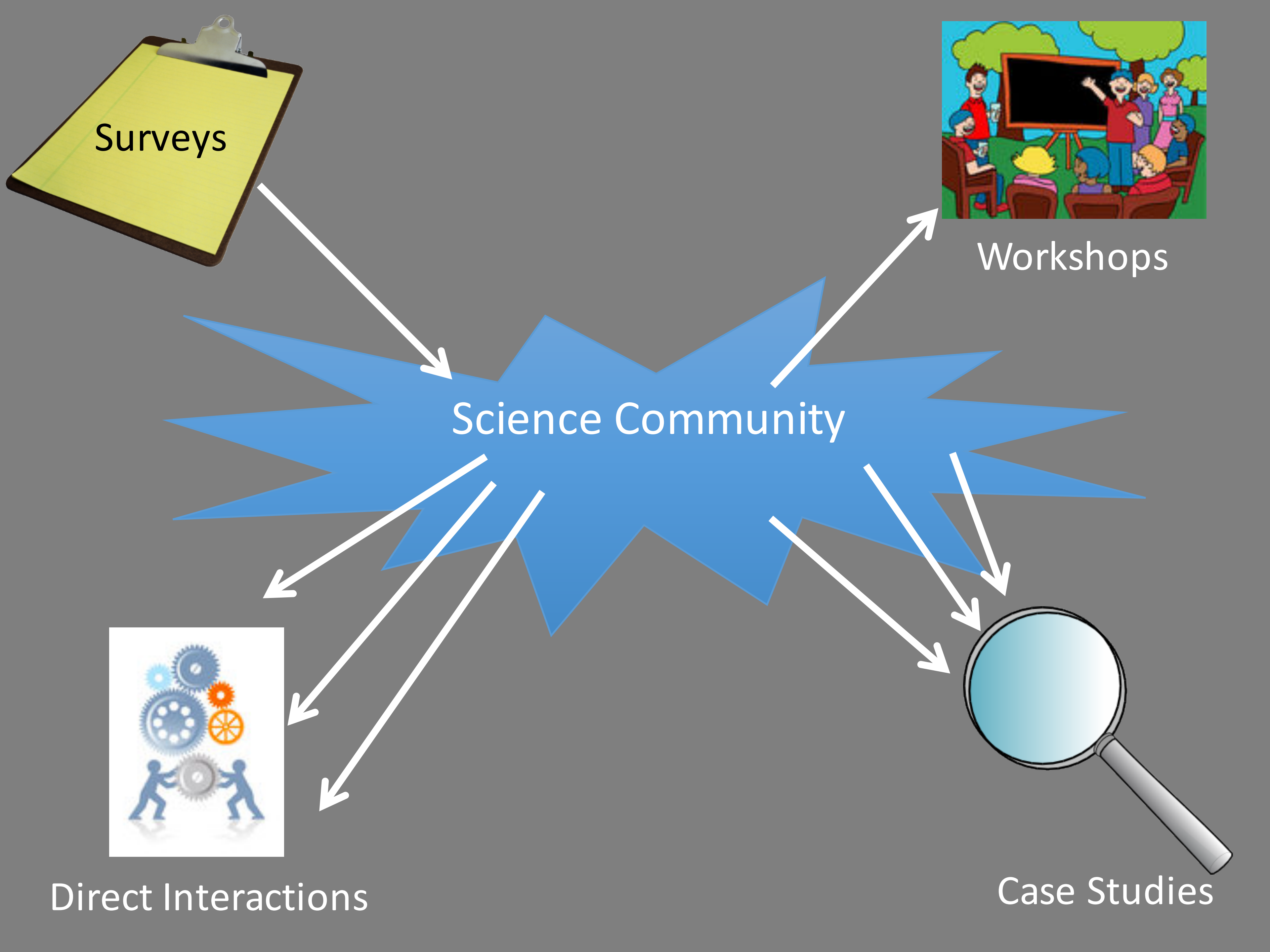
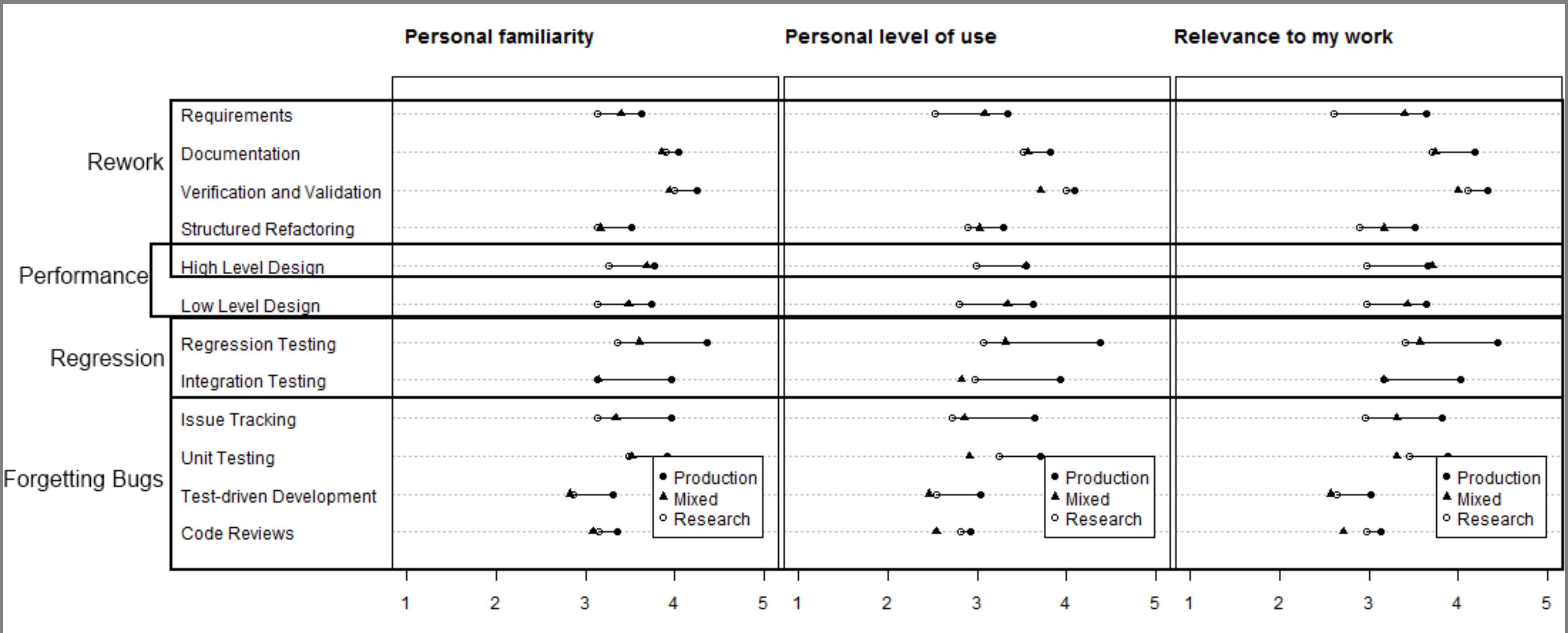
What We Have Learned About Using Software Engineering Practices in Scientific Software

Jeffrey Carver
University of Alabama
http://carver.cs.ua.edu/Projects_CSE.htm


Surveys
Workshops
Science Community
Direct Interactions
Case Studies

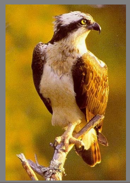
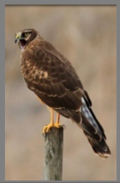
Community Surveys



Case Studies





	FALCON	HAWK	CONDOR	EAGLE	NENE	OSPREY	HARRIER
Application Domain	Prediction of Product Performance	Predication of Manufacturing Process	Analysis of Product Performance	Signal Processing	Calculate Molecule Properties	Weather Forecasting	Engineering Mesh Generation
Duration (Years)	~ 10	~ 6	~ 20	~ 3	~ 25	~10	~8
# of Releases	9 (production)	1	7	1	?	?	~16
Staffing (FTEs)	15	3	3-5	3	~10 (100's of contributors)	~10	5 primary + students
Customers	< 50	10s	100s	None	~ 100,000	100s	10s
Code Size (LOC)	~ 405,000	~ 134,000	~200,000	< 100,000	750,000	150,000	50,000
Primary Languages	F77 (24%), C (12%)	C++ (67%), C (18%)	F77 (85%)	C++, Matlab	F77 (95%)	Fortran	C++ (50%), Python (50%)
Other Languages	F90, Python, Perl, ksh/csh/sh	Python, F90	F90, C, Slang	Java Libraries	C	C	None
Target Hardware	Parallel Supercomputer	Parallel Supercomputer	PCs to Parallel Supercomputer	Embedded Hardware	PCs to Parallel Supercomputer	Parallel Supercomputer	Linux, Windows

Case Studies:
Lessons Learned

- Verification and Validation are difficult
- Performance competes with other goals
- Use of higher-level languages is low
- Developers prefer command line over IDE
- Agile development methods are useful
- Primary language does not change
- External software is risky
- Multi-disciplinary teams are important
- Success/failure depends keeping customers/sponsors satisfied

SE4Science Workshops

SE4Science Workshop Series
http://SE4Science.org

- Facilitate interaction between SE and Scientists

- Held at ICSE, ICCS, and SC

- Discussion Topics
  - Research Software vs. IT Software
  - CSE software quality goals
  - Crossing the communication chasm
  - Measuring impact on scientific productivity
  - Reproducibility of results

SE4Science Workshops
Domain Differences

- Complex domains
- Main focus on science
- Long lifecycles
- Investigation of unknown introduces risk
- Unique characteristics of developers
  - Deep knowledge of domain – lack formal SE
  - Often the main users of the software

SE4Science Workshops
Quality Goals

- Lack of viable V&V techniques
- Focus on process transparency
- Guaranteed not to give an incorrect output
- Other SE characteristics not as important
  - Testability, reusability, maintainability

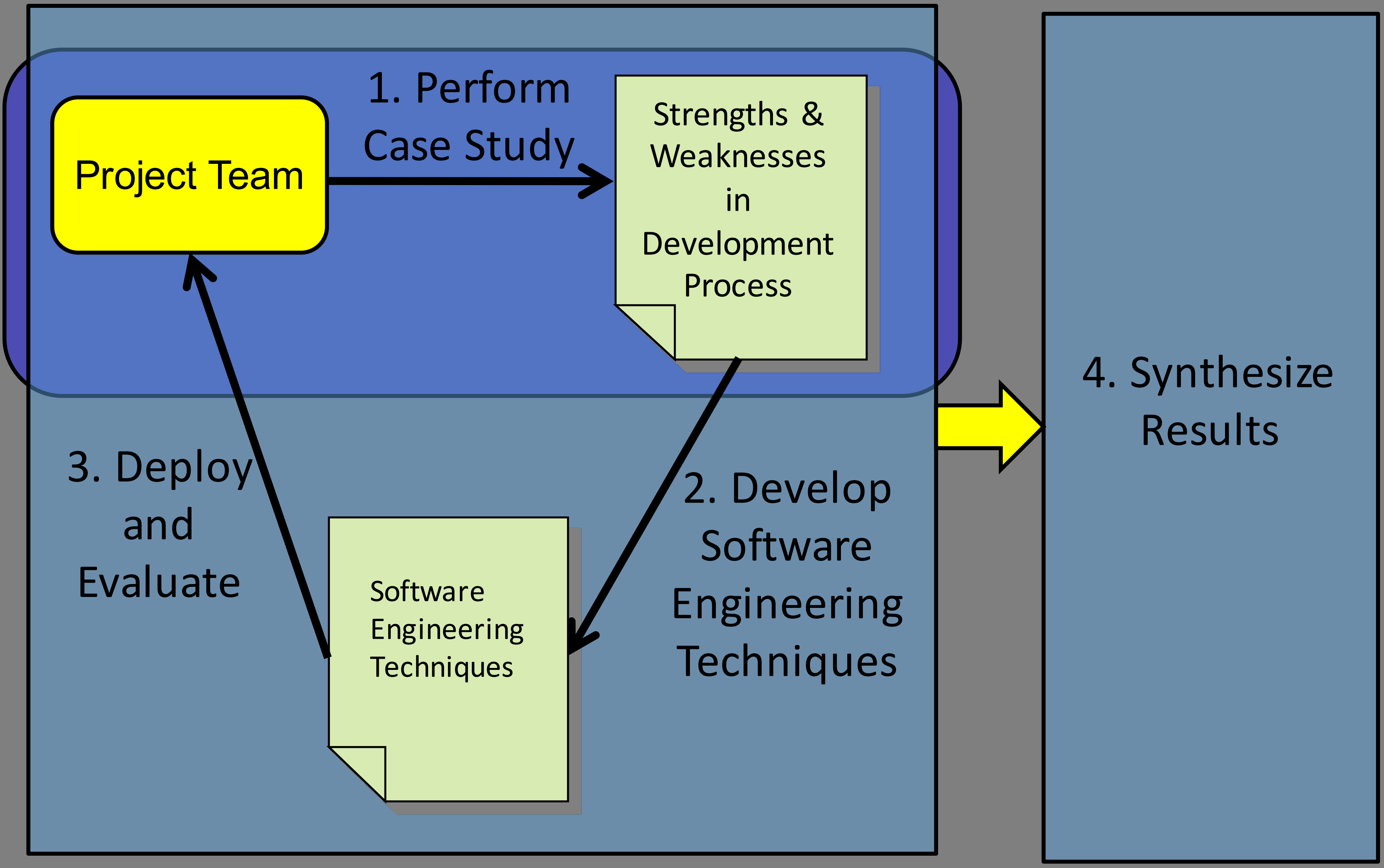
SE4Science Workshops
Communication

- Need to eliminate the stigma associated with SE
- Software Engineers need to
  - Understand domain constraints
  - Understand specific problems
  - Learn from Computational developers
  - Describe SE concepts in terms familiar to Computational developers
- Need people with expertise in both SE & Science
- Science teams need:
  - To realize a problem before needing help
  - Real examples of SE success within their domain

SE4Science Workshops
Scientific Productivity

- Need to evaluate impact
- Scientific productivity ≠ Software productivity
- Need results in a relatively short time
  - Self-assessments
  - Word of mouth

Methodology for Direct Interactions



For More Papers/Information

- Please see my website for papers related to this work:

- http://carver.cs.ua.edu/Projects_CSE.htm

What We Have Learned About Using Software Engineering Practices in Scientific Software

Jeffrey Carver
University of Alabama
carver@cs.ua.edu

THE UNIVERSITY OF ALABAMA

1243887, 1445344