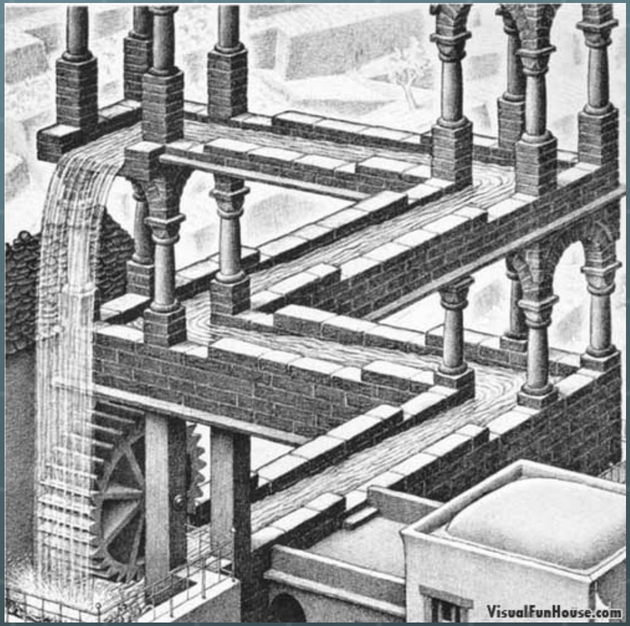
Workshop on Sustainable Software for Science: Practice and Experiences



Development Experiences
Mark C. Miller, LLNL (FASTMath, VisIt, Silo)

LLNL-PRES-646558 This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

13 Articles about development experiences

- MOOSE
- R/qtl
- FASTMath*
- Chombo/FLASH*
- MINRES-QLP
- KHMER
- Future Architectures*
- VisIt
- VisTrails
- MVAPICH
- OWL/Bean*
- BaTLab
- DUNE

*more about practices than specific products

Longevity:4-13 years, #Developers:4->100?

Some Common Themes

- Distributed Development
- Testing
- Documentation (wikis)
- Subversion (5) / Git (7)
- Funding
- Regular Releases
- Multiple roles of contributors

- Multiple stakeholding institutions
- Core group has responsibility to maintain
- Responsive to users
- Eat your own dogfood

Less Common

- Performance / Scalability
- GPU and other exotic hardware
- Agile
- Configure/Build
- Windows
- Auto-gen documentation
- Examples apart from tests
- Gui testing

Unique Perspectives

- Managing API Changes by Embedding Applications
- Same Software Engineering Practices for Data
- Dealing with “diffs”
- Automating citation of software contributions
- Remember VHS vs. Beta

Managing API Changes

- Backwards compatibility has costs
  - Can "clutter" code and unnecessarily burden
  - Costs developer time (maintaining old and new)
  - Slows user migration to new APIs
- Works best when…
  - Applications are few in number
  - Applications are "close to" changing APIs
    - MOOSE: Applications are in same repository
    - Chombo: Application developers work closely

Same Software Engineering Practices are needed for Data

- Vocabularies (metadata) and provenance of test simulation results
- Often architected/designed for specific evaluative purpose
- Need revision control
- Can have “bugs” (that need to be fixed)
- Evolves and grows just as code respositories do
- Require documentation

Dealing with “diffs”

- Variations in computed outputs due to variations in...
  - code
  - compilers
  - operating systems
  - third-party libraries
- Detecting “diffs” is easier than quantifying them as “pass” or “requires a closer look” or “fail”
  - Often need domain-specific metrics

Automating citation of software contributions

- Not described in any of the submissions I read but…
- PETSc is a sort of meta-solver library
    - hypre, ML, NOX, BoomerAMG, Parasails, SuperLU, MUMPS, ESSL, UMFPACK, LUSOL, PaStiX. MATLAB
- When an app performs a solve, PETSc keeps track of packages used to produce the result and can print out TeX bibliographic references for each

Recall VHS vs. Beta

- Higher quality doesn’t always mean greater success
- No argument beta was higher quality picture than VHS
- Why did VHS “win”
  - More players available and at lower cost
  - More titles available = attracted more consumers
  - Costs for titles and players came down

What does "sustainable" mean?

- Has a long and successful life?
- A lot of people have a stakehold
- Funding continues to flow in
- Too big to fail?
- Is DOS a good example of sustainable software

Learn from the “App” World?

- Android Apps change interfaces every time I update them
- iOS apps do much better at this
  - But Apple uses a very different governance model for iPhone apps

Going "Open Source" is more than...

- Slapping an OS License on
- Creating a github presence
- Responding to the occasional email
- A lot of pre-existing projects in DOE have gone this route. Did that make them better?
- Need to be able to demonstrate quality
- Need to solve a problem others value
- Needs to be easier to start with pkg X than from scratch
  - Requires documentation, developer support, etc
  - pkg X needs to be preceived as sustainable