Third International Workshop on Software Engineering for High Performance Computing Applications

International Conference on Software Engineering

May 26, 2007

Software Engineering Group: Context

- At least two different foci
  - Computational Science
    - Focused on science or knowledge
    - Typically large machines
  - ~~Business~~
    - Focused on making money/customers
    - Typically smaller machines

Software Engineering Group

- What are the top things that the software engineering community has to offer the HPC community?
  - Communicate SE successes to larger Computational Science community
    - Find HPC groups with good practices (e.g. version control, regression testing, inspections, …) have mostly learned them the hard way (passed down from elders)
    - Only get it if there were on a good project
  - Elementary Individual Practices that could be use with little effort (e.g. version control, unit testing, regression testing, …)
    - Removes a boundary to HPC use
  - A view of software that takes other attributes into account rather than just current stuff
    - Maintenance
    - Portability
  - Component-based software engineering (compilers, libraries)
  - Knowledge of middleware – for example, simplifying the access to grids

Software Engineering Group

- What are some problems/frustrations you have had in trying to work with the HPC community or the research domain?
  - Clear difference in priorities, the software is merely a tool (like a calculator), the focus is on the paper. The software is thrown away and often not valued
  - Software is not always viewed as an "asset" in the same way that it is in the IT industry
  - SE community and HPC communities have different views of the "problems" that need solving – e.g. Quality Assurance techniques
  - The view that SE's are "just imposing more process on us", rather than "letting us write our algorithms"
  - Often projects do well without SE, because of size, smart people, … so they don't see the need for SE
  - Ivory tower – "you don't know about our problems so why should we listen to you"
  - SE's don't always have access to study projects that are not successful

Software Engineering Group

- What are things we'd like to offer, but don't have yet?
  - A software lifecycle and tool set for "research" projects
    - SE often lack a deep understanding of how to actually write the stuff
  - We need more evidence to show:
    - What the real problems are
    - What techniques will work in specific domains
  - Process where domain experts can develop a version of the code that is then optimized by a software engineer (a parallelization expert)
    - "on-demand" parallelization (semi-automatic)