Session 11: Code Development within the FUN3D Framework

Mike Park
Computational AeroSciences Branch
Is There a Better Way to Develop Software?

- Development of Langley CFD solvers has traditionally been a 1- or 2-person operation
- Lacked a rigorous testing environment
- Standardization and portability often overlooked
- Version control became a nightmare
- Low-level collaboration within LaRC very difficult; off-site collaborations virtually impossible
- Amount of source code and expertise needed for today’s advanced algorithms quickly becomes unwieldy
Modern Software Development Practices

• Formal version control using Subversion; repository sits outside LaRC firewall
• Enables anyone anywhere to work directly on our source code in real-time
  - Frequent commits encouraged (~1 hour or less; real-time view on website)
  - Integrates all capabilities in one centralized suite
  - User support vastly easier
• Documented coding standard ensures uniformity, portability
  - Pre-commit script checks for adherence, rejects sloppy code
  - Enables automated scripting of templates, other extensive source code manipulations
• Hundreds of tests performed continuously around-the-clock on wide range of compilers/hardware: x86/Sun/SGI/DEC/Mac/Opteron/HP/etc
  - Failures reported to team members immediately via email, SMS
  - Bugs confined to a few lines of code and identified within an hour, rather than thousands of lines developed over months
  - Compiler bugs identified routinely
Modern Software Development Practices

• Weekly scrums foster collaboration and communication
  - Management/observers encouraged to attend, but only workers allowed to talk
  - One at a time, each member reports “did – doing – will do – in the way”
  - Scrum master notes impediments to progress, strives to remove them
  - Goal is to be done in 15 minutes
  - No “Death by PowerPoint”
• Additional discussion via mailing list, wiki
• Website also kept under Subversion, maintained collectively by entire team
  - Automatically generated and placed on server whenever text files in repository are updated
  - Team members need not know fancy HTML to contribute
• Unit testing slowly becoming more prevalent
  – Tough to retrofit legacy code
  – Released “FUnit”, framework for unit testing Fortran code at http://nasarb.rubyforge.org/
• Pair-programming practiced regularly
Compiler Gauntlet and Build Dashboard

- Hierarchy of cascading continuous builds keeps close tabs on code status
- Failures reported to team members via E-mail/SMS
- Bugs typically found/corrected in minutes
- Interactive GUI allows access to current and archived build information
- If a revision passes the entire hierarchy, it is tarred up, ready for external release
- Website also built continuously

- Continuous integration for warnings across ~10 major compilers
- Interface allows team members to see any complaints/warnings
FUN3D Development Widespread Developers work on central source code in real-time

Advanced Engineering Environments Branch
• Dana Hammond - HPC, computer science
• Bill Jones - geometry, gridding, adaptation

Aerothermodynamics Branch
• Karen Bibb – high-energy applications
• Peter Gnoffo - high-energy algorithms
• Bil Kleb - software practices, applications

Computational AeroSciences Branch
• Bob Biedron - dynamic simulations
• Jan-Renee Carlson – turbulence, jet flows, bc’s
• Mark Carpenter - solvers
• Beth Lee-Rausch - applications
• Eric Nielsen - solvers, adjoints, design
• Mike Park - cut-cells, adaptation
• Chris Rumsey - turbulence
• Jim Thomas - solvers, discretizations
• Veer Vatsa - applications
• Jeff White - hypersonics

Flow Physics and Control Branch
• P. Balakumar - turbulence

NASA Glenn

National Institute of Aerospace Academia
• Georgia Tech
• MIT
• NC A&T
• Penn State
• U. of Tennessee-Chattanooga
• U. of Wyoming

OGA
• US Army/AMRDEC-Huntsville
• Argonne, Oak Ridge National Labs

Industry Visitors/Students

Very broad mix of theoretical, development, and applied personnel:
• Fundamental research
• Real-world applications