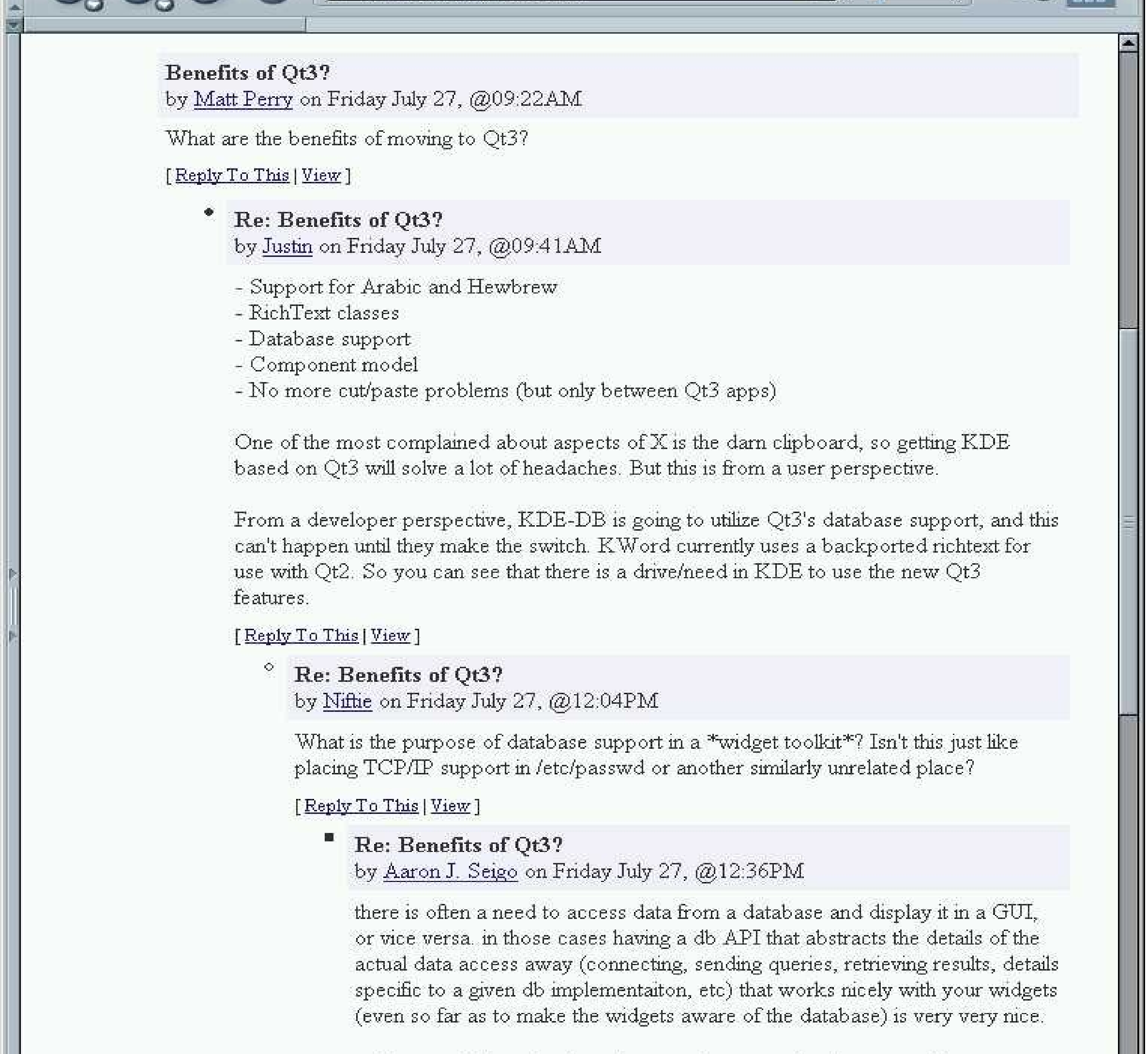
- Sustained commitment of personal resources helps *subsidize* FOSSD projects
  - Personal computer(s)
  - Internet access
  - Hosting personal Web site
  - Hosting project repositories
  - Personal choice of software development tools or tool set

Beliefs supporting FOSSD

- *Freedom of expression*
  - What to develop or work on
  - How to develop it
  - What tools to employ
- *Freedom of choice*
  - When to release work products
  - Expressing what can be said to whom with or without reservation

FOSSD Informalisms

- Software *informalisms*--artifacts participants use to describe, proscribe, or prescribe what’s happening in a project
- Informalisms capture detailed rationale and debates for what changes were made in particular development activities, artifacts, or source code files



Benefits of Qt3?
by Matt Perry on Friday July 27, @09:22AM

What are the benefits of moving to Qt3?

[Reply To This | View]

- **Re: Benefits of Qt3?**
  by Justin on Friday July 27, @09:41AM

  - Support for Arabic and Hewbrew
  - RichText classes
  - Database support
  - Component model
  - No more cut/paste problems (but only between Qt3 apps)

  One of the most complained about aspects of X is the darn clipboard, so getting KDE based on Qt3 will solve a lot of headaches. But this is from a user perspective.

  From a developer perspective, KDE-DB is going to utilize Qt3's database support, and this can't happen until they make the switch. KWord currently uses a backported richtext for use with Qt2. So you can see that there is a drive/need in KDE to use the new Qt3 features.

  [Reply To This | View]

  - **Re: Benefits of Qt3?**
    by Niftie on Friday July 27, @12:04PM

    What is the purpose of database support in a *widget toolkit*? Isn't this just like placing TCP/IP support in /etc/passwd or another similarly unrelated place?

    [Reply To This | View]

    - **Re: Benefits of Qt3?**
      by Aaron J. Seigo on Friday July 27, @12:36PM

      there is often a need to access data from a database and display it in a GUI, or vice versa. in those cases having a db API that abstracts the details of the actual data access away (connecting, sending queries, retrieving results, details specific to a given db implementaiton, etc) that works nicely with your widgets (even so far as to make the widgets aware of the database) is very very nice.

FOSSD informalisms

Email lists	Discussion forums	News postings	Project digests
IM/Internet Relay Chat	Scenarios of usage	How-to guides	To-do lists
FAQ’s and item lists	Project Wikis	System documentation	External publications
Copyright licenses	Architecture diagrams	Intra-app scripting	Plug-ins
Code from other projects	Project Web site	Multi-project Web sites	Project source code web
Project repositories	Software bug reports	Issue tracking databases	etc.

Skilled, self-organizing developers

- Successfully developing an open architecture system requires prior experience
- Organizing project work as a *virtual organization*
  - Skill-based meritocracy
  - Informal rules of governance and control, but rules are readily recognized by participants
  - Control incorporated into software and informalisms
    - How, where, and when to access data via APIs, UIs, and other architectural features

Discretionary time and effort

- Self-determination
  - work on what's interesting
- Peer recognition
  - becoming a social gateway
- Project affiliation or identification
- Self-promotion
  - How to realize career advancement
- Belief in inherent value of FOSS

Trust and social accountability

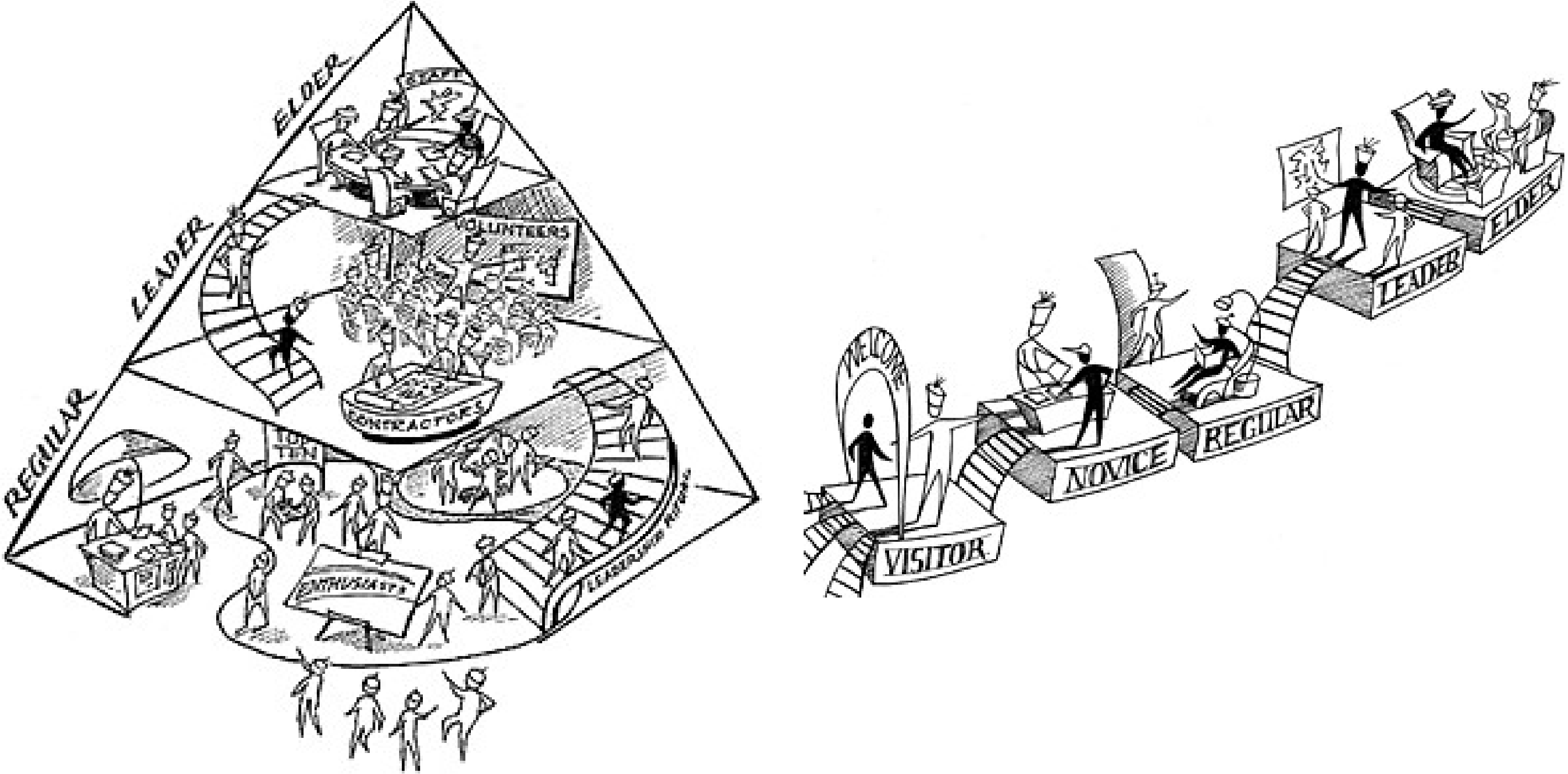
- *Social capital* accrues via:
  - Assuming ownership of a FOSS module
  - Voting on approval of other's actions
  - Shared peer reviewing
  - Contributing "gifts" that are reusable
- Accrued social capital is used to mitigate conflicts and accommodate resolutions
- Sustained social capital enables social networking externalities

Cooperation, coordination, and control in FOSSD projects

Software version control

- Enables stabilization and synchronization of dispersed, invisible FOSSD work
- SVC tools (CVS, SVN, Git, etc.) used as:
  - Central mechanism coordinating development
  - Online venue for mediating control over what changes will be accommodated
  - Gentle but sufficient social control mechanism that constrains overall project complexity

A meritocractic role hierarchy and role migration paths for FOSSD

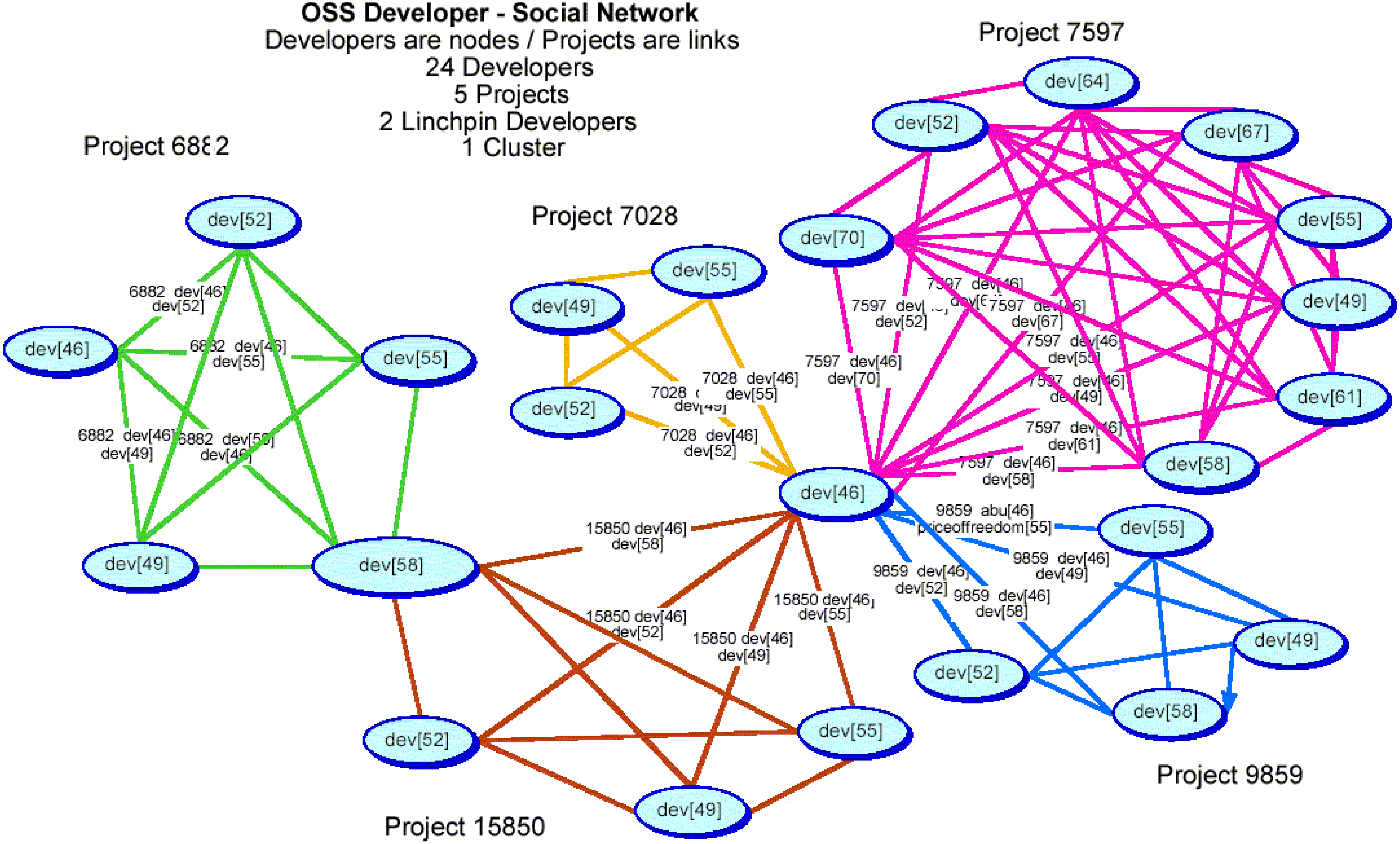


(images from A.J. Kim, *Community Building on the Web*, 2000)

Implicit project management

- FOSSD projects self-organize as a *meritocractic role-hierarchy* and *virtual project management*
  - Meritocracies embrace incremental innovations over radical innovations
  - VPM requires people to act in leadership roles based on skill, availability, and belief in project community
- Reliance on evolving web of software informalism content constrains collective action within FOSSD project

Alliances, social networking, and community development



OSS Developer - Social Network
Developers are nodes / Projects are links
24 Developers
5 Projects
2 Linchpin Developers
1 Cluster

Source: G. Madey, *et al.*, 2005

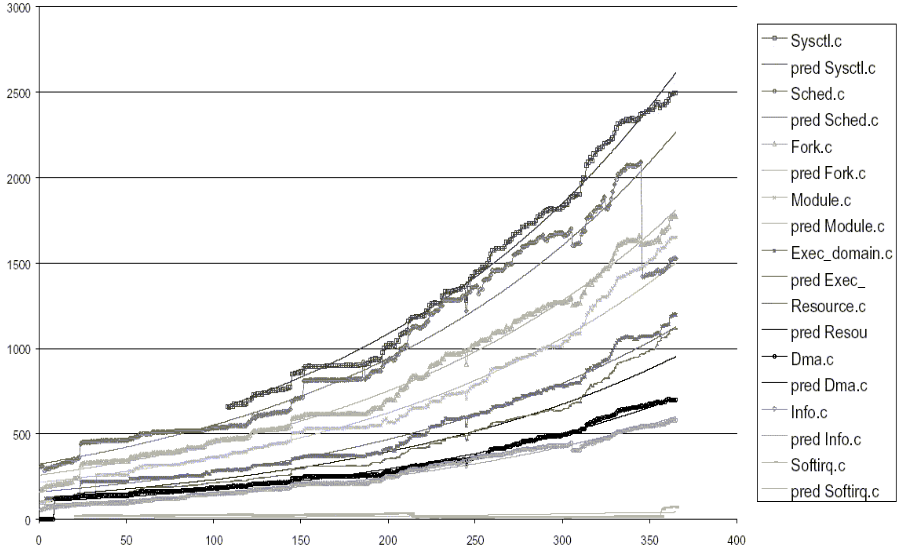
Community networking

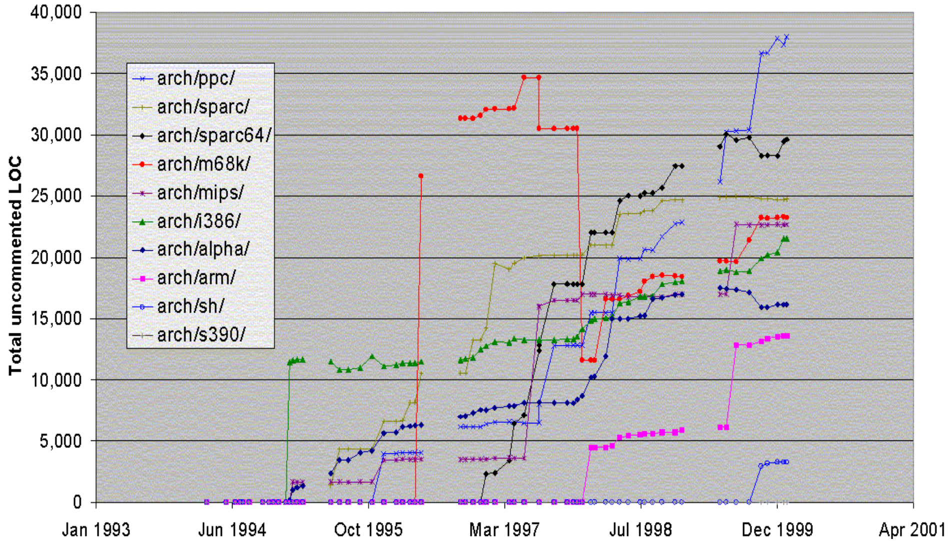
- Becoming a central node in a network of FOSS developers increases social capital
  - *Linchpin developers* as social gateways
  - Sharing beliefs, tools, artifacts enables shared experience, camaraderie, collective learning
- Multi-project clustering enables small projects to merge into sustainable projects
- Intellectual property regime fosters alignment and alliance with other projects and organizations

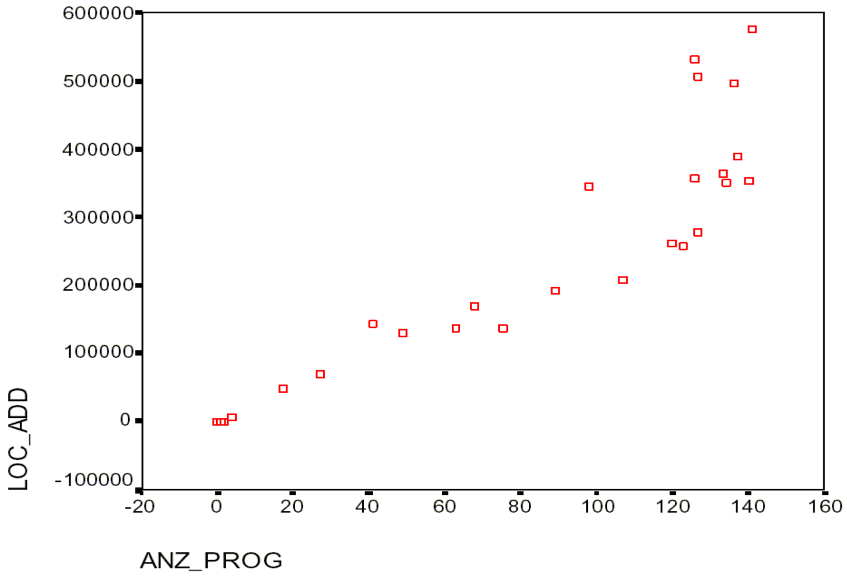
FOSS as multi-project software ecosystems

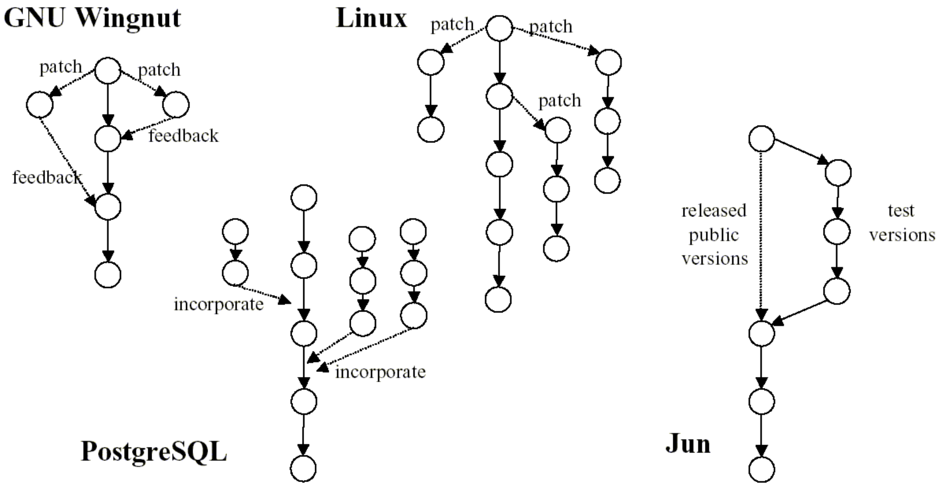
Multi-project software ecosystem

- Mutually dependent FOSS development and evolution propagate architectural styles, dependencies, and vulnerabilities
- *Architectural bricolage* arises when autonomous FOSSD projects, artifacts, tools, and systems co-mingle or merge
  - Enables discontinuous or exponential growth of FOSS source code, functionality, complexity, contributions


3000
2500
2000
1500
1000
500
0
0
50
100
150
200
250
300
350
400
Sysctl.c
pred Sysctl.c
Sched.c
pred Sched.c
Fork.c
pred Fork.c
Module.c
pred Module.c
Exec_domain.c
pred Exec_
Resource.c
pred Resou
Dma.c
pred Dma.c
Info.c
pred Info.c
Softirq.c
pred Softirq.c


Total uncommented LOC
40,000
35,000
30,000
25,000
20,000
15,000
10,000
5,000
0
Jan 1993
Jun 1994
Oct 1995
Mar 1997
Jul 1998
Dec 1999
Apr 2001
arch/ppc/
arch/sparc/
arch/sparc64/
arch/m68k/
arch/mips/
arch/i386/
arch/alpha/
arch/arm/
arch/sh/
arch/s390/


LOC_ADD
600000
500000
400000
300000
200000
100000
0
-100000
-20
0
20
40
60
80
100
120
140
160
ANZ_PROG


GNU Wingnut
patch
patch
feedback
feedback
Linux
patch
patch
patch
PostgreSQL
incorporate
incorporate
released public versions
test versions
Jun

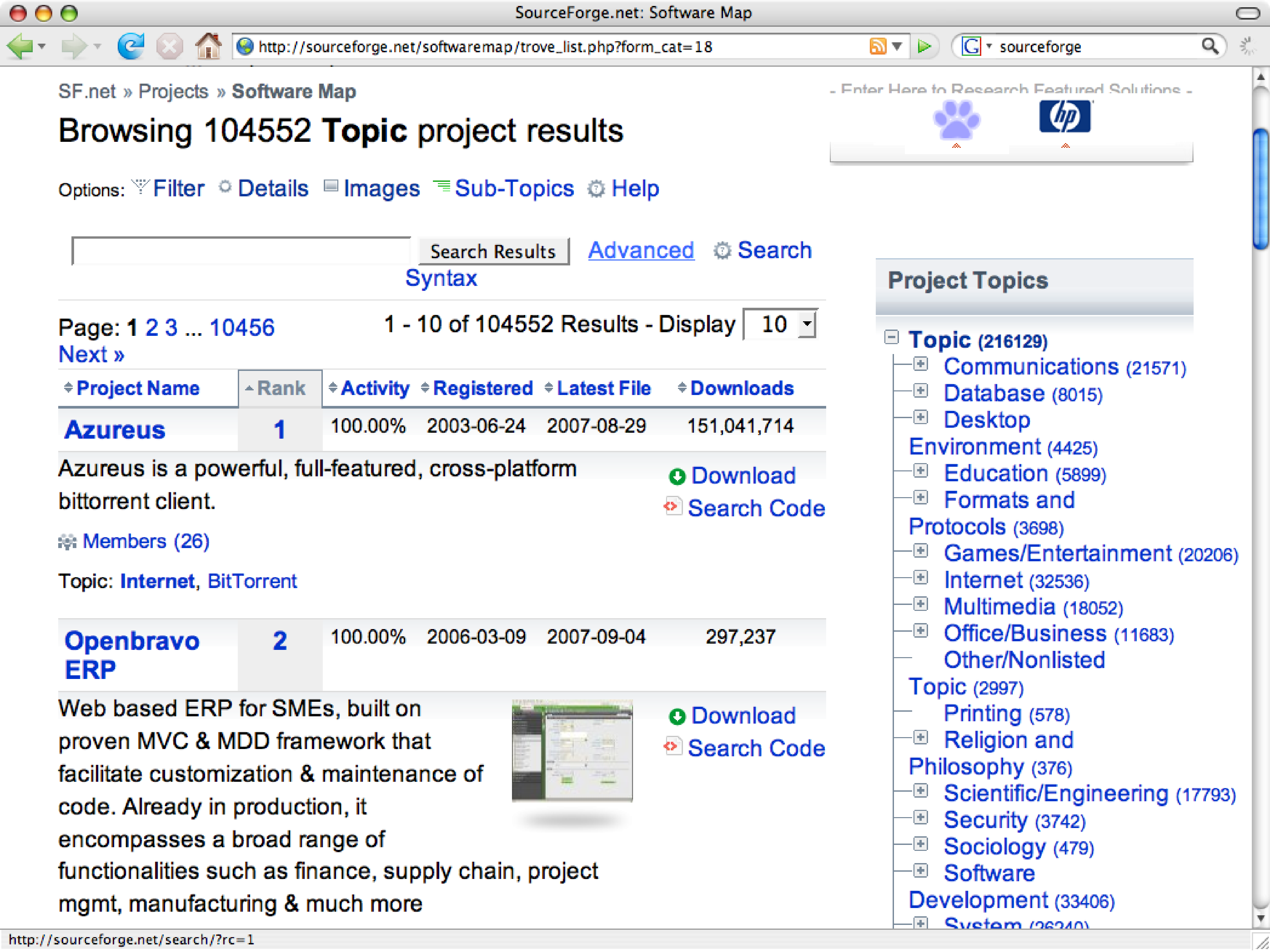
Evolutionary redevelopment, reinvention, and redistribution

- Overall evolutionary dynamic of many FOSSD projects is *reinvention and redevelopment*
  - Reinvention enables continuous improvement and collective learning
- FOSS evolve through minor mutations
  - Expressed, recombined, redistributed via incremental releases
- FOSS systems *co-evolve* with their development community
  - Success of one depends on the success of the other

FOSS as social movement

FOSS as social movement

- Free/OSS property regimes and licenses
  - Reiterate and institutionalize FOSS *culture* (values, norms, and beliefs)
    - GNU Public License (GPL) for *free* software
    - More than 50 other open source licenses (http://opensource.org)
    - "Creative Commons" Project at Stanford Law School developing public license framework
- FOSS spanning multiple disciplines and institutions


SourceForge.net: Software Map
http://sourceforge.net/softwaremap/trove_list.php?form_cat=18
sourceforge
- Enter Here to Research Featured Solutions -
http://sourceforge.net/search/?rc=1

SF.net » Projects » **Software Map**

Browsing 104552 **Topic** project results

Options: Filter Details Images Sub-Topics Help

Search Results Advanced Search Syntax

Page: **1** 2 3 ... 10456 Next »

1 - 10 of 104552 Results - Display 10

Project Name	Rank	Activity	Registered	Latest File	Downloads
Azureus	**1**	100.00%	2003-06-24	2007-08-29	151,041,714

Azureus is a powerful, full-featured, cross-platform bittorrent client.

Download
Search Code

Members (26)

Topic: **Internet**, BitTorrent

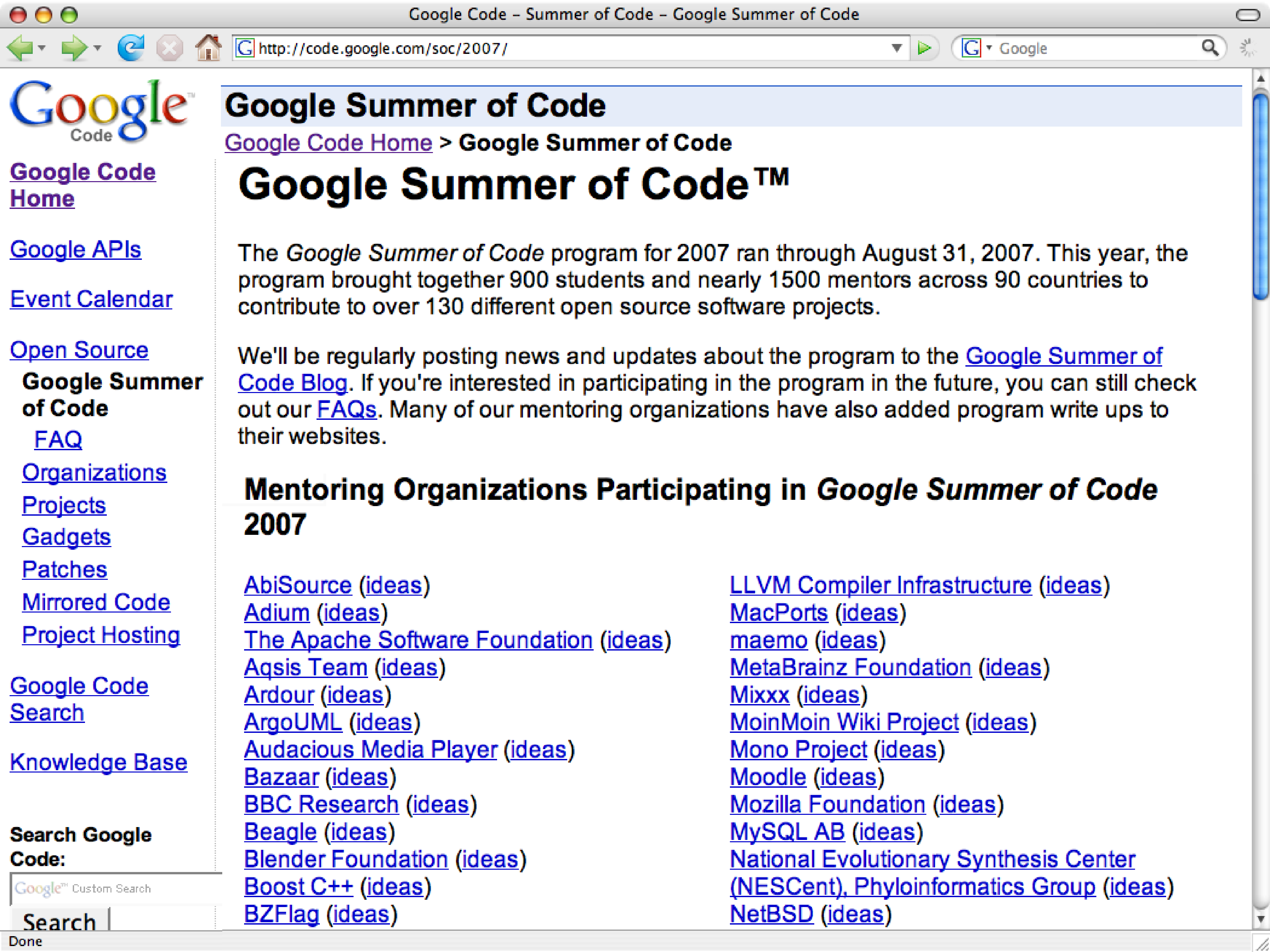
Project Name	Rank	Activity	Registered	Latest File	Downloads
Openbravo ERP	**2**	100.00%	2006-03-09	2007-09-04	297,237

Web based ERP for SMEs, built on proven MVC & MDD framework that facilitate customization & maintenance of code. Already in production, it encompasses a broad range of functionalities such as finance, supply chain, project mgmt, manufacturing & much more

Download
Search Code

Project Topics

- **Topic** (216129)
  - Communications (21571)
  - Database (8015)
  - Desktop Environment (4425)
  - Education (5899)
  - Formats and Protocols (3698)
  - Games/Entertainment (20206)
  - Internet (32536)
  - Multimedia (18052)
  - Office/Business (11683)
  - Other/Nonlisted Topic (2997)
  - Printing (578)
  - Religion and Philosophy (376)
  - Scientific/Engineering (17793)
  - Security (3742)
  - Sociology (479)
  - Software Development (33406)
  - System (26240)



Google Summer of Code

Google Code Home > Google Summer of Code

Google Summer of Code™

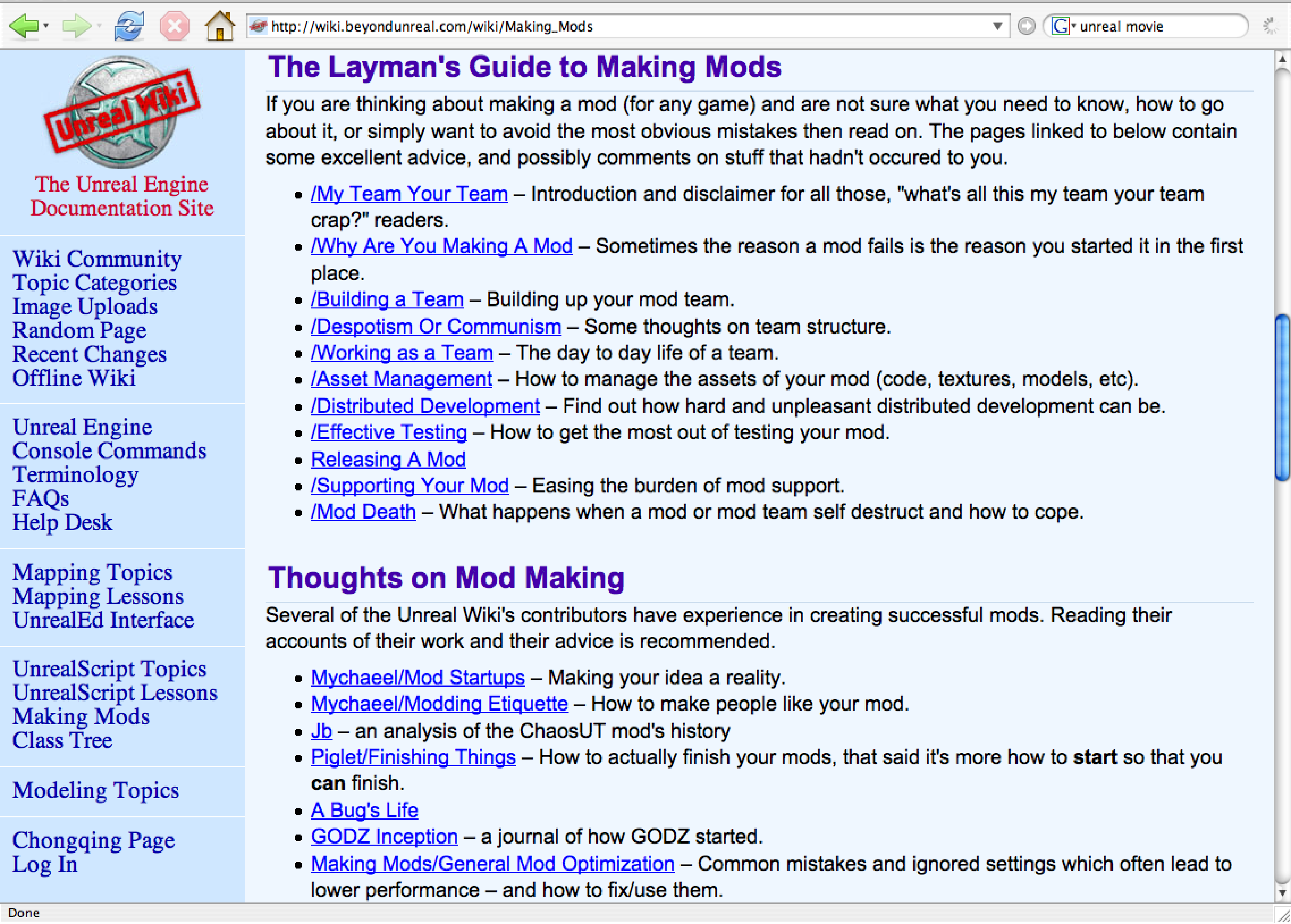
The *Google Summer of Code* program for 2007 ran through August 31, 2007. This year, the program brought together 900 students and nearly 1500 mentors across 90 countries to contribute to over 130 different open source software projects.

We'll be regularly posting news and updates about the program to the Google Summer of Code Blog. If you're interested in participating in the program in the future, you can still check out our FAQs. Many of our mentoring organizations have also added program write ups to their websites.

Mentoring Organizations Participating in *Google Summer of Code* 2007

AbiSource (ideas)
Adium (ideas)
The Apache Software Foundation (ideas)
Aqsis Team (ideas)
Ardour (ideas)
ArgoUML (ideas)
Audacious Media Player (ideas)
Bazaar (ideas)
BBC Research (ideas)
Beagle (ideas)
Blender Foundation (ideas)
Boost C++ (ideas)
BZFlag (ideas)

LLVM Compiler Infrastructure (ideas)
MacPorts (ideas)
maemo (ideas)
MetaBrainz Foundation (ideas)
Mixxx (ideas)
MoinMoin Wiki Project (ideas)
Mono Project (ideas)
Moodle (ideas)
Mozilla Foundation (ideas)
MySQL AB (ideas)
National Evolutionary Synthesis Center (NESCent), Phyloinformatics Group (ideas)
NetBSD (ideas)



The Layman's Guide to Making Mods

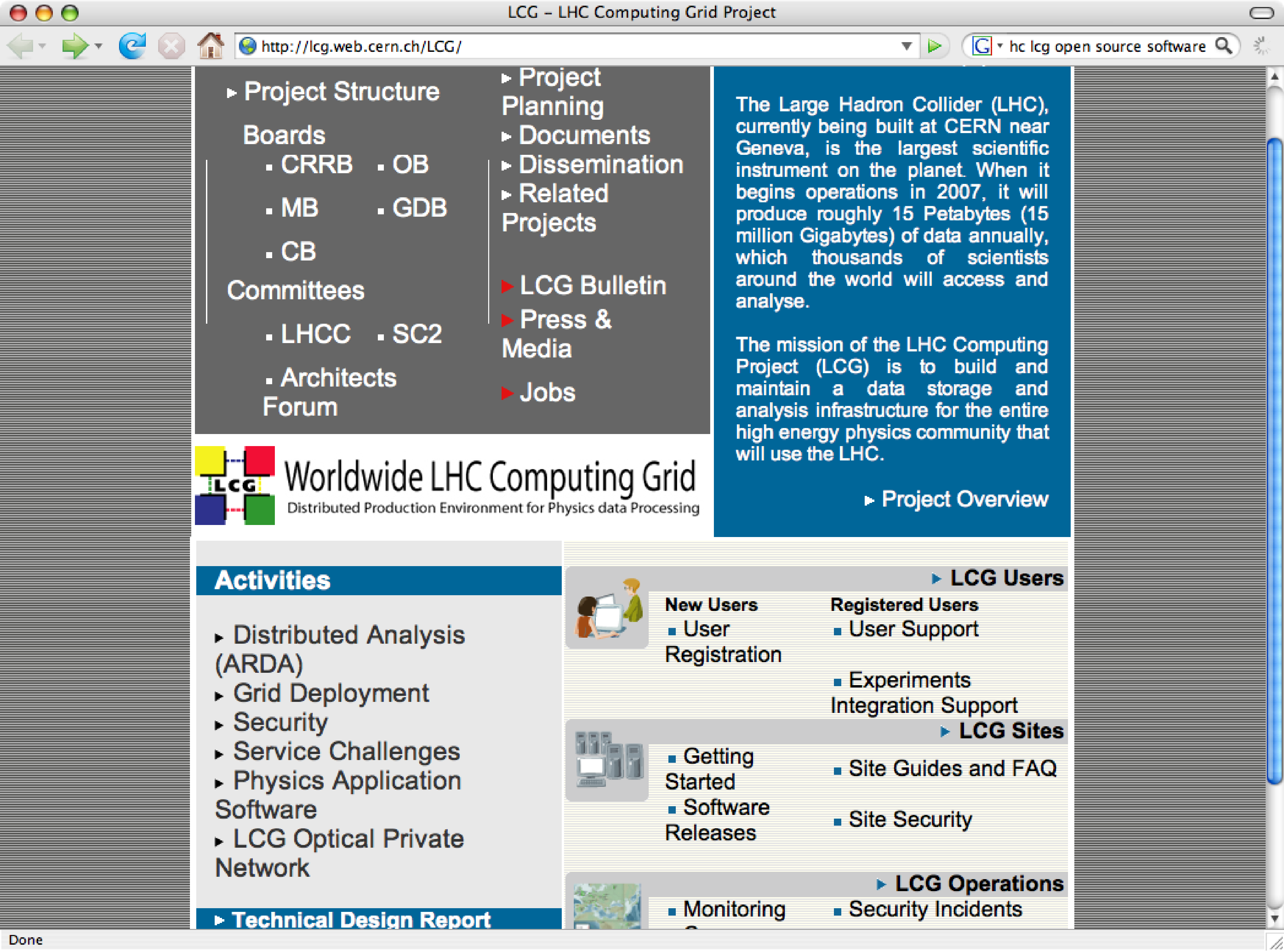
If you are thinking about making a mod (for any game) and are not sure what you need to know, how to go about it, or simply want to avoid the most obvious mistakes then read on. The pages linked to below contain some excellent advice, and possibly comments on stuff that hadn't occured to you.

- /My Team Your Team – Introduction and disclaimer for all those, "what's all this my team your team crap?" readers.
- /Why Are You Making A Mod – Sometimes the reason a mod fails is the reason you started it in the first place.
- /Building a Team – Building up your mod team.
- /Despotism Or Communism – Some thoughts on team structure.
- /Working as a Team – The day to day life of a team.
- /Asset Management – How to manage the assets of your mod (code, textures, models, etc).
- /Distributed Development – Find out how hard and unpleasant distributed development can be.
- /Effective Testing – How to get the most out of testing your mod.
- Releasing A Mod
- /Supporting Your Mod – Easing the burden of mod support.
- /Mod Death – What happens when a mod or mod team self destruct and how to cope.

Thoughts on Mod Making

Several of the Unreal Wiki's contributors have experience in creating successful mods. Reading their accounts of their work and their advice is recommended.

- Mychaeel/Mod Startups – Making your idea a reality.
- Mychaeel/Modding Etiquette – How to make people like your mod.
- Jb – an analysis of the ChaosUT mod's history
- Piglet/Finishing Things – How to actually finish your mods, that said it's more how to **start** so that you **can** finish.
- A Bug's Life
- GODZ Inception – a journal of how GODZ started.
- Making Mods/General Mod Optimization – Common mistakes and ignored settings which often lead to lower performance – and how to fix/use them.


LCG – LHC Computing Grid Project
http://lcg.web.cern.ch/LCG/
hc lcg open source software
Done

- Project Structure

Boards

- CRRB
- OB
- MB
- GDB
- CB

Committees

- LHCC
- SC2
- Architects Forum

- Project Planning
- Documents
- Dissemination
- Related Projects

- LCG Bulletin
- Press & Media
- Jobs

Worldwide LHC Computing Grid

Distributed Production Environment for Physics data Processing

The Large Hadron Collider (LHC), currently being built at CERN near Geneva, is the largest scientific instrument on the planet. When it begins operations in 2007, it will produce roughly 15 Petabytes (15 million Gigabytes) of data annually, which thousands of scientists around the world will access and analyse.

The mission of the LHC Computing Project (LCG) is to build and maintain a data storage and analysis infrastructure for the entire high energy physics community that will use the LHC.

- Project Overview

Activities

- Distributed Analysis (ARDA)
- Grid Deployment
- Security
- Service Challenges
- Physics Application Software
- LCG Optical Private Network

- Technical Design Report

LCG Users

New Users

- User Registration

Registered Users

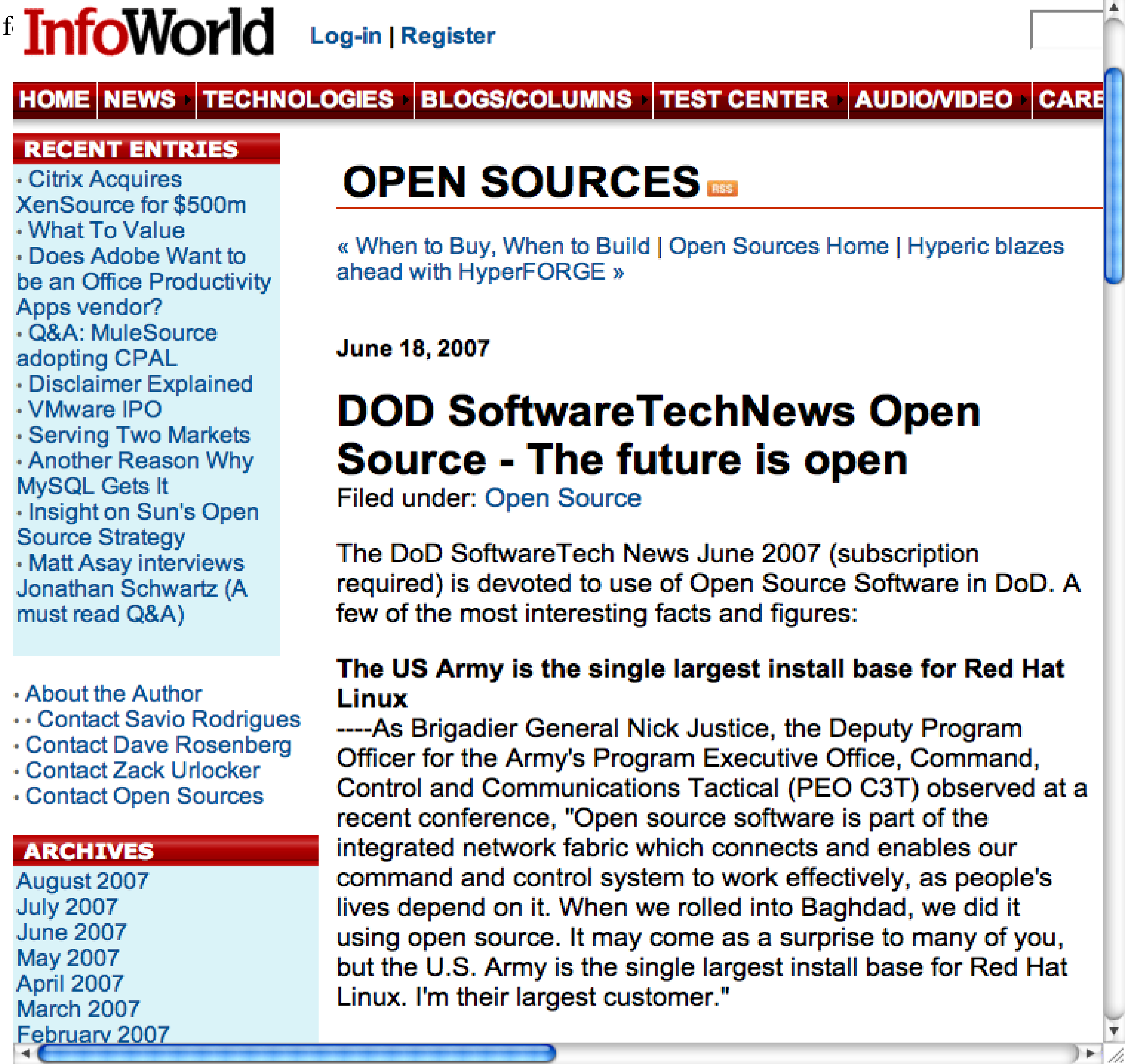
- User Support
- Experiments Integration Support

LCG Sites

- Getting Started
- Software Releases
- Site Guides and FAQ
- Site Security

LCG Operations

- Monitoring
- Security Incidents



RECENT ENTRIES

- Citrix Acquires XenSource for $500m
- What To Value
- Does Adobe Want to be an Office Productivity Apps vendor?
- Q&A: MuleSource adopting CPAL
- Disclaimer Explained
- VMware IPO
- Serving Two Markets
- Another Reason Why MySQL Gets It
- Insight on Sun's Open Source Strategy
- Matt Asay interviews Jonathan Schwartz (A must read Q&A)

- About the Author
- • Contact Savio Rodrigues
- Contact Dave Rosenberg
- Contact Zack Urlocker
- Contact Open Sources

ARCHIVES

August 2007
July 2007
June 2007
May 2007
April 2007
March 2007
February 2007

OPEN SOURCES

« When to Buy, When to Build | Open Sources Home | Hyperic blazes ahead with HyperFORGE »

June 18, 2007

DOD SoftwareTechNews Open Source - The future is open

Filed under: Open Source

The DoD SoftwareTech News June 2007 (subscription required) is devoted to use of Open Source Software in DoD. A few of the most interesting facts and figures:

The US Army is the single largest install base for Red Hat Linux

----As Brigadier General Nick Justice, the Deputy Program Officer for the Army's Program Executive Office, Command, Control and Communications Tactical (PEO C3T) observed at a recent conference, "Open source software is part of the integrated network fabric which connects and enables our command and control system to work effectively, as people's lives depend on it. When we rolled into Baghdad, we did it using open source. It may come as a surprise to many of you, but the U.S. Army is the single largest install base for Red Hat Linux. I'm their largest customer."

FOSS as social movement

- Emerging as a global-scale socio-technical movement that increasingly permeates society at an institutional, governmental, and international level in ways no prior software development regime has previously achieving.
- Unlikely any company/nation can inhibit FOSS in the near-term

Discussion and limitations

Defining characteristics of FOSSD projects

- Public availability of project data and artifacts
  - – Collecting FOSSD process data may be more cost effective compared to proprietary SE projects
  - – *Prediction*: growing share of empirical SE research will be performed using FOSS data

FOSSD research limitations

- Individual participation
  - Some form of reciprocity and intrinsic, self-serving motivation is necessary
- Cooperation, coordination, and control
  - Negotiation and conflict management are part of the cost FOSS developers incur in order to have their believes fulfilled
  - Time, effort, and attention are spent negotiating socio-technical dependencies

FOSSD research limitations

- Alliances and community development
  - FOSSD projects give rise to new kinds of requirements for community building, community software, and community information sharing systems
  - Alliances and community require attention to sustain their effectiveness, and to prevent them from becoming self-serving and bureaucratic

FOSSD research limitations

- Empirical studies of FOSSD are expanding the scope of what we can observe, discover, analyze, and learn about large software systems.
  - Mining software repositories
  - Multi-modal modeling and analysis of socio-technical processes and networks found in sustained FOSSD projects

Research opportunities

Research opportunities

- FOSSD is poised to alter the calculus of empirical SE
  - Software process discovery, modeling, and simulation
  - Repository mining can support software visualization, refactoring/redesign studies
  - Comparison of SE versus FOSSD approaches to software inspection and peer review

Research opportunities

- Based on results from individual motivation, participation, role migration, and turnover in FOSSD projects, SE world would benefit from empirical studies that examine similar patterns in conventional software development projects
  - Is FOSSD more fun, interesting, and rewarding than SE?

Research opportunities

- Conventional software cost estimation techniques (e.g., “total cost of operation”) slight/ignore social capital and socio-technical resources
  - Miscalculation of total resources and capabilities that affect predicted/actual costs of software development or FOSSD

Research opportunities

- Results from study of cooperation, coordination and control in FOSSD
  - Virtual project management and role migration can provide a lightweight approach to SE project management
  - Unclear whether proprietary software projects willing to embrace VPM

Research opportunities

- Alliance formation and social networking results suggest SE projects operate at a disadvantage compared to FOSSD projects
  - SE projects tend to produce systems whose growth/evolution is limited
  - FOSSD projects can produce systems capable of sustained exponential growth/evolution of both software and developer-user community

Research opportunities

- How best to encourage the emergence of a social movement that combines best practices of FOSSD and SE
  - Consider participation or study of open source software engineering (OSSE) projects at Tigris.org
  - OSSE seeks to combine SE and FOSSD tools, techniques, and concepts

Project tools
- Project home
- Project News
- Bugs and Issues
- Cookbook
- Project Membership
- Mailing lists
- File Sharing
- Source code
- Project FAQ
- Developer Zone

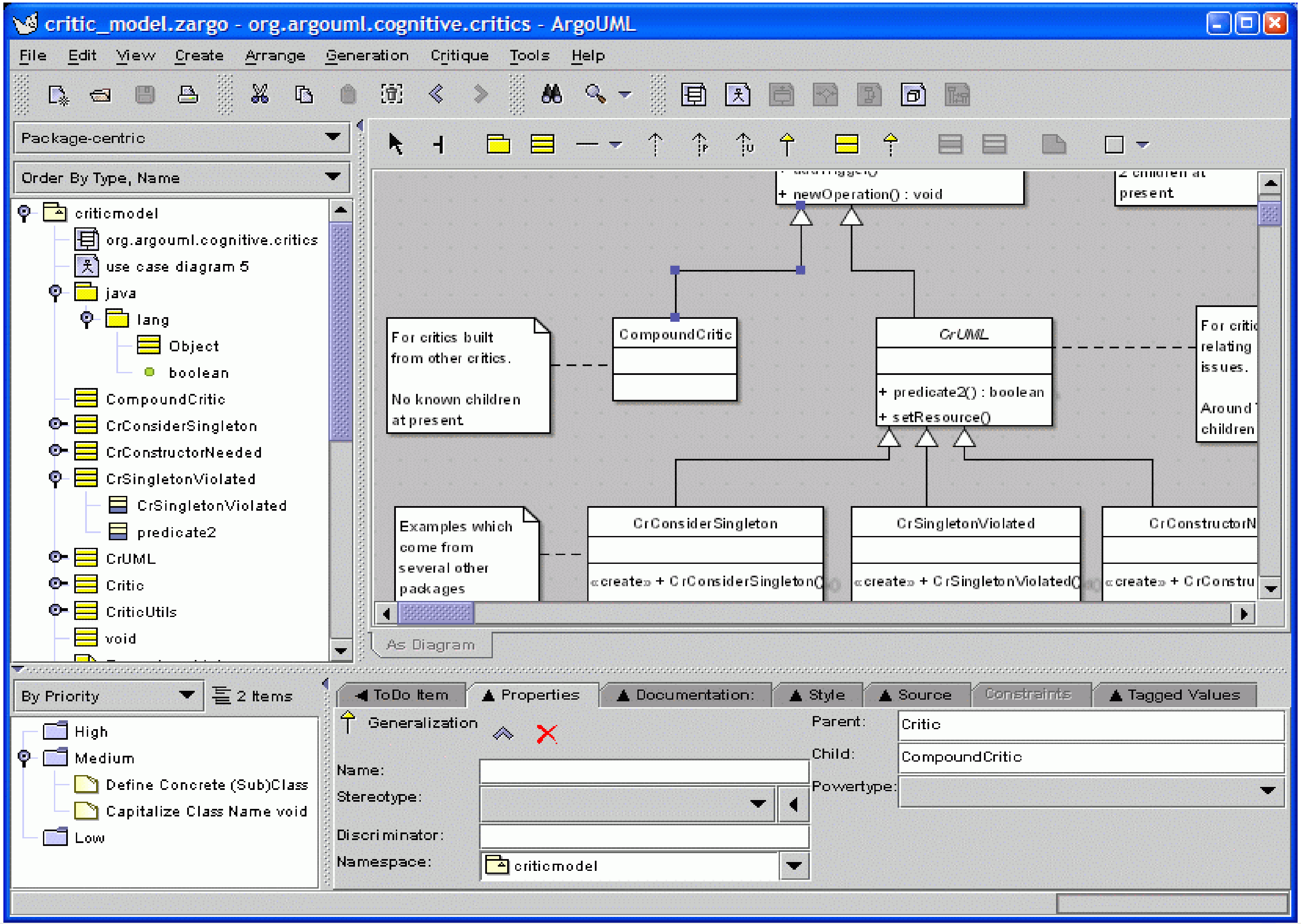
Using ArgoUML
- Quick guide
- User Manual
- FAQ
- Documentation
- Tour
- Downloads

Search
This project
Go
Advanced search

How do I...
- Get help?

Category	Featured projects
scm	Subversion, RapidSVN, TortoiseSVN
issuetrack	Scarab
requirements	xmlbasedsrs
design	ArgoUML
techcomm	eyebrowse, binarycloud
construction	phpcreate,

ArgoUML Main Window



For critics built from other critics.

No known children at present

For critic relating issues.

Around children

Examples which come from several other packages

Generalization

Name:

Stereotype:

Discriminator:

Namespace: criticmodel

Parent: Critic

Child: CompoundCritic

Powertype:

Detailed study report available

- W. Scacchi, Free/Open Source Software Development: Recent Research Results and Methods, in M. Zelkowitz (ed.), *Advances in Computers*, Vol. 69, 243-295, 2007.
- http://www.ics.uci.edu/~wscacchi/Papers/New/Draft_Chapter_Scacchi.pdf

Acknowledgements

- *Project collaborators*:
  - Mark Ackerman, UMichigan, Ann Arbor
  - Les Gasser, UIllinois, Urbana-Champaign
  - John Noll, Santa Clara University
  - Margaret Ellliot, Chris Jensen, UCI-ISR
  - and others ar ISR
- *Funding support*:
  - National Science Foundation: #0083075, #0205679, #0205724, #0350754 and #0534771.
  - Digital Industry Promotion (DIP) Agency, Global R&D Collaboration Center, Daegu, South Korea
  - No endorsement implied.