



Open Source Tools and the Software Engineering Process

Stephen L Arnold
Gentoo Linux Developer

sci, dev-tools, comm-fax, ppc, embedded
Lunch time Tech Talk, January 22, 2009

stephen.arnold@acm.org

nerdboy@gentoo.org



What Tools?

- Emulators and interface tools
- Configuration / document management
- Text processing – from sed to python
- Documentation and XML tools
- Reverse engineering and design analysis
- Static and dynamic code analysis/profiling
- Compilers, libraries, development environments
- Modeling, workflow, and process support



Basic Analysis Process

- Review/pre-process and configure source
 - Plan branch/tag strategy in advance
 - Checkouts start with head/trunk
- Run metrics and documentation tools
 - Review stats and metrics for required language tools and potential problem areas
 - Use doxygen to extract design information
- Define a project
 - Use your favorite editor, IDE, or Eclipse
- Run static analysis tools (add stubs and instrument code as needed)
- Determine build/runtime dependencies



Software CM Tools

- Many open source tools available: CVS, SVN, Monotone, and more
- Excellent integration with other tools
 - Editors, IDEs, etc
 - Web interfaces, project management
 - Other SCM tools
- Use the tool capabilities to perform:
 - Process support
 - Basic analyses
 - Peer review
 - Status reporting



Source Code Processing

Stand-alone text-mode tools for statistics, metrics, and documentation

- Easy to run and automate
- Can produce voluminous output
- **Statistics and metrics**
 - SLOCCount – line counts for multiple languages, project statistics
 - CCCC – Statistics and several software metrics
- **Documentation and design extraction**
 - Doxygen – Source code (API) documentation, software design and structure diagrams (many output formats)
 - Robodoc, JavaDoc, other



Projects and Artifacts

- Editor or IDE? Use what works best...
 - emacs/Xemacs, vi(m), kate, other
 - Eclipse, Anjuta, Jdeveloper, Source Navigator, other
- Analyze developer project files
 - Know what they build against
- Define your own project
 - Source, header files
 - Toolchain and build system
 - Required libraries
- Document the architecture and interfaces
 - Compare to design artifacts
 - Annotate source code



Static Analysis Tools

Variety of tools to choose from depending on source language:

- Command-line tools, GUI tools, Eclipse plugins
 - Lint-style tools for Fortran, C, C++, etc
 - ftnchek, splint, cppcheck
 - Can incorporate in Makefiles, build hosts
- GUI tools
 - Stand-alone, eg, Grasp, or plugins for existing development platforms
- Eclipse (and other) plugins
 - Analysis – PMD, Findbugs, Jlint, CppChecker
 - Modeling – GMF, TOPCASED
 - Metrics, Visualization, Testing, Profiling, more...



Example Java Analysis

Example: Eclipse with Java code

- Configure, count, and document code
 - SVN w/ apache, viewvc, trac, mod_auth_*
 - Doxygen, SLOCCount, CCCC
- Create project
 - Eclipse Ganymede release with full JDT
 - Sun JDK 1.6, PMD, and FindBugs
 - Import sources, add libraries
 - Analyze build errors and any project artifacts
 - Add any additional libs/packages, repeat
 - Configure and run analysis tools
 - Analyze results and adjust as needed
 - Add annotations, modify analysis rules, create reports



Resources

- Statistics and metrics
 - SLOCCount - <http://www.dwheeler.com/sloccount/>
 - CCCC - <http://cccc.sourceforge.net/>
- Documentation and design extraction
 - Doxygen - <http://www.stack.nl/~dimitri/doxygen/>
 - JavaDoc - <http://java.sun.com/j2se/javadoc/>
- Eclipse - <http://www.eclipse.org>
 - Plugins - <http://www.eclipseplugincentral.com>
 - TOPCASED - <http://www.topcased.org>
- Software engineering tools at tigris.org
 - Subversion, clients, plugins
 - ArgoUML (supports DoDAF notation)



This work is an original work by Stephen L Arnold
<nerdboy@gentoo.org>

<<http://dev.gentoo.org/~nerdboy>>

Portions copyright 2009 Stephen L Arnold. Some rights reserved.

The Gentoo Linux logo is Copyright 2004 Gentoo Foundation, used with permission.

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike License. To view a copy of this license, visit <<http://creativecommons.org/licenses/by-nc-sa/1.0>> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.



Please contact Stephen L Arnold <stephen.arnold@acm.org> for commercial uses of this work.