Addressing the Software Application Productivity Challenges for Extreme-scale Computing

13-14 Jan 2014, Gaithersburg, MD

Software Engineering Institute | Carnegie Mellon

DOD HPC MODERNIZATION PROGRAM

Dr. Douglass Post

HPCMP Chief Scientist & Associate Director CREATE

Approved for public release; distribution is unlimited. HPCMP 14-1 (6 Jan 2014)

Key Points for Discussion

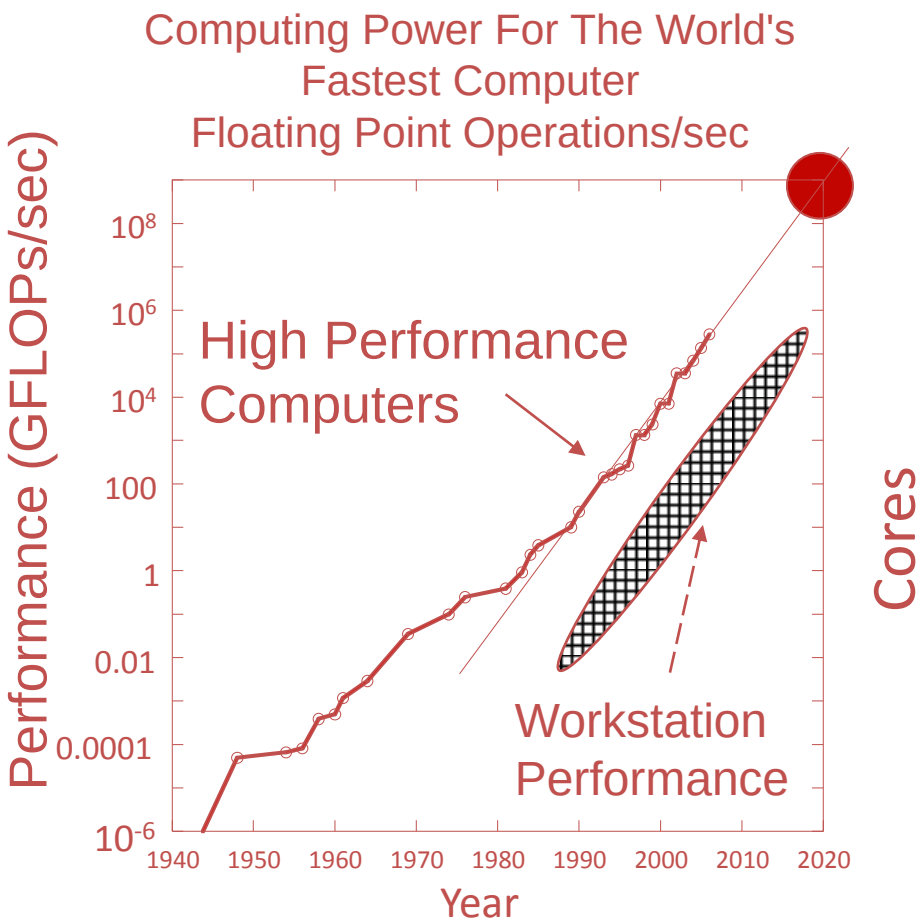
1. **The Challenges**
2. **The Context**
3. **Addressing the Challenges**

Key Points for Discussion

1. **The Challenges**
2. **The Context**
3. **Addressing the Challenges**

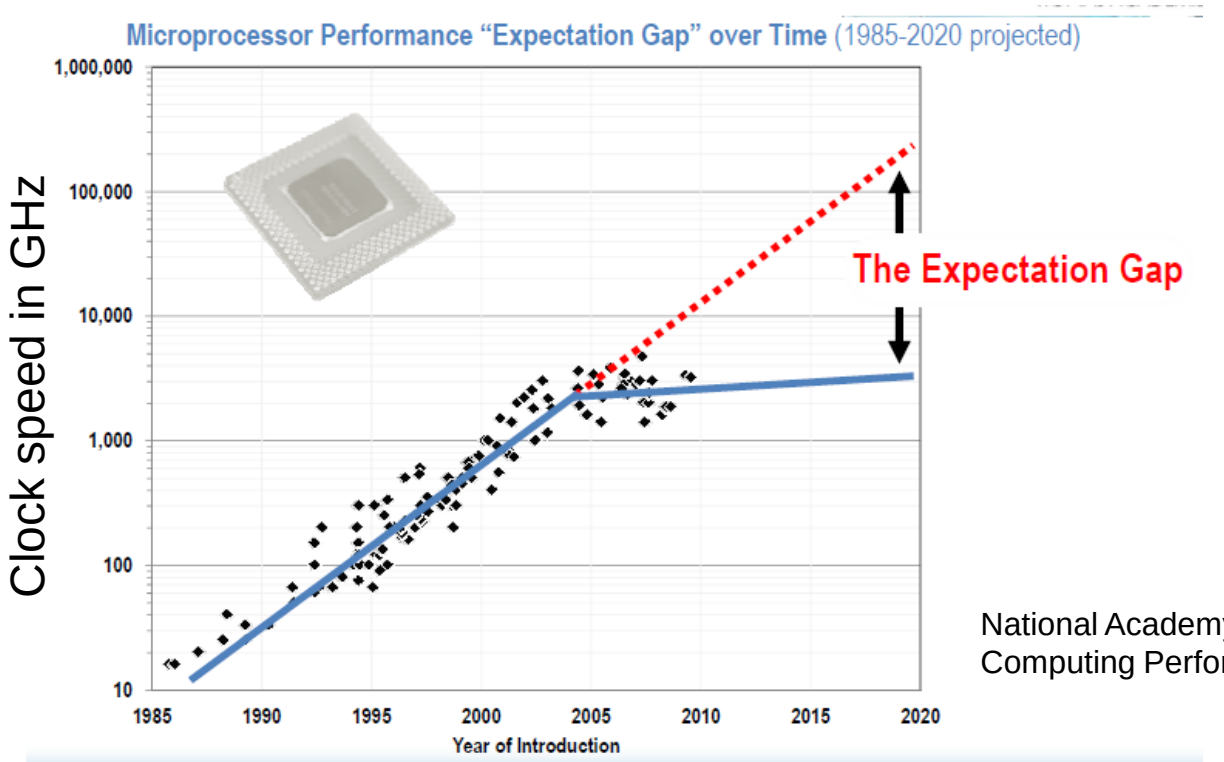
The Potential

- **$10^{15\text{-}18}$ increase in computer power since 1945 enables the development of HPC software applications that can:**
  - **Utilize accurate solution methods**
  - **Include all the effects we know to be important → multi-physics**
  - **Model a complete system**
  - **Complete parameter surveys and analyze data from experiments with usable turnaround times**
- **In ~ 10 years, workstations will be as powerful as today's high performance computers**
- **Software applications capable of exploiting this computer power are the missing link!**



Moore's "Law"

A Major Challenge: Computer Complexity



- **Due to the "Power Wall", clock-speed will not increase exponentially**

National Academy Study: The Future of Computing Performance (2010)

- **Feature size can still decrease, so feature & core density will increase**
- **Memory/core will decrease even as memory takes more chip space**
- **Massive parallelism and special purpose processors are in our future†**
- **Applications will need to be modified to run on these next-generation computers**

† D. Patterson, "The Trouble with Multicore", IEEE Spectrum, JULY 2010

Development and Sustainment of "Virtual Research and Test Facilities" Involve Immense Challenges*

- **Scientific and Engineering:**
  - Trade-off many different strongly interacting effects across many orders-of-magnitude of multiple time- and distance-scales
  - Verify and validate highly-complex applications and quantify the uncertainties
  - Develop problem generation methods for larger, more complex problems
  - Analyze and visualize larger, more complex datasets
- **Project:**
  - Evolve from small teams to large multi-disciplinary and multi-institutional code development teams
  - Achieve stable support for teams and provide customer support
- **Programming:**
  - Changing computer architectures every 3-4 years.
  - Develop and run codes for computers that will be 10^2 to 10^4 faster with 10^2 to 10^3 times more processors and greater memory architecture complexity than today
  - Achieve efficient performance for next-generation computers
  - Develop codes and run massively-parallel applications with relatively immature tools
- **Present Federal communities have very limited experience developing engineering applications → additional challenges**

*c.f. The Opportunities, Challenges and Risks of High Performance Computing in Computational Science and Engineering, *D.E. Post, R.P. Kendall and R.F. Lucas, Advances in Computers, Quality Software Development, 66, (2006), M. Zelkowitz, Ed., Academic Press pp. 239-301.*

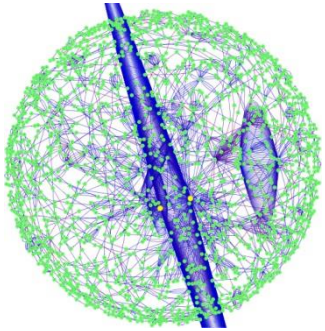
Key Points for Discussion

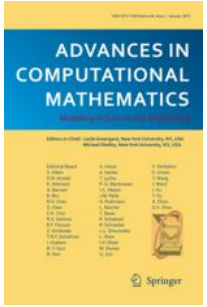
1. The Challenges
2. The Context
3. Addressing the Challenges

HPC Becoming Ubiquitous Tool for Research, Prediction, Testing and Design

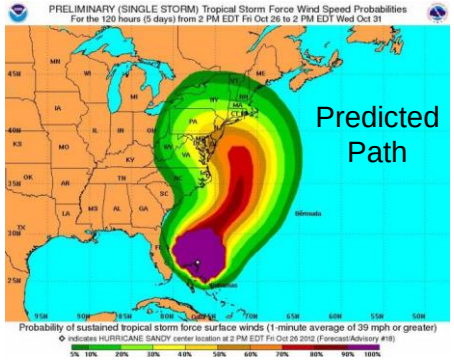
- **Computational modeling is part of experimental and theoretical work in almost all the hard sciences and engineering, and social sciences as well**

Political Science Agent-based simulation graph

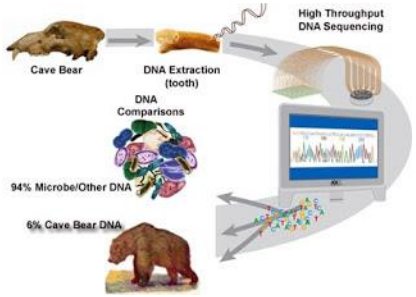




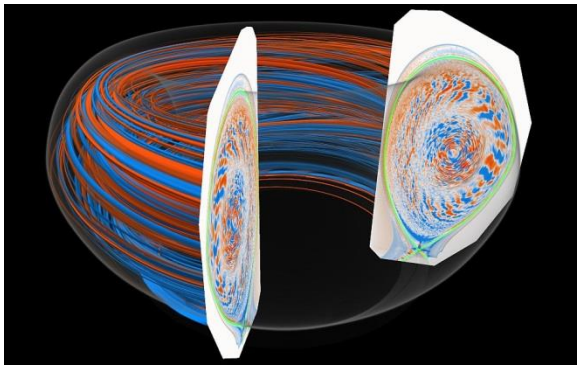
Hurricane Sandy



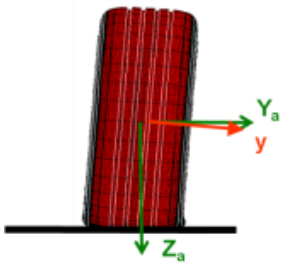
Gene Sequencing in Paleontology



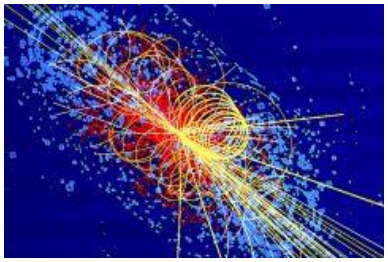
Calculated Tokamak 3-D Turbulent Flow



Calculated Tire Performance Hydroplane test rig



Large Hadron Collider Higgs Boson Tracks



"It takes a Village"
HPC Science-based Research, Test and Engineering Ecosystem is Complex

Sponsors

SME*

Customers

Theoretical Researchers

Experimental Researchers

Product Development Engineers

Codes | **V&V** | **Networks** | **Computers**

Multi-science, Multi-scale Research, Test and Engineering Applications

V&V Expts.

High-Speed, High-Bandwidth, Network

HPC Computers

Data Storage

Portals

Code Development Services

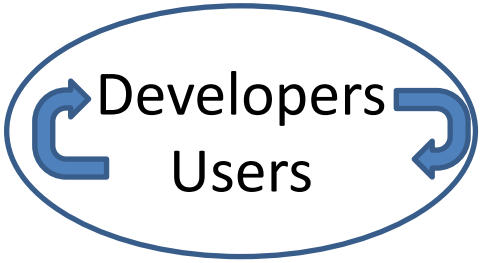
*SME=Subject Matter Experts

Approved for public release; distribution is unlimited. HPCMP 14-1 (6 Jan 2014)

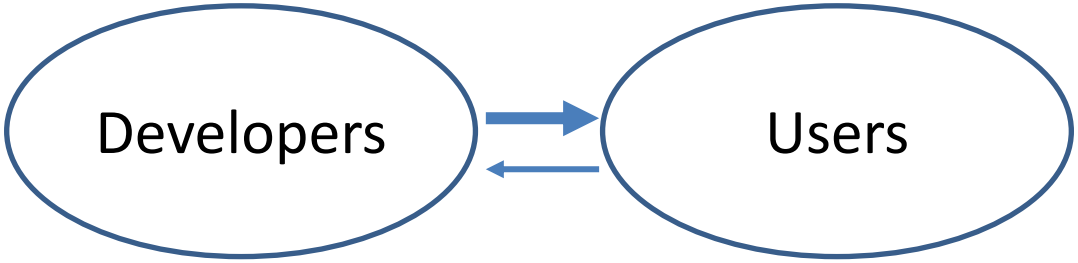
Application Paradigm is changing

- **In the past, code developers were often the major users**
  - SMEs had detailed understanding of the code's strengths and weaknesses
  - Codes not designed for general use (focused V&V, hard to use, minimal documentation, …)
- **Now, many SMEs use codes developed and supported by others**
  - Examples: Chemistry and Materials, Fluid Mechanics, Structural Mechanics, Climate and Weather, … e.g., GAMESS, NWCHEM, OVERFLOW, NASTRAN, WRF, FLASH, ISV codes,…
  - SMEs less aware of code strengths and weaknesses

1960s → today



1990s → today

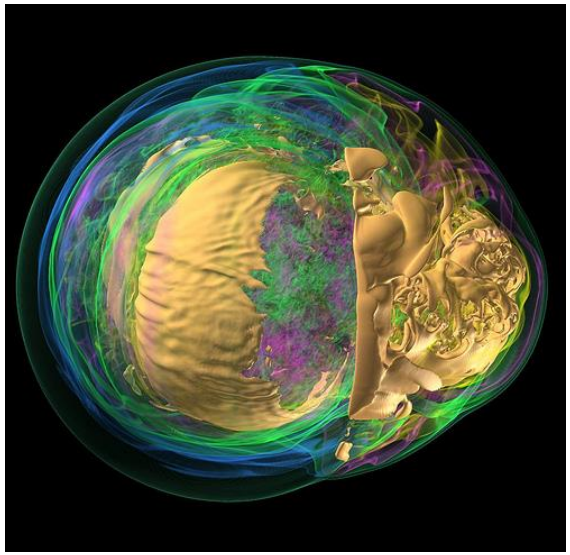


Virtual and Experimental Facilities

- **Today→ Two Types of Facilities**
  1. Experimental Test and Research Facilities
  2. Virtual Ecosystems for Research and Testing
     - Virtual Facilities include codes, computers, networks, etc.
     - Like Experimental Facilities, Virtual Facilities require sustainment and modernization support

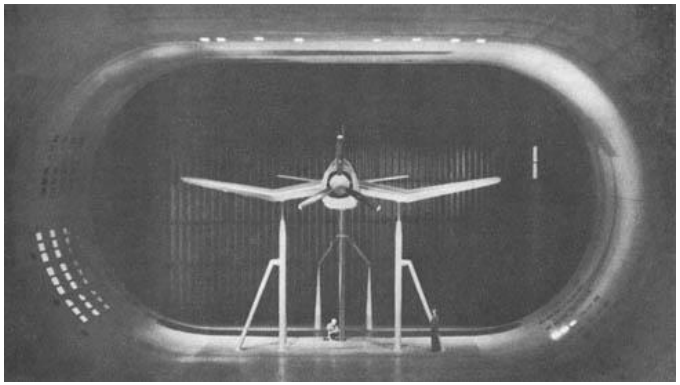
Research Facilities

Very Large Array Radio Telescopes

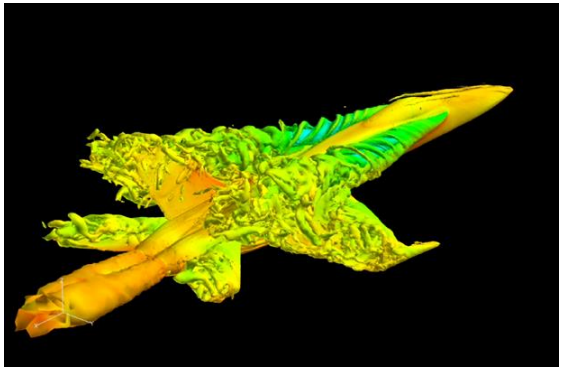


Supernova Simulation-FLASH

Test Facilities



Douglas XSB2D-1 airplane in NASA 40- by 80-foot tunnel.



Kestrel F-18 Simulation

HPC Ecosystems are Virtual Facilities

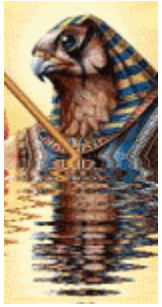
- **These ecosystems share the needs of design instruments and facilities**
  - All require sustainment & continuous support, maintenance & modernization
  - Ecosystems must be reliable, accurate (extensive V&V), easy-to-use, extensively documented, flexible, maintainable, extensible, portable, accessible
- **Need process for producing and sustaining codes, i.e., develop and implement required software engineering practices & processes, etc.**
- **Need explicit financial support for whole life cycle just like "real" facilities**
- **Decision makers understand the need to explicitly fund physical facilities (accelerators, large telescopes, satellite and planetary probes, test centers,…), but don't understand the need to support all elements of the HPC infrastructure, especially code development and support**
- **All elements of the HPC ecosystem (computers, networks, code development and support teams, etc.) require financial support**

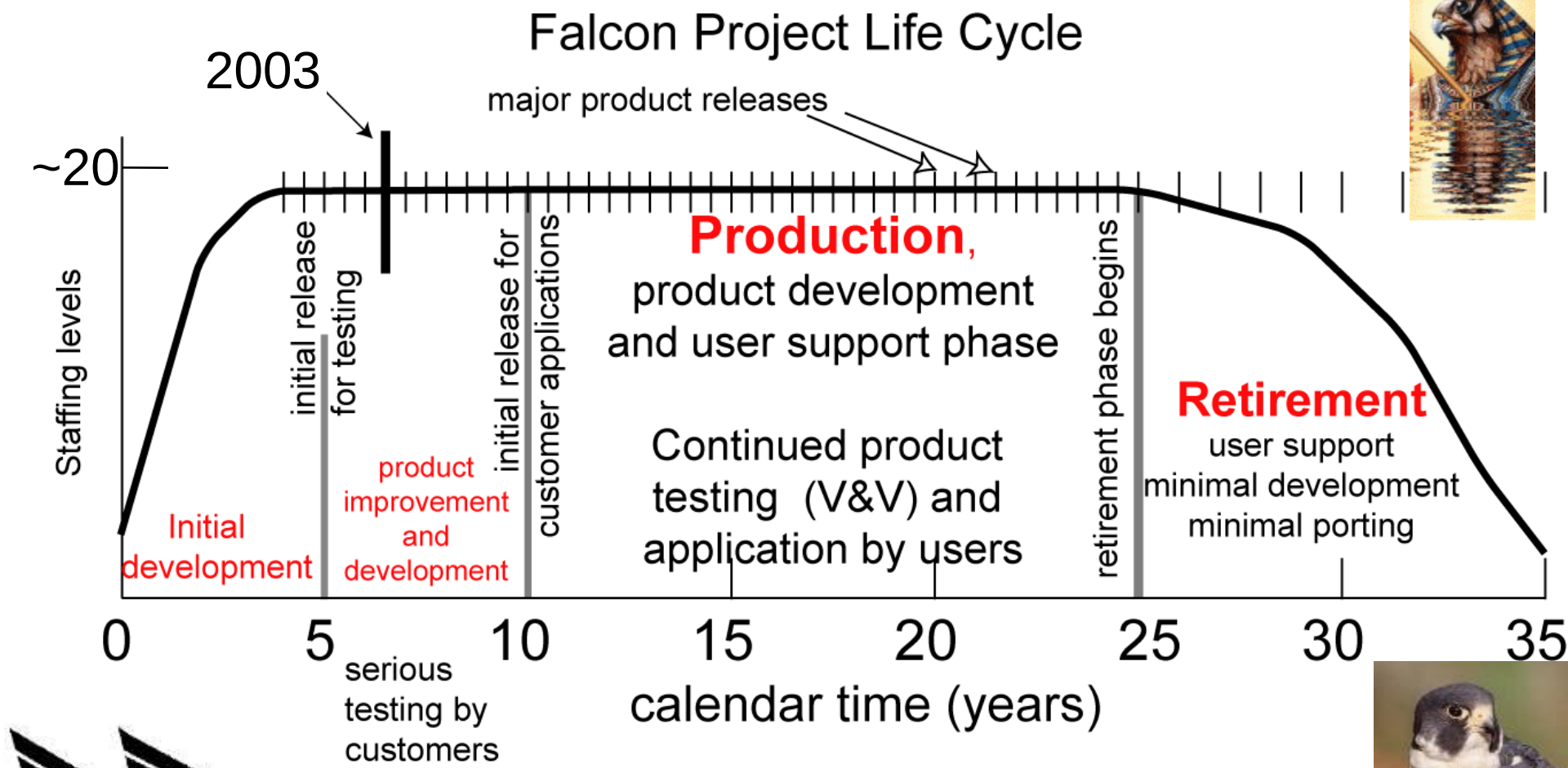
Science-based Code Productivity Challenge

- **Codes developed & used by researchers are generally successful**
  - Criterion for success is published research results, metrics: published papers
- **Development of science-based codes for use by others takes a long time (≥ ~10 years) and a large multi-disciplinary team (~5 to 20), longer and larger than pure research codes**
  - e.g., NWCHEM, GAMESS, NNSA Design Codes, ISVs; case studies (DARPA HPCS Phase II)
- **Codes developed for use by others have more stringent requirements for success**
  - Useful for others for successful research or product design measures level of adoption, impact,…
  - Need adoption into customer community work flow, measures: level of adoption, impact,…
  - Higher level of software quality, user support, documentation, … Very important for success
- **Evidence is that successful development rate for such codes is much lower.**
  - Relatively few research codes graduate to "community" codes, measures: survival rate, # of users
  - Many contractor-built large-scale complex software projects failed (FBI, NSA, FAA, …) at the $100M to $2B level
  - Many ISV products are not profitable

Approved for public release; distribution is unlimited. HPCMP 14-1 (6 Jan 2014)

Science-Based Codes Have a Long Life Cycle (>30 years)





From: D. Post, R. Kendall and E. Whitney, "Case Study of the Falcon Code Project", Proceedings of the Workshop on Software Engineering for High Performance Computing, International Conference on Software Engineering, May 15, 2005, St. Louis, Missouri.

Key Points for Discussion

1. The Challenges

2. The Context

3. Addressing the Challenges

Software Application Productivity is a Key Issue for the Extreme-scale HPC Ecosystem

- **Our extended community has a lot of experience with computers and networks**
  - Extreme-scale computers and networks have many challenges, but there is a lot of focus and resources committed to meeting those challenges
- **Relatively little effort has been focused on the producibility of Extreme-scale software applications**
  - Without software applications there is no business case for Extreme-scale computing (or any other type of computing)
- **Explicit resources are needed to address the challenges for development, deployment, sustainment and user support of Extreme-scale software applications**

Approved for public release; distribution is unlimited. HPCMP 14-1 (6 Jan 2014)

Need Appropriate Software Engineering Processes and Practices for Productivity

- Need software engineering processes and practices specifically designed for science-based codes
- Base them on data and analysis, e.g., case studies (DARPA HPCS Phase II), community experience, and lessons-learned
- Measure of success was measured by number of users, their scientific and programmatic impact, longevity, and ability to meet users needs in a timely manner
- **Need funds and process for developing, deploying, and sustaining these codes and supporting their use for the whole ecosystem life cycle**
- Need appropriate development tools and environments

Examples of Success –V&V and SE on GAMESS, FLASH, DoD HPCMP CREATE, …

- **GAMESS—community chemistry code led by Iowa State (Mark Gordon)**
  - Iowa State (Mark Gordon), ~ 10 staff on site, 150 contributors, ~6000 research papers written based on GAMESS use, distributed funding
- **FLASH—Astrophysics (Super-Nova) Code (U. of Chicago)**
  - Funded by DOE NNSA ASCI Alliance and NSF
  - Community code widely used by astrophysical and HEDP community
- **DoD HPCMP Computational Research and Engineering Acquisition Tools and Environments (CREATE)**
  - Program to develop design tools for DoD Air Vehicle, Ship and RF Antenna Major weapons systems; funding started in FY2008
  - 9 separate distributed code teams
  - Multi-physics, multi-scale, HPC software
  - At half-way point, CREATE software being used by ~ 70 DoD Programs (government and defense industry)

Metrics: # of users, impact, publications, longevity, financial support, …

Example: Sample Best-Practices for Science-based HPC Software Applications (DoD)

Development Team

1. **Lean (<10), close-knit development teams led by technical experts.**
2. Multi-disciplinary development team linked with relevant technical communities.

Customer Focus

3. Strongly connects to, and overseen by, senior stakeholder and user communities.
4. Pilots projects to solicit customer reaction and feedback.
5. Frequent reporting to stakeholders.

Technical Maturity

6. Use of proven software technologies.
7. VV&UQ in alignment with NRC/NAS best-practices for science-based codes.

Development Methods

8. Milestone-driven workflow management with flexible workflow execution (e.g., SCRUM) and timed (e.g., annual) releases.
9. Configuration management and supportive code development environment.
10. Multi-platform code builds based on extensive testing.
11. Adequate code documentation.

Requirements Definition

12. Reliance on use cases and iteration with prototypes to define requirements.

Points for Research Program Proposals

- **Identify Software Engineering Processes and Practices that lead to Success**
  - Case studies needed to identify barriers to success
  - Capture lessons-learned from successful and unsuccessful code projects
  - Fund experiments (code development projects) to test candidate processes and practices
  - Identify useful tools and development environments
- **Develop Scalable Algorithms and Development Methods for Next-Generation Computers**
  - Incorporate those algorithms in libraries that hide computer complexity from the developer
  - Develop and support code development tools and environments, including a good parallel compiler
- **Identify and develop methods for V&V and UQ**
- **Develop approaches and methods to develop, deploy and sustain HPC Ecosystems for the full life cycle**
  - Without explicit financial support for all of the HPC Ecosystem for the whole life cycle, scientific and engineering computing won't deliver on its promise

Approved for public release; distribution is unlimited. HPCMP 14-1 (6 Jan 2014)

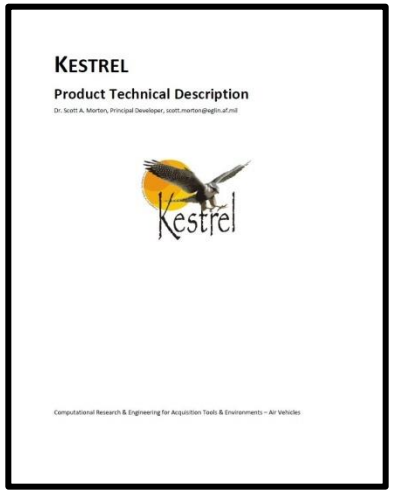
Reference Slides

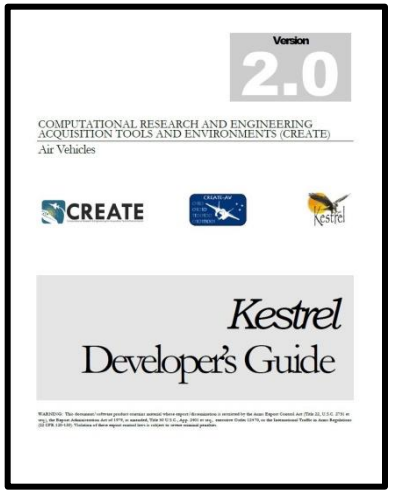
Large-Scale Codes Fail Regularly

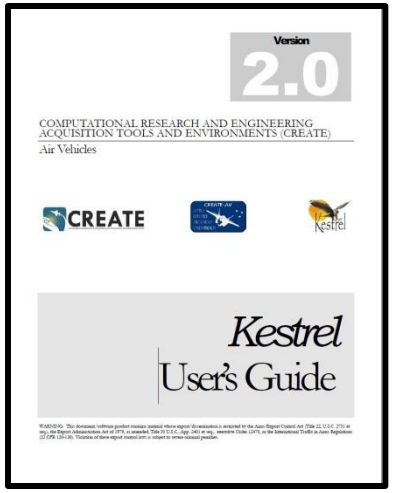
- **Between 2001 and 2004, the FBI spent $171M on failed Virtual Case File (VCF) program[1]**
- **Between 2005 and 2010, the FBI spent another $451M on Sentinel (son of VCF) that was also failing (Lockheed was the prime) and the FBI de-scoped it, and pulled it inside[1]**
- **NSA spent more than $1B on Trailblazer (SAIC) and never got a product[2]**
- **Lessons-learned:**
  - "because of their enormous complexity, large IT programs rarely succeed'
  - large government contractors, "IT cartels" are usually only out for the money; and
  - program mangers have a tendency to put a rosy spin on projects even when they are clearly in trouble." [1]
- **Small, highly-competent in-house and small contractor teams, good planning and oversight**

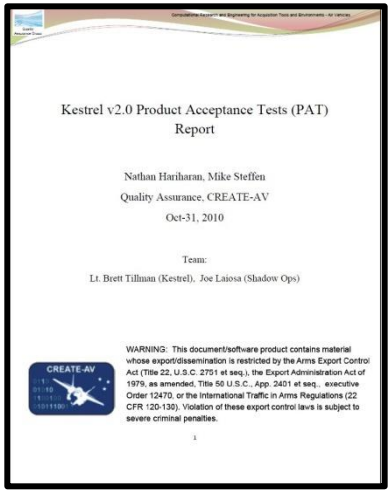
[1]J. Israel, "Why the FBI Can't Build a Case Management System", Computer, June 2012, pp73-80
[2]B. Koeppel, "Intelligence Contractor's Complex", Washington Spectator, June 15, 2012, pp1,3.

Kestrel Software Engineering Documents









CREATE Annual Product Release Cadence

Fiscal Year	FY2010				FY2011				FY2012				FY2013-planned			
Quarter	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
AV-DaVinci									1					2		3
AV-Helios		1						2					3	4		
AV-Kestrel				1		2							3		4	
MG-Capstone					1				2				3			
RF-SENTRI	1	1.5						2					3			
Ships-IHDE	1				2				3				4			
Ships-NavyFoam					1				2				3			
Ships-NESM	0.1				1				1.1				2.1			
Ships-RSDE									0.5				1.0			

- **Approximately every year, a fully-tested upgraded code with the new features identified in the roadmap is released**