

Open Questions in Software Engineering for HPC 16 November 2009



Tom Epperly <epperly2@llnl.gov>

Center for Applied Scientific Computing

Option: Auspice statement or other directorate information

Lawrence Livermore National Laboratory

- Equation-oriented language: ASCEND III
- Global optimization (numerical algorithms)
- Optimization for reactor design under uncertainty
- Real time optimization for chemical plants (SEI CMM level 3)
- CCA/Babel - high-performance language interoperability
 - Interoperability between next gen languages
- Tokamak reactor modeling
- Build & configure
- HPC \approx Scientific Computing

If you build it, they will come. Field of Dreams



- Well... it's what would we wish were true
- HPC is largely resistant to software engineering innovations
- Very few technologies are successful
 - Source code repositories (cvs, svn, git, ...)
 - Testing
 - Performance tools
 - Components & 3rd party libraries
 - Python
- Hardware drives changes in software engineering
- New software engineering is only widely adopted when the pain is great



The opportunities in HPC software engineering



- Education targeting physical scientists & engineers
- Improvements in build & configure tools
- New static analysis based on tools like ROSE
 - correctness checking
 - style & complexity analysis to target development
 - higher level semantics checking
- Lowering the barrier to entry for component tools (e.g., Swig, Bocca, OnRamp, etc.)
- Building bridges from legacy to new approaches
- Folding component ideas into normal programming languages

