BERKELEY LAB
LAWRENCE BERKELEY NATIONAL LABORATORY

U.S. DEPARTMENT OF ENERGY

Best Practices, Challenges and Recommendations

Summarized

Anshu Dubey

The Papers

- Represent two different perspectives
  - Software engineers working with or studying software development in applications domains
  - The experiences and insights of developers of scientific codes / libraries
- The dominant common themes are
  - open development and community support
  - Tighter interactions among domain scientists and code developers, and developers and their users (which are often domain scientists)
  - Users prefer robust and simple to use codes and platforms
  - Continuous integration is good
  - Sustainability is challenging for many reasons: changing science, changing platforms and technology, and funding are some of the important ones

The Main Points

- Take from the developers side :
  - Rethinking code architecture in terms abstraction that can plug into the technologies under development to combat disruptive hardware changes
  - Migration of composition and extensibility to the runtime

- Findings from studies and workshop
  - The reasons for failure are many, but what succeeds looks broadly similar
  - Takes visionary leadership and a tightly knit central management
  - Tight coupling between scientists, developers and software engineers generally beneficial
  - Adoption of some software process is better than none

General Recommendations

- The Editorial on rules for producing good software lists
  - use existing code/tools where possible
  - code well, be simple and transparent,
  - use your code, nurture your community, promote, find support
  - be satisfied with less than perfection
  - keep the scientific goal in always in your focus

General Recommendations

- Recommendations from Carver et.al emphasize
  - Unit testing and test driven development
  - High level requirements specification and metrics
  - Documentation and code review
  - Abstractions and continuous integration
- Tools to do some of these
  - Code repositories, wikis
  - Issues tracking and build systems
  - Project management, task trackers

Software Sustainability Institute

- Should be vested with authority to guide and oversee functions such as
  - Development and oversight of standardized quality control, testing, regression, and documentation processes
  - A central, common repository
  - A central resource for services and consulting
  - A “think tank,” of sorts, where the issues and requirements for sustainability can be debated
  - A software orphanage/retirement center