
CVSAnalY

An analysis tool for your CVS repository

Gregorio Robles, Alvaro Navarro
{grex, anavarro}@gsyc.esct.urjc.es

February 26th 2005



Why public analysis?

- ✓ No central authority (who is boss?)
- ✓ We don't interfere in the project
- ✓ Geographical dispersion (user and developers)
- ✓ projects generates data:
 - ✓ source repositories
 - ✓ mailing list
 - ✓ bugzilla

CVS: Concurrent Version System

Why CVS?

- ✓ Meeting point for users and developers
- ✓ actual release and previous versions available
- ✓ We can see the life of the project with logs files
- ✓ SVN and OpenCVS

CVSAnaly implementation

- ✓ Python code
- ✓ independent modules.
- ✓ MySQL
- ✓ plugins possibility

How does CVSAAnaly work?

Three main steps:

- ✓ Preprocessing:
 - ✓ Download Source code and logs
 - ✓ Parse logs (commiters, dates, file types, LOCs)
- ✓ Database insertion
- ✓ Postprocessing:
 - ✓ Graphs
 - ✓ Social Networks Analyze
 - ✓ Statistical webs

Which results are available?

- ✓ Numbre of modules, committers and commits
 - ✓ files added or remove
 - ✓ lines added or remove
- ✓ Historical data
 - ✓ number of modules now
 - ✓ number of modules one year ago, two years ago...
- ✓ *Generations* of developers

CVSanaly for the final user

CVSanaly philosophy: anyone can analyze his project

- ✓ Easy to use
 - ✓ Wizard to make config file (ncurses / gtk)

- ✓ Easy to distribute
 - ✓ Tarball
 - ✓ Debian package
 - ✓ FreeBSD port

Conclusions

- ✓ Remote and unattended analysis
- ✓ Global view of the software projects
- ✓ Any kind of libre software analysis should be justified with empirical data

Future work

- ✓ Integrated analysis: mailing list and bug tracking system
- ✓ Fixed bugs ;-)
- ✓ Documentation (configuration, FAQ, Howtos..)
- ✓ Packaging (deb, rpm, port..)