# Open Source & Agile Software Development

Michael McLay

June 3, 2004

mclay@python.net

(A summary of the May 11th Workshop)

# All this in 15 Minutes!

- Collaborative Development Services
  - Richard Kilmer, InfoEther LLC
- The Business Case for Agile Languages
  - Stephen Ferg, Bureau of Labor Statistics
- How Python is Developed
  - Andrew Kuchling, Python Software Foundation
- OSS and Scientific Computing
  - Paul Barrett, Space Telescope Science Institute
- Computer programming for Everyone
  - Jeffrey Elkner, Yorktown High School

# Collaborative Development Services

Learning From the Open Source Agile Development Process

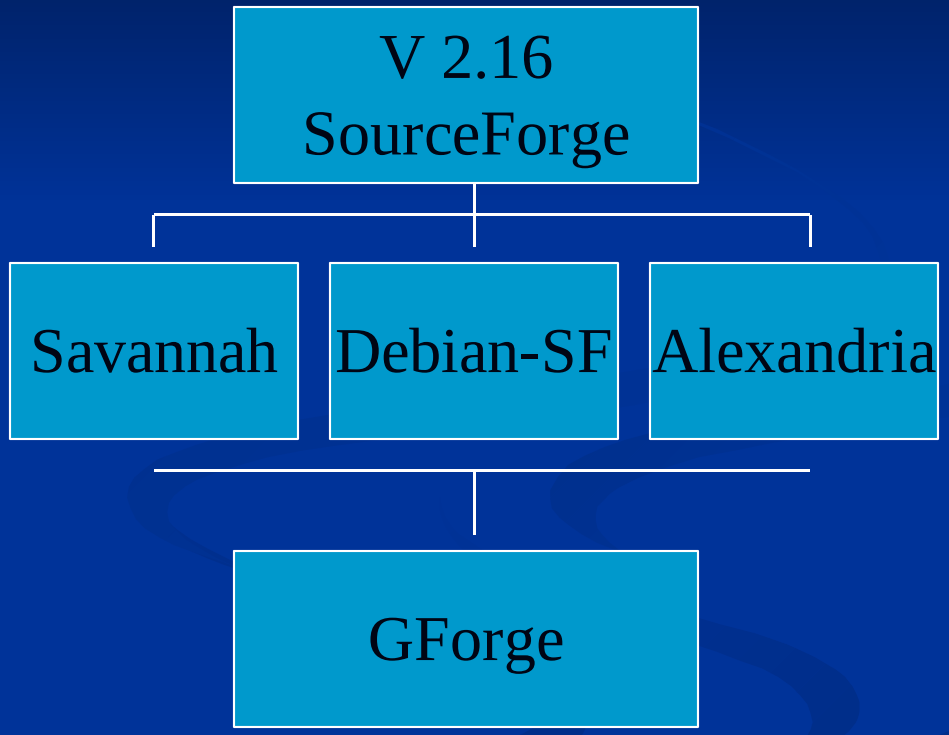
Richard Kilmer, InfoEther LLC

# SourceForge.net

- World's largest OSS development site
  - 80,000+ projects & 844,000+ registered developers
- Provides centralized location for OSS developers to control their development process
- Visited by thousands of end users for downloading software, participating in mailing lists, forums, etc
- Owned by the Open Source Developer Network, Inc (OSDN)

# The Emergence of GForge

- VA Software takes SourceForge commercial
- SourceForge fork
- Tim Perdue begins GForge
  - Simplify installation
  - Simplify User Interface
  - Releases/Support



# GForge Statistics

- At least 93 websites are using GForge around the world
- Scalable (47000 to 2 users)
- Some sites:
  - Philips
  - NASA Goddard Space Flight Center
  - NOAA
  - National Science Digital Library
  - DARPA (cougaar.org, semwebcentral.org)

Patches | Search | Log In | New Account

Home | My Page | Project Tree | Code Snippets | GForge

Summary | Forums | Tracker | Tasks | Docs | Surveys | News | SCM | Files

## Tracker: Patches

**Submit New** | **Browse** | **Admin**

Assignee: (?) Timothy Perdue | State: (?) Open | Category: (?) Any | Group: (?) Any | Changed: (?)

Order by: (?) ID | Ascending | Browse

| ID | Summary | Open Date | Assigned To | Submitted By |
|---|---|---|---|---|
| 551 | fix homedir_prefix and groupdir_prefix in cvs-cron | * 2003-10-14 10:58 | Timothy Perdue | Ramon van Alteren |
| 573 | Tracker: upload multiple attachment files | * 2003-11-02 03:48 | Timothy Perdue | Hidenari Miwa |
| 574 | Set any Data-type to any Tracker-type | * 2003-11-03 04:55 | Timothy Perdue | Hidenari Miwa |
| 629 | Preliminary WebDAV/SVN integration patch for comments | * 2003-12-11 03:24 | Timothy Perdue | Sung Kim |
| 749 | Patch for allowing multiple email addresses for Doc/Task/Tracker updates | * 2004-03-30 08:13 | Timothy Perdue | David Hirst |
| 761 | mod-limited.php patch for trackers. | * 2004-04-13 10:21 | Timothy Perdue | David Hirst |
| 790 | added resolution_select_box for tracker browse page | 2004-05-05 06:49 | Timothy Perdue | dominik haas |

* Denotes requests > 30 Days Old

**Priority Colors:**

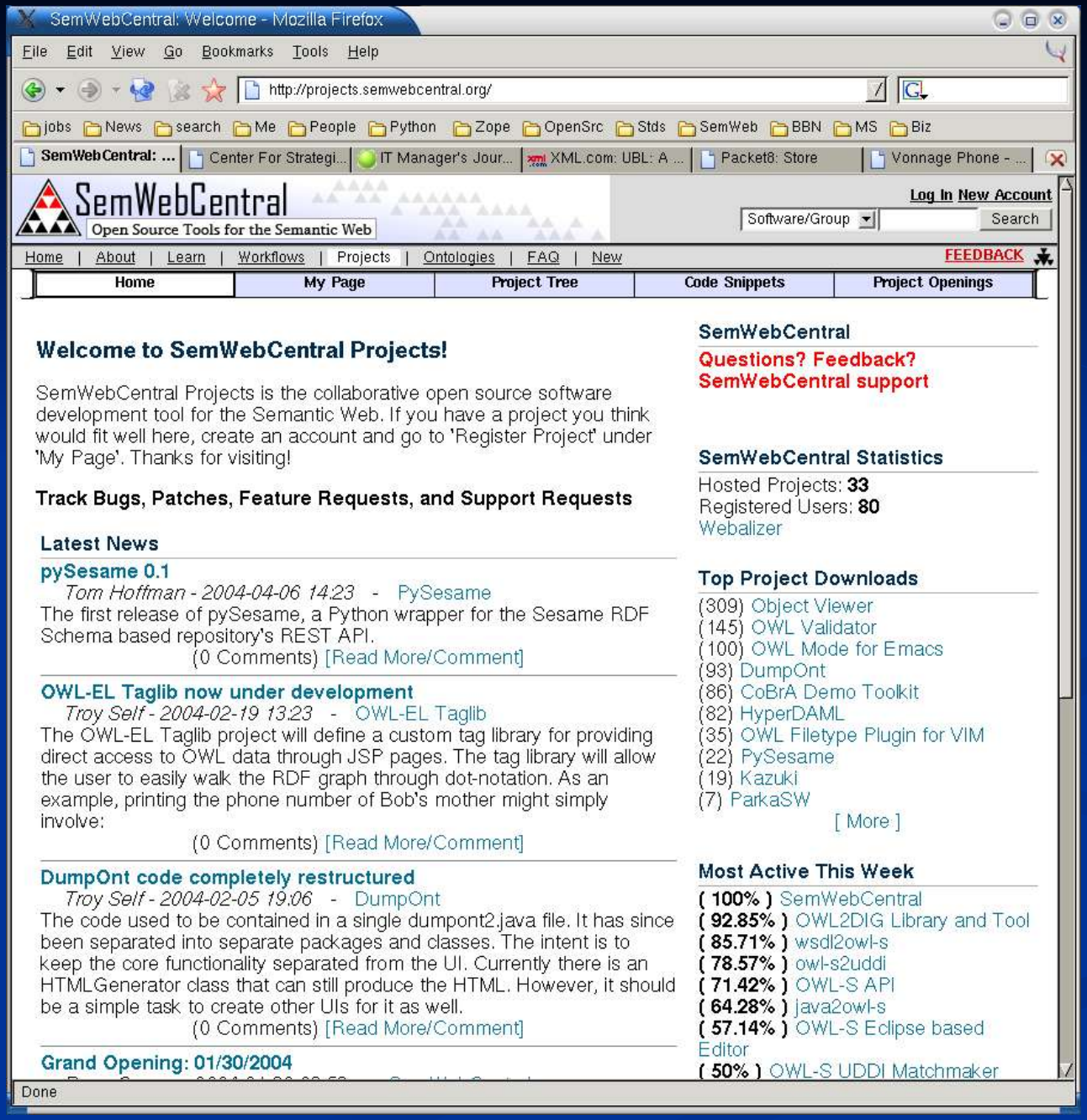
1 2 3 4 5 6 7 8 9

# GForge Capabilties

- Communications tools
  - Web forums
  - Mailing lists
  - News publishing system
- Tracking tools
  - Bugs and patches
  - Support requests
  - Task and project management tools
- Information sharing tools
  - Versioning system (CVS)
  - Shared “code snippets”
  - Documentation manager & File release system

# DARPA and GForge

- SemWebCentral.org (DAML)
  - Enabling service for the Semantic Web community
    - Project hosting (all GForge capabilities)
    - Learning about the Semantic Web
  - Semantic Web extensions to Gforge
- Cougaar.org (UltraLog)
  - Large-scale multiagent framework
  - Building survivable systems
  - 800,000+ lines of Java source
  - BBN Technologies led development effort
  - SOAP extensions to GForge
  - Optimizations to GForge database


SemWebCentral: Welcome - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://projects.semwebcentral.org/
jobs News search Me People Python Zope OpenSrc Stds SemWeb BBN MS Biz
SemWebCentral: ... Center For Strategi... IT Manager's Jour... XML.com: UBL: A ... Packet8: Store Vonnage Phone - ...
SemWebCentral
Open Source Tools for the Semantic Web
Log In New Account
Software/Group Search
Home | About | Learn | Workflows | Projects | Ontologies | FAQ | New
FEEDBACK
Home My Page Project Tree Code Snippets Project Openings
Welcome to SemWebCentral Projects!
SemWebCentral Projects is the collaborative open source software development tool for the Semantic Web. If you have a project you think would fit well here, create an account and go to 'Register Project' under 'My Page'. Thanks for visiting!
Track Bugs, Patches, Feature Requests, and Support Requests
Latest News
pySesame 0.1
Tom Hoffman - 2004-04-06 14:23 - PySesame
The first release of pySesame, a Python wrapper for the Sesame RDF Schema based repository's REST API.
(0 Comments) [Read More/Comment]
OWL-EL Taglib now under development
Troy Self - 2004-02-19 13:23 - OWL-EL Taglib
The OWL-EL Taglib project will define a custom tag library for providing direct access to OWL data through JSP pages. The tag library will allow the user to easily walk the RDF graph through dot-notation. As an example, printing the phone number of Bob's mother might simply involve:
(0 Comments) [Read More/Comment]
DumpOnt code completely restructured
Troy Self - 2004-02-05 19:06 - DumpOnt
The code used to be contained in a single dumpont2.java file. It has since been separated into separate packages and classes. The intent is to keep the core functionality separated from the UI. Currently there is an HTMLGenerator class that can still produce the HTML. However, it should be a simple task to create other UIs for it as well.
(0 Comments) [Read More/Comment]
Grand Opening: 01/30/2004
SemWebCentral
Questions? Feedback?
SemWebCentral support
SemWebCentral Statistics
Hosted Projects: 33
Registered Users: 80
Webalizer
Top Project Downloads
(309) Object Viewer
(145) OWL Validator
(100) OWL Mode for Emacs
(93) DumpOnt
(86) CoBrA Demo Toolkit
(82) HyperDAML
(35) OWL Filetype Plugin for VIM
(22) PySesame
(19) Kazuki
(7) ParkaSW
[ More ]
Most Active This Week
( 100% ) SemWebCentral
( 92.85% ) OWL2DIG Library and Tool
( 85.71% ) wsdl2owl-s
( 78.57% ) owl-s2uddi
( 71.42% ) OWL-S API
( 64.28% ) java2owl-s
( 57.14% ) OWL-S Eclipse based Editor
( 50% ) OWL-S UDDI Matchmaker
Done

# Summary

- OSS collaboration software services are great tools!
- Use of Web-based systems to manage software works well
  - Scales from small to large projects
  - Facilitates an agile process
  - Centralizes all project information (code, communications, coordination)
- Many Government agencies using GForge internally and externally (for OSS)
- Agile languages can help ANY development process (from simple to complex)

# The Business Case for Agile Languages

Stephen Ferg
Bureau of Labor Statistics
ferg_s@bls.gov
www.ferg.org

# Agile programming languages

"Scripting" languages
aren't just for scripting any more.

We really should stop
calling them
"scripting" languages.

"Agile" languages
would be more accurate.

# What is an *agile* language?

- Interpreted
  - requires a run-time interpreter or virtual machine
- Untyped or dynamically typed
  - No data declarations
- No compilation step
- *(And interactive in some cases)*

**The hallmark of agile languages...**

*Vastly increased productivity!*

Productivity ::= the number of machine instructions that a programmer can produce per year.

*A programmer can write roughly the same number of lines of code per year regardless of language.*

- Assembly Language

  One line = one machine instruction

- System programming languages

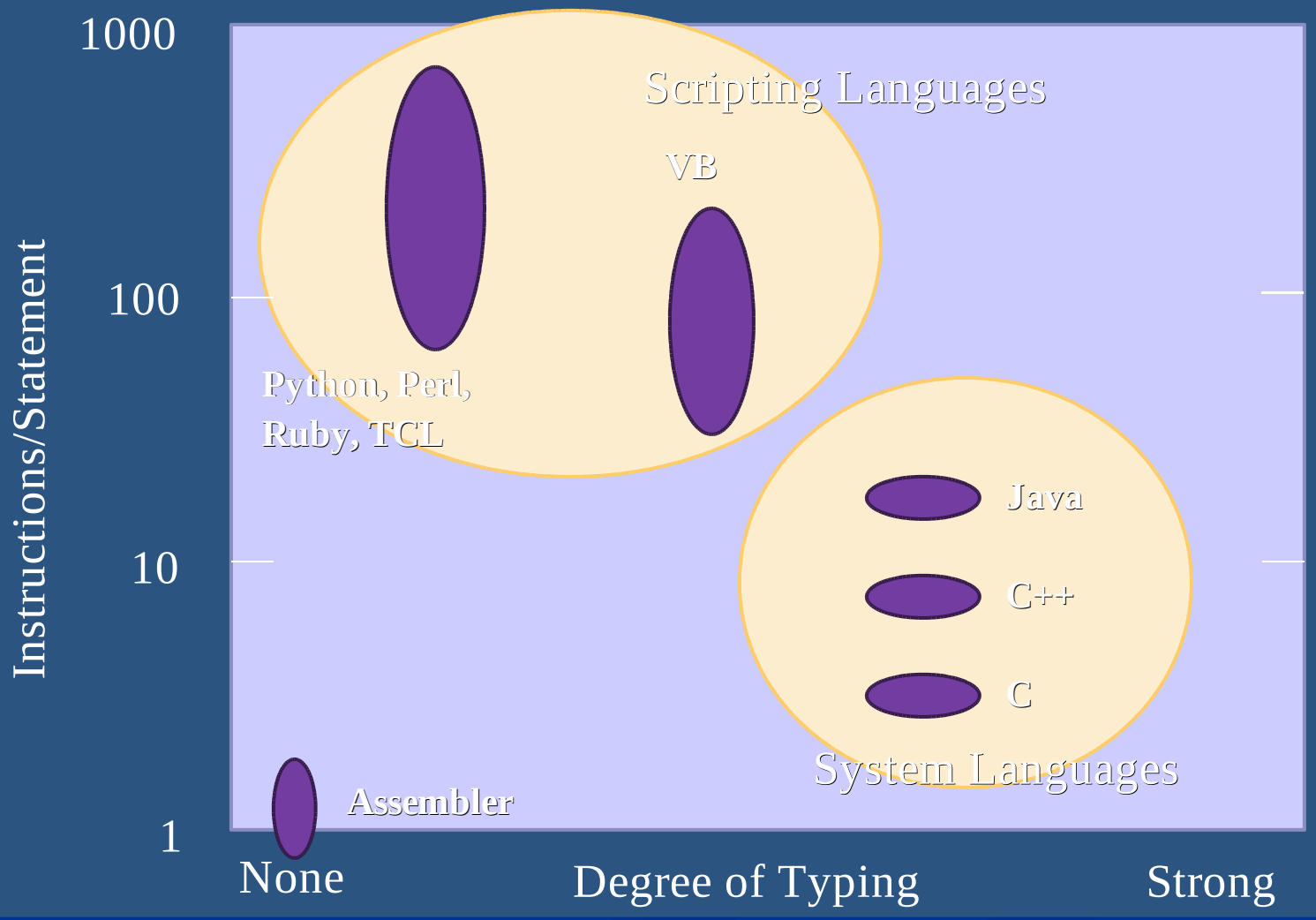
  One line = 3 to 7 machine instructions

- Scripting/agile languages

  One line = 100s to 1000s of instructions

**"Scripting: Higher Level Programmingfor the 21st Century" by John K. Ousterhout**
***IEEE Computer* magazine, March 1998** -- http://home.pacbell.net/ouster/scripting.html

# Language Levels and Productivity




From "Scripting: Higher Level Programming for the 21st Century" by John K. Ousterhout.
This version prepared by Dana Moore and updated by Stephen Ferg

# Developer Reports

- I find that I'm able to program about three times faster in Python than I could in Java, and three times faster in Java than I could in C.
  - *Andy Hertzfeld*
- *When a 20,000 line project went to approximately 3,000 lines overnight, and came out being more flexible and robust ... I realized I was on to something really good.*
  - -- Matthew "Glyph" Lefkowitz
- *...the lines of Python code were 10% of the equivalent C++ code.*
  - -- Greg Stein, eShop

# Fast to write... but maintainable?

... the real punchline of the story is this:

weeks and months after writing my Python program, I could still read the code and grok what it was doing without serious mental effort.

- Eric S. Raymond, author of *The Cathedral and the Bazaar*

# "Programming is fun again!"

- Over and over on *comp.lang.python*:

"Now that I've discovered Python,
I enjoy programming again!"

# Common Concerns Answered

- Who uses Free and Open Source Software (FOSS)?
- What about vendor longevity?
- *Anybody* can change (and maybe mess up) an open-source product.
- Is FOSS as good as commercial software?

# DoD

In 2002, a Mitre study found 115 FOSS (free and open-source) products in use in the U.S. Dept. of Defense.

- http://egovos.org/pdf/dodfoss.pdf

# ... and IBM

- In September 2003, IBM began promoting Linux with a series of television ads depicting a young boy receiving lessons from famous innovators and teachers. The boy represents the next generation of humanity, learning from teachers who – like the open-source community – freely share their accumulated expertise.

- Google, a leading Internet search engine, is powered by Python.
- Yahoo uses Python for its groups site, and in its Inktomi search engine.
- The Philips (formerly IBM) Fishkill semiconductor manufacturing facility uses Linux and Python for factory tool control.

- NASA uses Python in several large projects, including a CAD/CAM system and a graphical workflow modeler used in planning space shuttle missions.

- The National Institutes of Health and Case Western Reserve University are building cutting-edge genetic analysis software with Python.

- The National Weather Service uses Python to prepare weather forecasts.

- Lawrence Livermore National Laboratories is basing a new numerical engineering environment on Python.

- Theoretical Physics Division at Los Alamos National Laboratoı uses Python to control large-scale simulations on massively parallel supercomputers.

*What if Guido*

*got run over by a bus?*

**... or "What about vendor longevity?"**

# Common Misconceptions

- Open-source software has no vendor
- The vendor of an open-source product is the (single) person who developed it.
- *Anybody* can change (and maybe mess up) an open-source product.

# The Python Software Foundation

- A non-profit organization
- Holds Python's intellectual property rights
- Produces the core Python distribution: the Python language, standard libraries, documentation, source code, etc.
- Manages contributions to the Python codebase

The PSF will outlive Guido.

There is no guarantee of any commercial vendor's long-term survival.

*Oh, by the way... you do have software escrows for all of your commercial software don't you?*

# *Is OSS as good as commercial software?*

- Vendors and products vary widely in both the commercial and open-source arena.
- The fact that a piece of software is commercial is no guarantee of its quality.
- The best open-source software is as good as the best commercial software.

# The Bottom Line

- Each product and vendor should be evaluated on its own merits, regardless of whether it is commercial or open-source.

- Some open-source software is in the same league as the best software anywhere, commercial or not.

# The Bottom Line...

*"Use the Best Tool for the Job: Put Both a Scripting and Systems Language in Your Toolbox"*

*- Bill Venners*

http://www.artima.com/commentary/langtool.html

# How is Python Developed

A.M. Kuchling

May 11, 2004

Python will be examined as a detailed example.

Other groups (Linux, Apache, etc.) have similar processes.

# Python Overview

- Agile programming language
  - Designed/Implemented by Guido van Rossum
  - First implemented around 1991
- Applications: scripts, numeric programming, Web tasks, GUI applicaions, teaching...
- Copyright held by non-profit PSF
- License allows commercial use, closing the source.
  - Some commercial software embeds Python
- See www.python.org for more info.

# Good software engineering practices

- Use a version control system.
- Write specifications before code.
- Separate systems into independent modules.
- Review code for correctness.
- Provide new developers with mentoring.
- Use tools to track the code.

# The code: Structure of core Python

- Language interpreter
  - ~90,000 lines of C
- C extension modules
  - regular expressions, POSIX interfaces, math functions, Unicode data, data compression, date/time types, Tk support
  - ~132,000 lines of C
- Python modules
  - XML parsing, Internet protocols, file reading
  - building packages, development environment
  - utilities portability...
  - ~200,000 lines of Python

# The code: Version Control

- Why version control?
  - Makes it easy to roll back changes
    - revert to a previous version
    - determine the history of a file
    - determine what you've changed.
  - Helps resolve conflicts when multiple people are editing the same code.
  - Provides access control
    - Anyone can see the current code
    - Making changes is restricted to the developers
- Version control needs network capability.
  - Python uses CVS. Perl uses Perforce, ...Subversion...

# The code: Change Notification

- Changes are sent to python-checkins mailing list
  - ~100 readers
  - Usually around 10 e-mails per day.
  - Ocassionally hundreds of e-mails (major reorgs)
- (example deleted)
- Purpose:
  - Provides second level of review.
  - Keeps developers aware of which sections of code are changing
- Python reviews after commit. Mozilla reviews before commit.

# The code: Rules for committing changes

- Stability is important; don't leave the tree in a broken state.
- If in doubt, record your patch in the SF patch manager and get it reviewed.
  - ... especially if your've just been granted CVS write access.
- Run the test suite before committing.
- If you're fixing a bug, add a test that would have caught it.

- Usually, if you check out a copy of the CVS trunk, it will compile and run just fine.
- Somp projects have their CVS tree broken or difficult to use for long stretches (e.g. GNOME).
- Source distribution includes 277 test scripts.

# The code: Tracking bugs and patches

| | | | | |
|---|---|---|---|---|
| 946373 | do not add directory of sys.argv[0] into sys.path | 2004-05-02 08:51 | nobody | wrobell |
| 946207 | Non-blocking Socket Server | 2004-05-01 22:45 | nobody | jesperjurcenoks |
| 946153 | askstring => grab fail | 2004-05-01 19:59 | loewis | wweston |
| 945642 | nonblocking i/o with ssl socket not working at all | 2004-04-30 18:14 | nobody | tinolange |
| 944928 | Bugfix for dircheck() in Objects/fileobject.c | 2004-04-29 18:20 | nobody | uberning |
| 944110 | urllib2 authentication mishandles empty pass bugfix 944082 | 2004-04-28 20:52 | loewis | yangir |
| 943953 | Add maketrans to string object | 2004-04-28 15:43 | nobody | siva1311 |
| 943898 | A simple 3-4% speed-up for PCs | 2004-04-28 14:33 | nobody | arigo |
| 943206 | Convert glob.glob to generator-based DFS | 2004-04-27 14:25 | nobody | cben |
| 941881 | PEP309 Partial implementation | 2004-04-25 14:05 | nobody | perky |
| 941486 | Fixes for bug 940578 (glob.glob on broken symlinks) | 2004-04-24 16:23 | nobody | cben |
| 941229 | Allow any encodings other than latin-1 in interactive interp | 2004-04-24 05:49 | loewis | perky |
| 941071 | Replace if/elif chain with dispatch pattern in sre_compile | 2004-04-23 20:07 | niemeyer | rhettinger |
| 940026 | Explain 'in' keyword as it is first used | 2004-04-22 09:35 | loewis | gerrit |

# Planning: The python-dev list

- python-dev is the mailing list where the developers of the Python core congregate.
- Currently has ~600 subscribers, but most of them are lurkers.
  - ~10 people perform the bulk of the work.
  - ~40 people contribute intermittent assistance.
  - ~60-100 offer opinions.
- Lists are primary; there are dozens/hundreds of them.
- IM not used much in Python, though the PSF directors use it for meetings. Often used for assisting users, or for chats between a few individuals. Large meetings are difficult (time zones, keeping the meeting on track.)
- Sprints are face-to-face meetings, used at conferences, and often have agendas. Developers work on a focused set of tasks.

# Planning:Day-today and long-term

- van Rossum is the Benevolent Dictator for Life
  - In theory, has final say on all design decision, and some on whether code is included in the core.
  - In practice, he'll defer the decision to someone else responsible for a given area.
- There's an informal voting process inspired by Apache
  - +1 indicates that the poster is in favor of the suggestion
  - -1 indicates they're against it.
  - +0 indicates 'I don't care, but go ahead'
  - -1 means, 'I don't care, so why bother?'

# Special Interest Groups (SIGs)

- *Focused development projects*
- Current and past SIGs:
  - I18N SIG
    - Produced Unicode sting type, localization
  - Matrix SIG
    - Produced numeric array data type, minor core language changes
  - Database SIG
    - Standard API for RDBMSes
  - Web SIG: ongoing...

# Python Enhancement Proposals (PEPs)

- PEPs are documents describing proposed changes:
  - Documentation
  - Design rationale
  - Alternative designs, and why they weren't used
- Modeled on the IETF RFCs.
- Available from www.python.org/peps/
- Requiring a PEP imposes some rigidity and contemplation

# Good software engineering practices

- Use a version control system.
  - Python uses CVS
- Write specification before code.
  - PEPs, python-dev discussion
- Separate systems into independent modules.
  - core/extension module/library separation
- Provide new developers with mentoring.
  - python-dev discussion, PEPs describing procedures
- Use tools to help manage development
  - mailing list manager, CVS, bug/patch tracking

# Concluding points

- Open source follows good software engineering practices
- Open source development is managed
  - in some ways loosely
    - distributed teams
    - changeable schedules
    - weak ownership of code
  - in other ways tightly
    - good communication
    - automatic tracking tools
    - code reviews
    - PEPs

# Concluding points

- Loosely:
  - you can't make people do things.
- Tightly:
  - the distributed nature requires tools for good communications
- Software Engineering Practices:
  - this is really in self-defense
  - projects that don't follow them don't survive very long

# Example: Scientific Computing

- Paul Barrett, Space Telescope Science Institute
  - Python is what BASIC should be!
    - Scientists are not programmers, nor do they want to be!
  - An agile language
    - span from a scripting language to an OO language
    - increased productivity
  - An Extensible Language
    - C/C++: SWIG, Weave, Pyrex, blitz, pysco
    - FORTRAN: pyfort
  - A Numerical language (imaginary numbers, arrays)
  - Python provides an interactive interface to C++ code
    - Improves testing and debugging.

# Example: Teaching Programming

- Jeff Elkner, Yorktown High School
  - Computer programming for everyone (CP4E)
    - Improve computer programming literacy rate
    - Open source community is a great education amplifier
  - Using Python as introductory programming language
    - Teaching a semester's worth of material in four weeks
    - Student projects are much more complex (and interesting) than would be possible in Java or C++
  - Guido van Robot
    - A tool for teaching basics of programming
    - Developed as a collaborative project by advanced students

# John Miller's Thesis

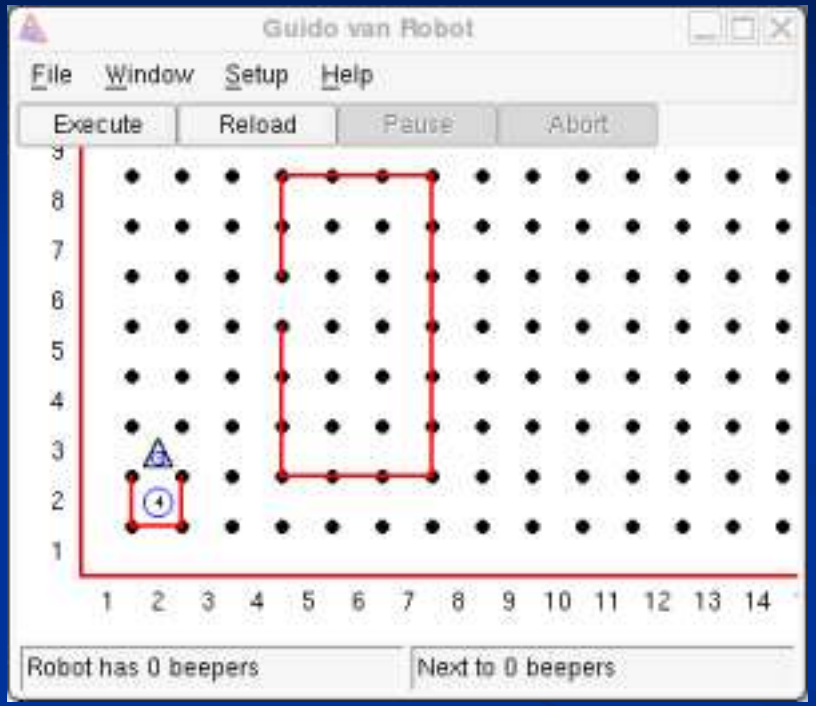
- Studied three years of Python edu-sig e-mails
  - The pseudo-code like syntax of Python facilitates teaching
  - Learning to Program improves the ability to learn
    - Develops logical problem solving skills
    - Programming becomes a tool for learning other disciplines
    - Even helps improve writing skills
      - Reinforces the structure of documents (grammar, punctuation)
      - Forces students to be accurate in spelling of variables

# GvR in the Classroom

Program Window (step12.gvr)

File

```
1   define go_to_trash_can:
2       move
3       turnleft
4       move
5
6   define pick_up_trash:
7       while next_to_a_beeper:
8           pickbeeper
9
10  define go_to_dumpster:
11      turnleft
12      turnleft
13      do 3:
14          move
15      turnleft
16      do 3:
17          move
18      turnleft
19      while front_is_clear:
20          move
21
22  define throw_away_trash:
23      while any_beepers_in_beeper_bag:
24          putbeeper
25
26  go_to_trash_can
27  pick_up_trash
28  go_to_dumpster
29  throw_away_trash
30  turnleft
31  turnleft
32  move
33  turnoff
```




# GvR as a Path to Python

- Free intro programming tool
- Pythonic syntax
- Python references in lessons

# Future Goals

- More teachers using GvR (GvR community)
- GUI world builder
- More lessons (including advanced applications)