

# Rapid Peer Education of a Computational NE Skill Suite

ANS Annual Conference, Hollywood, FL

Kathryn D. Huff, Anthony M. Scopatz  
Nicholas D. Preston , Paul P.H. Wilson

University of Wisconsin-Madison

June 30, 2011





# Outline

## ① Motivation

State of Nuclear Engineering  
Requisite Skill Suite  
University Curriculum

## ② Organization

BiWeekly Meetings  
Listhost  
Wiki

## ③ Bootcamps

Audience  
Topics  
Technique  
Feedback



# State of Nuclear Engineering

## Applications

Lattice Physics

Neutronics

Thermal Hydraulics

Fuel Performance

Criticality Safety

Depletion

Numerical Methods

Advanced Monte Carlo

Turbulent Flow

Sensitivity Analysis

Risk Assessment



Figure: Image  
courtesy of Argonne  
National Laboratory.

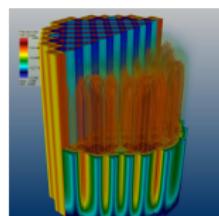


Figure: Image  
courtesy of ORNL.

.

.



# Requisite Skill Suite

## Essential Skills

- Parallelization
- Databases
- Unit Testing
- Version Control
- Fortran/C++/Python
- Statistics
- Advanced Visualization
- Development Workflow
- Documentation
- Debugging
- Unix/Linux



Figure: Image  
courtesy of Argonne  
National Laboratory.

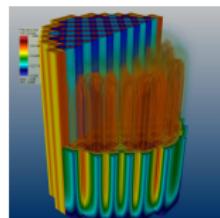


Figure: Image  
courtesy of ORNL.

.

.



## University Curriculum is Limited

CS 367	Introduction to Data Structures
CS 412	Introduction to Numerical Methods
EP/EMA 471	Engineering Problem Solving II
EP/EMA/NE 476	Computational Engineering

**Table:** Undergraduates must take one course from the list above.

**NE476:**

*Course Description: Basic tools of professional scientific computation for UNIX environments are taught. Programming skills in a compiled language are developed through engineering examples. Applications reinforce engineering problem-solving skills first examined in introductory courses, while motivating progressively more advanced computational methods.*



# Department Scope

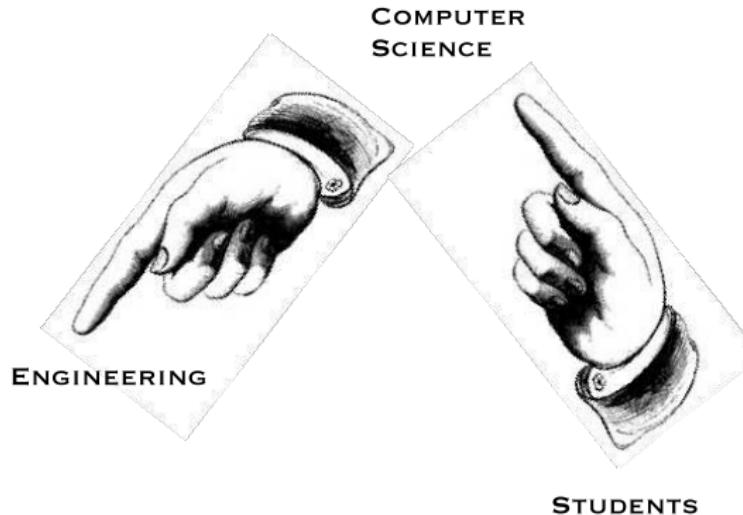
**COMPUTER  
SCIENCE**



**STUDENTS**

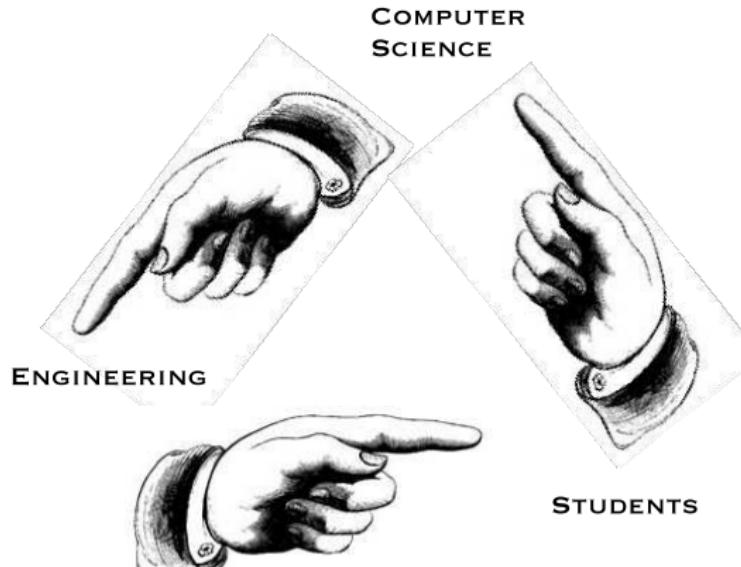


# Department Scope





# Department Scope





# Outline

## ① Motivation

- State of Nuclear Engineering
- Requisite Skill Suite
- University Curriculum

## ② Organization

- BiWeekly Meetings
- Listhost
- Wiki

## ③ Bootcamps

- Audience
- Topics
- Technique
- Feedback





# BiWeekly Meetings

Autotools	Automated Documentation	Awesome Sauce
Backup	Beamer	Bert Computing Cluster
Build Systems	C++0x	C++ And Fortran
Cmake	Condor	Cython
Debugging	In Defense of Fortran	Distributed Version Control
Doxxygen	FileSystem	FuturePlanning
GpGpu	GprofDdd	Large Sciency Documents
LaTeX	LibraryDesign	LightningTalks
MachineLearning	Matplotlib	OpenAccess
Parallelization with MPI	Parallelization with Python	PerlDataLanguage
PlayStation3	PrettyPrinters	Processes
PyCon2010	PythonCinterface	R
Sahana Haiti Relief	SavingScholarship	Scripting
The Shell	ShowAndTell	UnixAccess
Unit Testing	Version Control	Visualization
Visualization Course	Web Programming	XML





# Audience

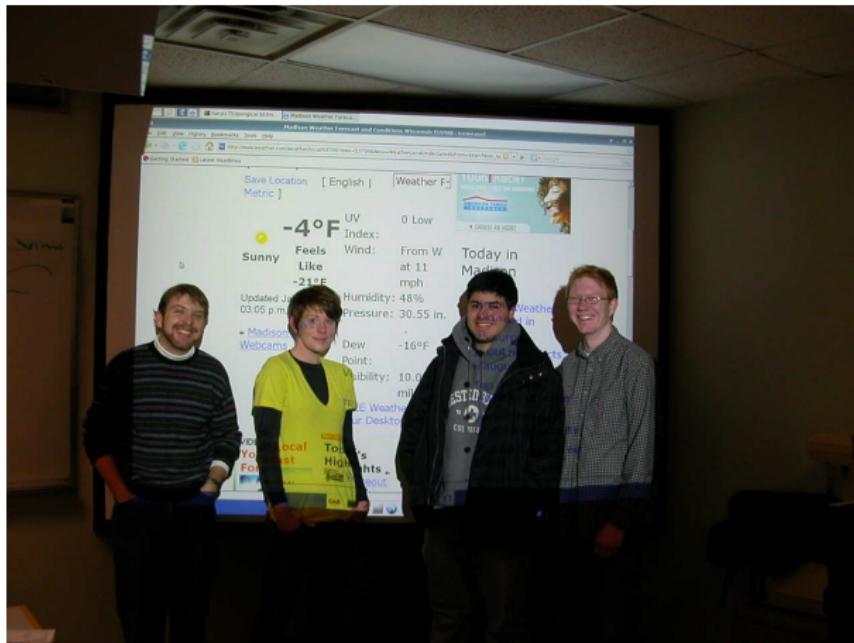


Figure: Hacker Within regulars develop a community.



# Listhost

An interdisciplinary, peer driven listhost of approximately 130 students gives students a quick way to ask questions of an experienced but casual group.



## Hacker Within



### Python question

Options

☆ 10 messages - [Expand all](#) - [Report discussion as spam](#)

Emily B. Sessa Hi, Wolfman sent me to seek advice from this group... I've got a very beginner-level question about Python - I'm a grad student and just need to ru... May 21, 11:42 pm

Matt Terry Wolfman is a wise one : ) I'm going to guess that numpy/scipy are installed correctly, but are installed for a version than the default python. From the co... May 22, 1:32 am

Anthony Scopatz Also, the command 'python -V' will tell you which version of Python you have installed, so you don't have to worry about guessing. On Sun, May ... May 22, 3:23 am

Mr. Puneet Kishor \$ which python will tell you the first python instance in your path. You can have multiple python versions and call them directly as '/usr/local/bin/...' May 22, 9:24 am

esessa Hi all, Thank you so much! This has already helped! Taking Matt's advice, up top, and just typing python2.6, I get this, which is what I was hoping for... call... May 22, 12:43 pm

Anthony Scopatz - Hide quoted text - Show quoted text - On Sun, May 22, 2011 at 11:43 AM, esessa <ese...@wisc.edu> wrote: > Hi all, > Thank you so much! Thi... May 22, 1:01 pm

Wiki



hackerwithin.org



# Outline

## ① Motivation

- State of Nuclear Engineering
- Requisite Skill Suite
- University Curriculum

## ② Organization

- BiWeekly Meetings
- Listhost
- Wiki

## ③ Bootcamps

- Audience
- Topics
- Technique
- Feedback



# Bootcamps

Topic	Year	Days	$\frac{hr}{day}$	Attendees
Unix	2009	4	2	18
C++	2009	4	2	30
Python	2010	3	4	82
Software Carpentry	2011	3	4	79
MSU	2011	2	5	40

**Table:** Previous Bootcamps have been conducted in two condensed formats. Feedback surveys from early bootcamps led to a three rather than four day structure.



# Programming Experience

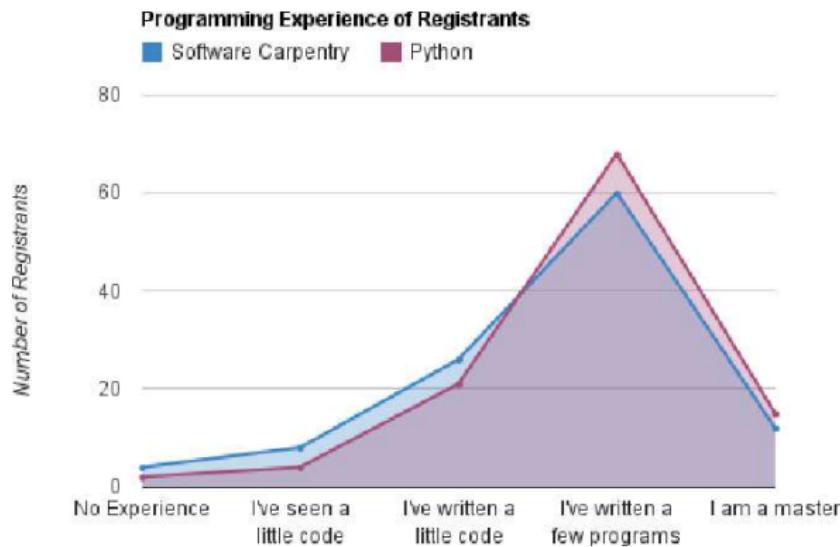


Figure: The participants reported varying levels of experience.



# University Standing

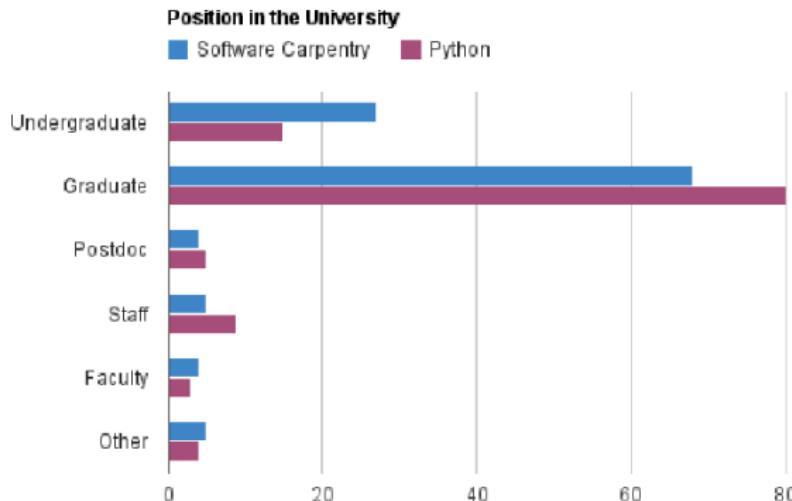


Figure: The participants reported varying levels of education.



# Disciplines

## Departments

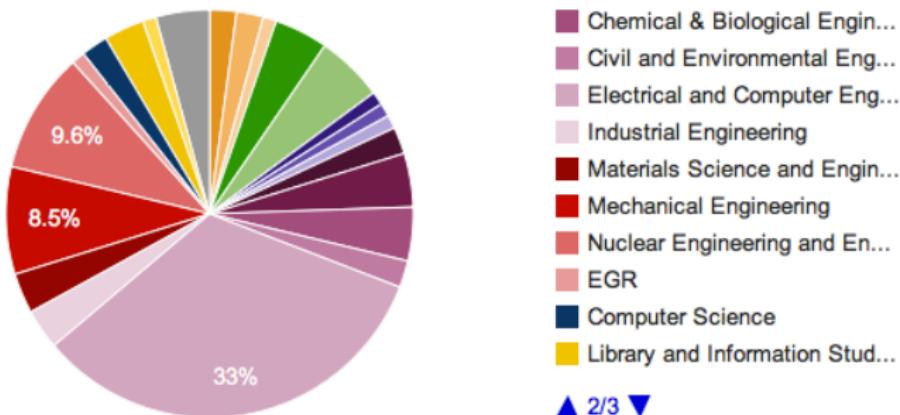


Figure: The participants come from various backgrounds. Here is the distribution from the Software Carpentry bootcamp.



# Bootcamp Topics

Unix	C++	Python	S.C.
Shell	basics	installation	Shell
Environment	conditionals	data types	Version Control
RegEx	loops	modules	Text Editors
Make	functions	numpy/scipy	Databases
Autotools	multi-file progs	matplotlib	Web Programming
	inheritance	django	Auto. Documentation
	pointers	pytables	Build Systems
	data structures	swig/f2py	Unit Testing
	templates	pytables	Debugging
	exceptionss	wxPython GUIs	

**Table:** The bootcamps have covered myriad topics pertinent to a fundamental computational nuclear engineering skill suite.



# Virtual Machine

Before the bootcamp, students are asked to:

- Install VirtualBox
- Download a 1GB file

When the bootcamp begins, the students open their virtual machines and may immediately perform exercises and try out tools on identical ubuntu computers pre-loaded with an array of appropriate software.



# Virtual Machine

For the software carpentry bootcamps the virtual machines were loaded with an array of awesome (free) software.

- python 2.6
- ipython
- scipy, numpy, matplotlib, pygame, pytables
- emacs
- vim
- nano, pico, nedit
- g++
- gfortran
- cmake
- doxygen
- mercurial, git, bzr, svn
- diff, colc, diffutils



# Example Driven

Source code is provided by a shared repository. Each student checks out their own version and makes local changes in sync with the class.

## C++ example with CMake

CMake is a C/C++ build system (with some additional support for Fortran, Java, and Python extensions). The website description:

Welcome to CMake, the cross-platform, open-source build system. CMake is a family of tools designed to build, test and package software. CMake is used to control the software compilation process using simple platform and compiler independent configuration files. CMake generates native makefiles and workspaces that can be used in the compiler environment of your choice.

## Get the example code

Our first order of business is to get you access to these notes and other course materials. We've placed them in a repository on our [Google Code site](#). However, instead of having you visit the site and download them manually, we will download the latest version with version control.

You may check out a read only or a committing copy of the repository. To check out a read only copy of today's session::

```
mkdir build_systems
cd build_systems
svn co http://hacker-within.googlecode.com/svn/trunk/sc/documentation/src cxx-example
```

## Build the example

CMake is designed to keep files that result from the build process separate from the source code directory. This makes it easier to keep the source directory under version control, and it makes it easy to start a fresh configuration and build with a simple `rm -rf` on the build directory.

Let's make a build directory

```
mkdir cxx-example_build
cd cxx-example_build
```

CMake has a Turing complete scripting language with scripting directives designed specifically for a C/C++ build system. These



# Peer Taught

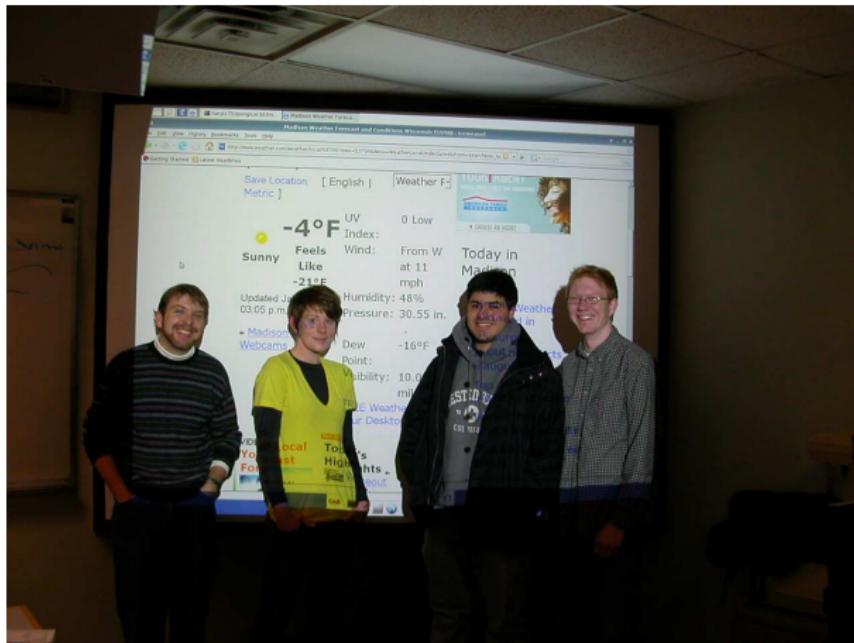
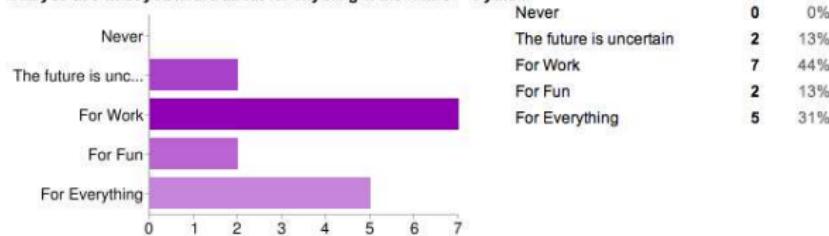


Figure: Hacker Within regulars learn by teaching and are not terribly intimidating.

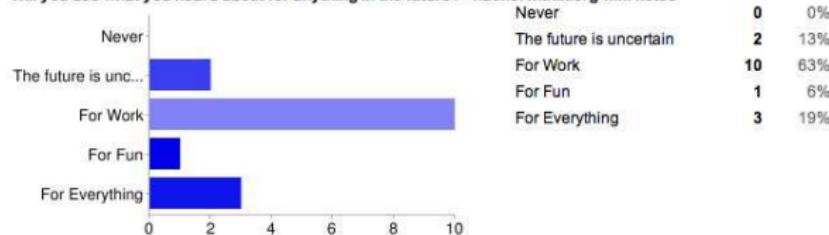


# Feedback : Python Bootcamp I

Will you use what you heard about for anything in the future? - Python



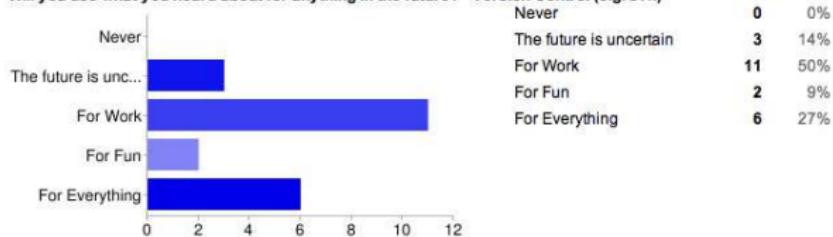
Will you use what you heard about for anything in the future? - hackerwithin.org wiki notes





# Feedback : Software Carpentry Bootcamp

Will you use what you heard about for anything in the future? - Version Control (e.g. svn)



Will you use what you heard about for anything in the future? - Build Systems (e.g. CMake)



# Lessons



- Scientific computation education is necessary



# Lessons

- Scientific computation education is necessary
- and best practices are absent from curriculum.

# Lessons



- Scientific computation education is necessary
- and best practices are absent from curriculum.
- Students appreciate a swift introduction to software tools

# Lessons



- Scientific computation education is necessary
- and best practices are absent from curriculum.
- Students appreciate a swift introduction to software tools
- particularly one that is hands on



# Lessons

- Scientific computation education is necessary
- and best practices are absent from curriculum.
- Students appreciate a swift introduction to software tools
- particularly one that is hands on
- and example driven.



# Lessons

- Scientific computation education is necessary
- and best practices are absent from curriculum.
- Students appreciate a swift introduction to software tools
- particularly one that is hands on
- and example driven.
- A community is helpful



# Lessons

- Scientific computation education is necessary
- and best practices are absent from curriculum.
- Students appreciate a swift introduction to software tools
- particularly one that is hands on
- and example driven.
- A community is helpful
- and can drive itself.



# References I

- [1] the.hacker.within.  
<http://hackerwithin.org/>, 2010.
- [2] J. E Hannay, C. MacLeod, J. Singer, H. P Langtangen, D. Pfahl, and G. Wilson.  
How do scientists develop and use scientific software?  
In *Proceedings of the 2009 ICSE Workshop on Software Engineering for Computational Science and Engineering*, pages 1–8, 2009.
- [3] Zeeya Merali.  
Computational science: ...Error.  
*Nature*, 467(7317):775–777, 2010.
- [4] G. V Wilson.  
Where's the real bottleneck in scientific computing?  
*American Scientist*, 2010.



## References II

[5] Greg Wilson.

Software carpentry: Essential software skills for research scientists.  
2006.