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
# Case Studies

<table>
<thead>
<tr>
<th></th>
<th>FALCON</th>
<th>HAWK</th>
<th>CONDOR</th>
<th>EAGLE</th>
<th>NENE</th>
<th>OSPREY</th>
<th>HARRIER</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Duration (Years)</strong></td>
<td>~10</td>
<td>~6</td>
<td>~20</td>
<td>~3</td>
<td>~25</td>
<td>~10</td>
<td>~8</td>
</tr>
<tr>
<td><strong># of Releases</strong></td>
<td>9 (production)</td>
<td>1</td>
<td>7</td>
<td>1</td>
<td>?</td>
<td>?</td>
<td>~16</td>
</tr>
<tr>
<td><strong>Staffing (FTEs)</strong></td>
<td>15</td>
<td>3</td>
<td>3-5</td>
<td>3</td>
<td>~10 (100’s of contributors)</td>
<td>~10</td>
<td>5 primary + students</td>
</tr>
<tr>
<td><strong>Customers</strong></td>
<td>&lt; 50</td>
<td>10s</td>
<td>100s</td>
<td>None</td>
<td>~100,000</td>
<td>100s</td>
<td>10s</td>
</tr>
<tr>
<td><strong>Code Size (LOC)</strong></td>
<td>~405,000</td>
<td>~134,000</td>
<td>~200,000</td>
<td>&lt; 100,000</td>
<td>750,000</td>
<td>150,000</td>
<td>50,000</td>
</tr>
<tr>
<td><strong>Primary Languages</strong></td>
<td>F77 (24%), C (12%)</td>
<td>C++ (67%), C (18%)</td>
<td>F77 (85%)</td>
<td>C++, Matlab</td>
<td>F77 (95%)</td>
<td>Fortran</td>
<td>C++ (50%), Python (50%)</td>
</tr>
<tr>
<td><strong>Other Languages</strong></td>
<td>F90, Python, Perl, ksh/csh/sh</td>
<td>Python, F90</td>
<td>F90, C, Slang</td>
<td>Java Libraries</td>
<td>C</td>
<td>C</td>
<td>None</td>
</tr>
<tr>
<td><strong>Target Hardware</strong></td>
<td>Parallel Supercomputer</td>
<td>Parallel Supercomputer</td>
<td>PCs to Parallel Supercomputer</td>
<td>Embedded Hardware</td>
<td>PCs to Parallel Supercomputer</td>
<td>Parallel Supercomputer</td>
<td>Linux, Windows</td>
</tr>
</tbody>
</table>
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

1. Perform Case Study

2. Develop Software Engineering Techniques

3. Deploy and Evaluate

4. Synthesize Results

Strengths & Weaknesses in Development Process

Project Team
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