

# Software Engineering Issues in Moving Legacy Codes to Future Architectures

Charles R. Ferenbaugh  
Los Alamos National Laboratory

SWP4XS Workshop  
January 13, 2014

LA-UR-14-20128

# Why legacy codes? Why not write new codes?

- Legacy codes are in constant use
  - We'd need to maintain, upgrade them in parallel with new code development
  - We probably won't have resources for two full code teams per code
- Legacy codes have a lot of V&V, both formal and informal
  - It takes a lot of effort to reconcile differences between legacy and new codes

*LANL ASC approach: upgrade legacy codes incrementally, in place*

# Legacy code issues we've encountered

- Coding style is difficult to understand, modify
- Global data structures make it hard to restructure data, follow data flow
- Test systems, debugging approaches aren't designed to handle non-reproducible results
- Numerical instability in legacy algorithms will amplify small changes in inputs



# Research areas that would help



- Design patterns and strategies for incremental modernization of large legacy codes
- Code analysis tools to help in understanding legacy code organization
- Testing frameworks that are suited to massively parallel architectures and new programming models
- Code analysis tools and processes to help determine code correctness, as supplements to testing

