B-CSE Software in Industry/Manufacturing

Lead: Richard Arthur
Presenter: Rick Arthur, Hai Zhu
Writer: Todd Simons

Breakout Participants

- Jim Kirby (NITRD)
- Todd Simons (RR)
- Wayne Miller (Livermore)
- Karen Tomko (OSC)
- Elaine Kant (SciComp)
- Rick Arthur (GE)
- Kevin Olson (NAG)
- Hai Zhu (DuPont)
- Paul Jones (FDA)
- Ram Sriram (NIST)
- Ahmed Taha (NCSA)
- Mike Heroux (Sandia)

Focus Themes: Sustainability of Software for Industry

Challenges

Legal

- Indemnity (Provenance of IP)
- End use IP protection
- Licensing models

Cost models

- Traditional licensing bars high end use
- Licenses prohibit performance comparison

ROI & Risk / Uncertainty

- Confidence in result impact
- Reduce cost risk (including $/core scaling)

Lack of “Building Codes” for Software

- Use context limited by inherent risks (support, documentation, process)
- Liability and certification - e.g., ISO 9000
- Talent burden + risk for workforce
- Validation suite + facilities for codes

Opportunities

Regulator-accepted codes

- Reduce cost/risk/time to market
- Industry-funded sustainability

Pipeline of new commercial codes (Gov / Uni)

- Highlight capabilities (TopX report)
- Encourage adoption / integration
- Validation process to promote trust in adoption
- Shrink-wrap into commercial entities

Funding-agency investment in scaling programs

- Underlying frameworks
  - Benefit broader end use, including ISV
- Curate “gold standard” canonical cases
  - For V&V + UQ
  - For competitive benchmarking
- Functional kernels for microservices

“Building Codes” for Software

- ISO 9000 style process, documentation
- Expands potential uses of software + value

Things to think about

- How to leverage the software process procedures the government and public institutes established;
- How to share the software developed by the industry without harm to the industry's competitiveness.
- Industry faces the same challenge in terms of software productivity and sustainability. How should we cooperate better?
- Cost of licensing and legal complication should be a big issue to keep industry to play any strong role. How we can get help to overcome the hurdle? open cooperation, standardization, certification?
- How coud technical community development, SE standardization, SC extraction, etc be beneficial to both sides?

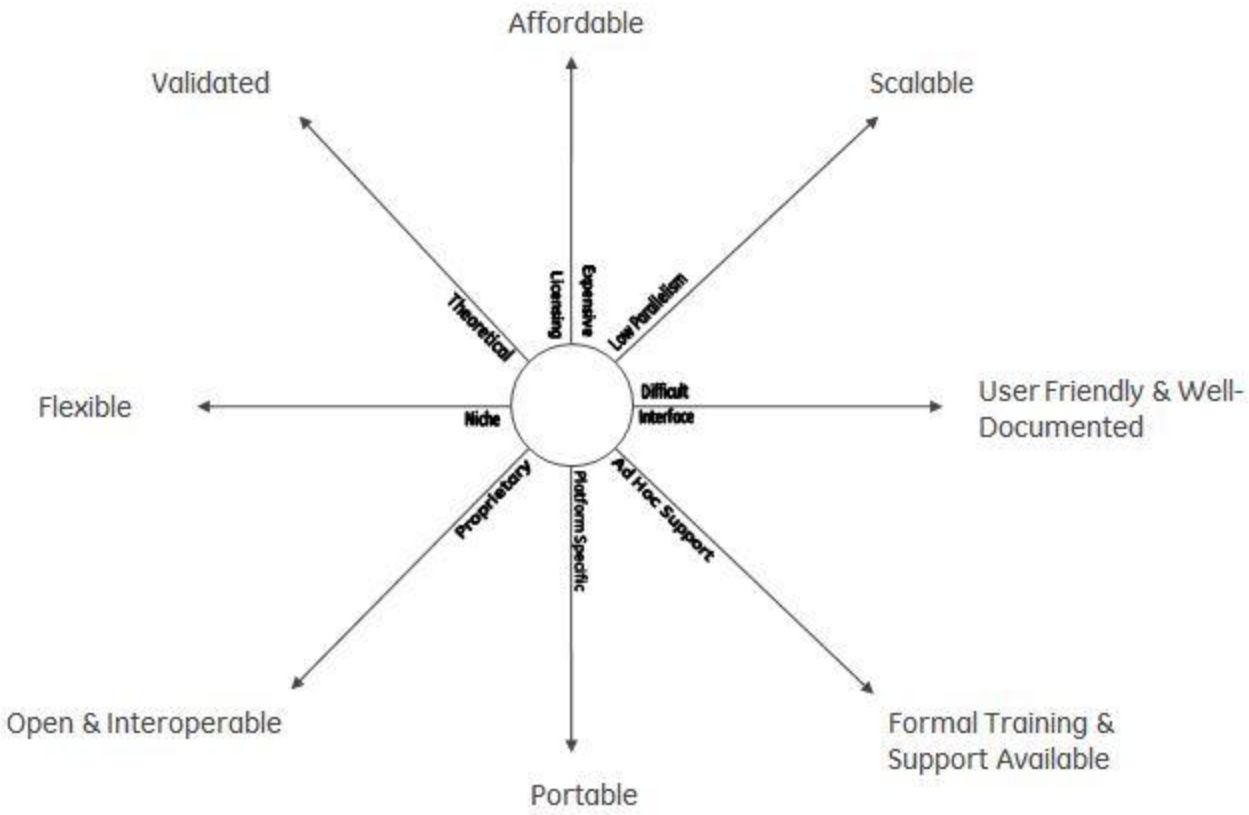
“Building Codes” for Software

Critical Infrastructure is deploying software, is designed on software

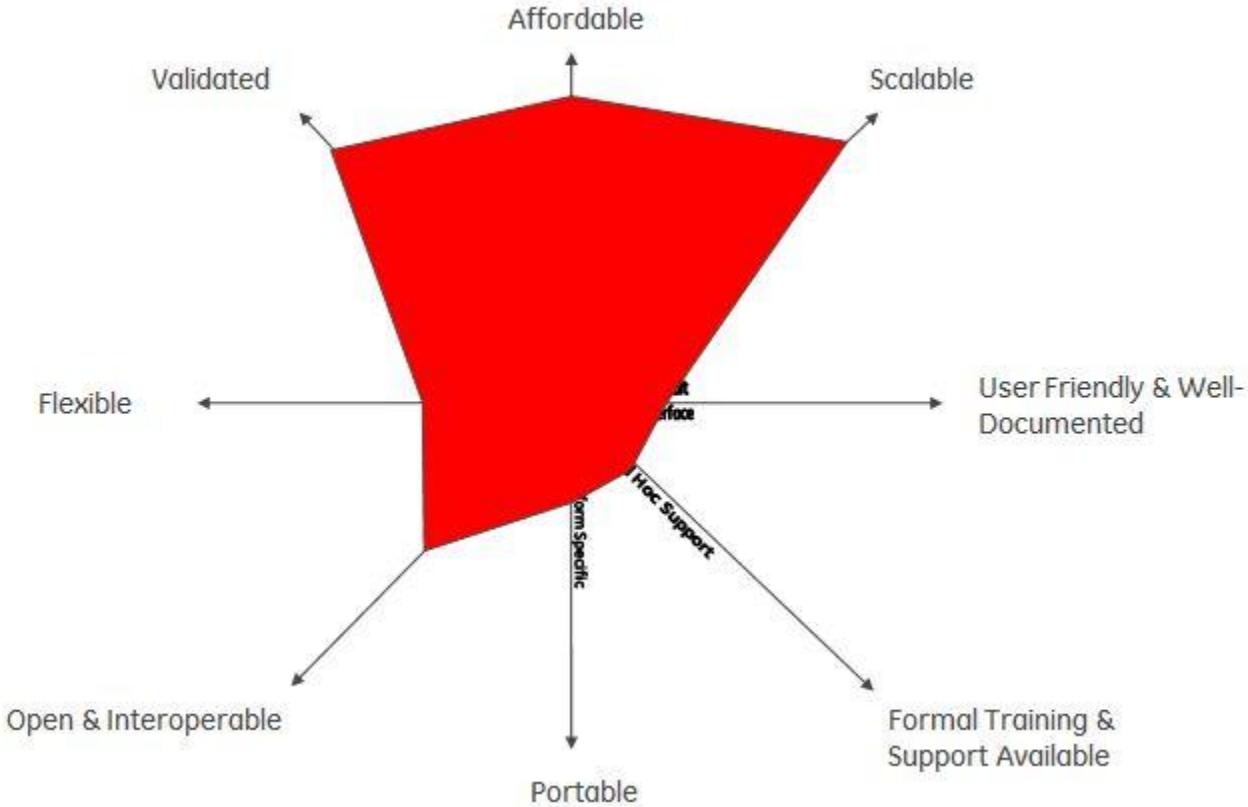
Physical Infrastructure (buildings, electrical, plumbing, etc.) have codes

- Public works can operate / fix
- Contractors can operate / fix
- Public relies upon
- Cannot be reliant on esoteric knowledge only in heads of individuals
  - Documentation
  - Testing
  - Reproducibility
- Sustainability means broader access / understanding to maintain
- Industry / Municipal etc. expected to (liable for) safety, stability, etc.
- Certification standards (e.g., ISO 9000) for open source origins?

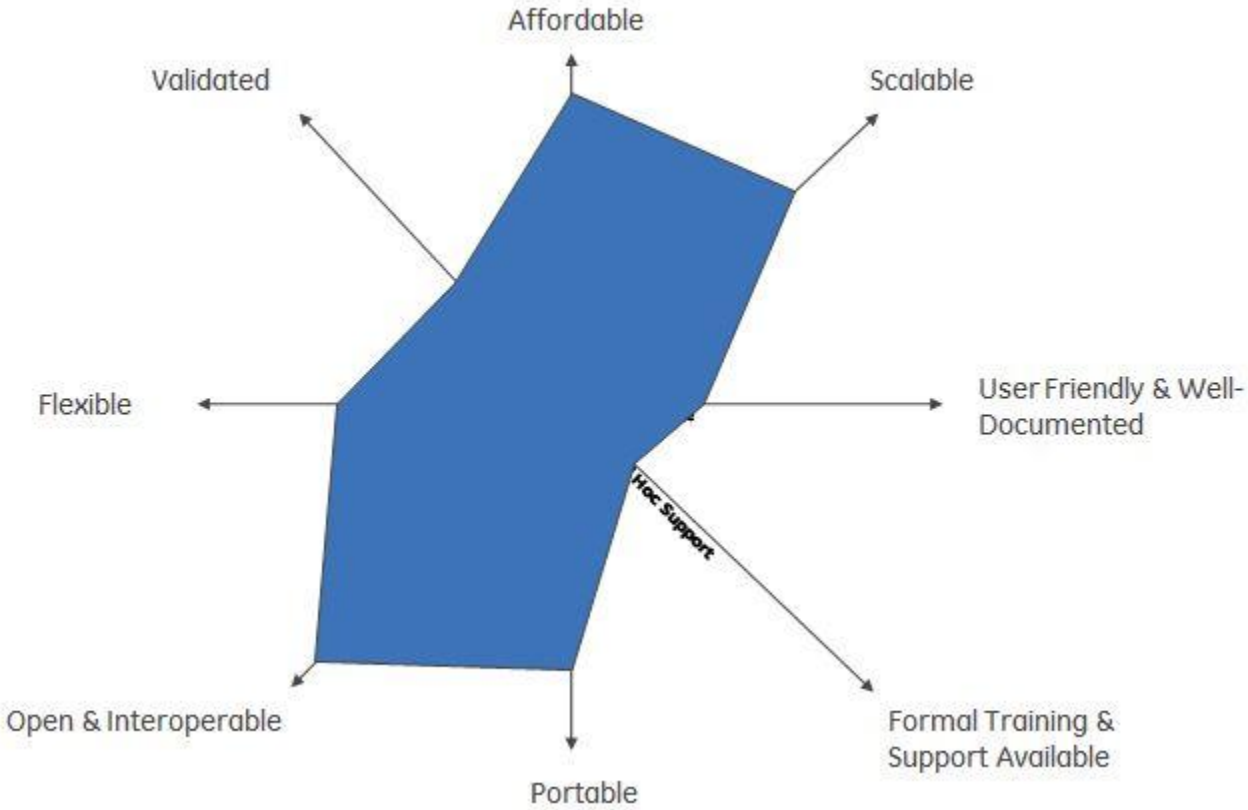
Software Characteristics



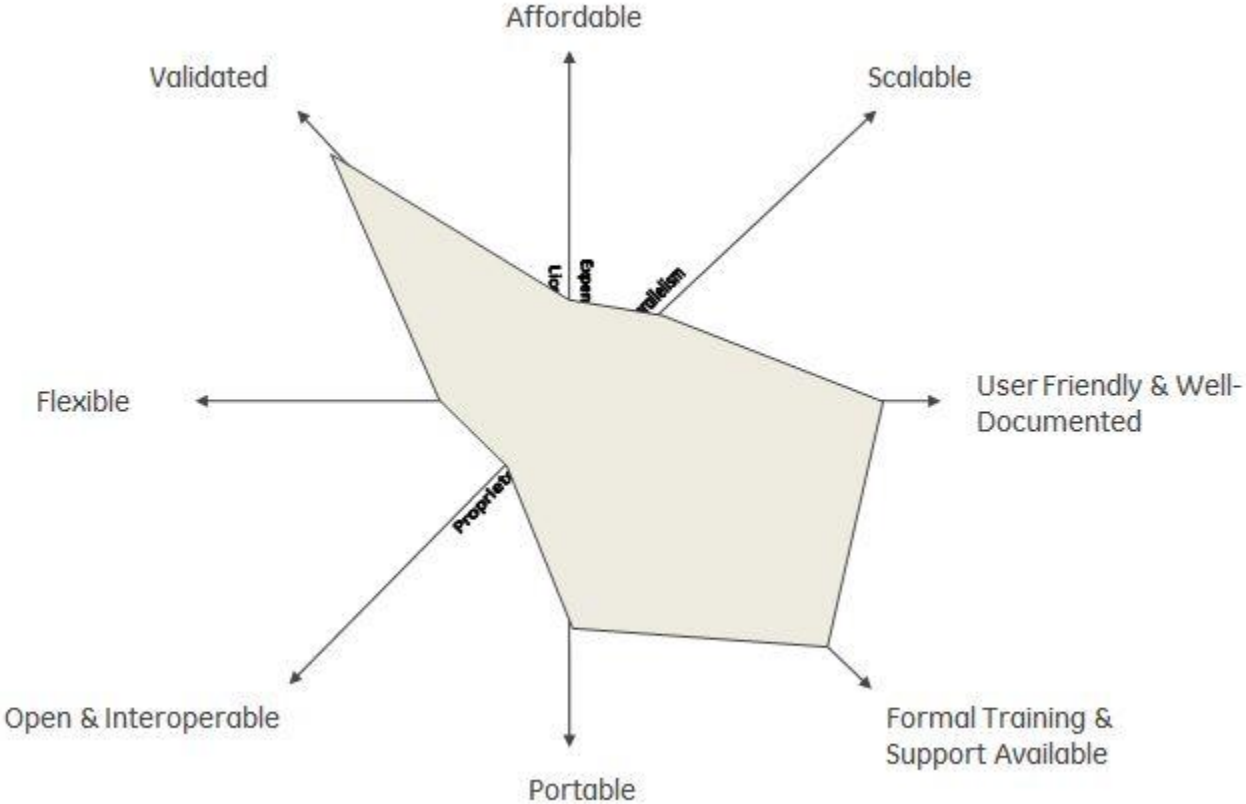
National Lab Codes?



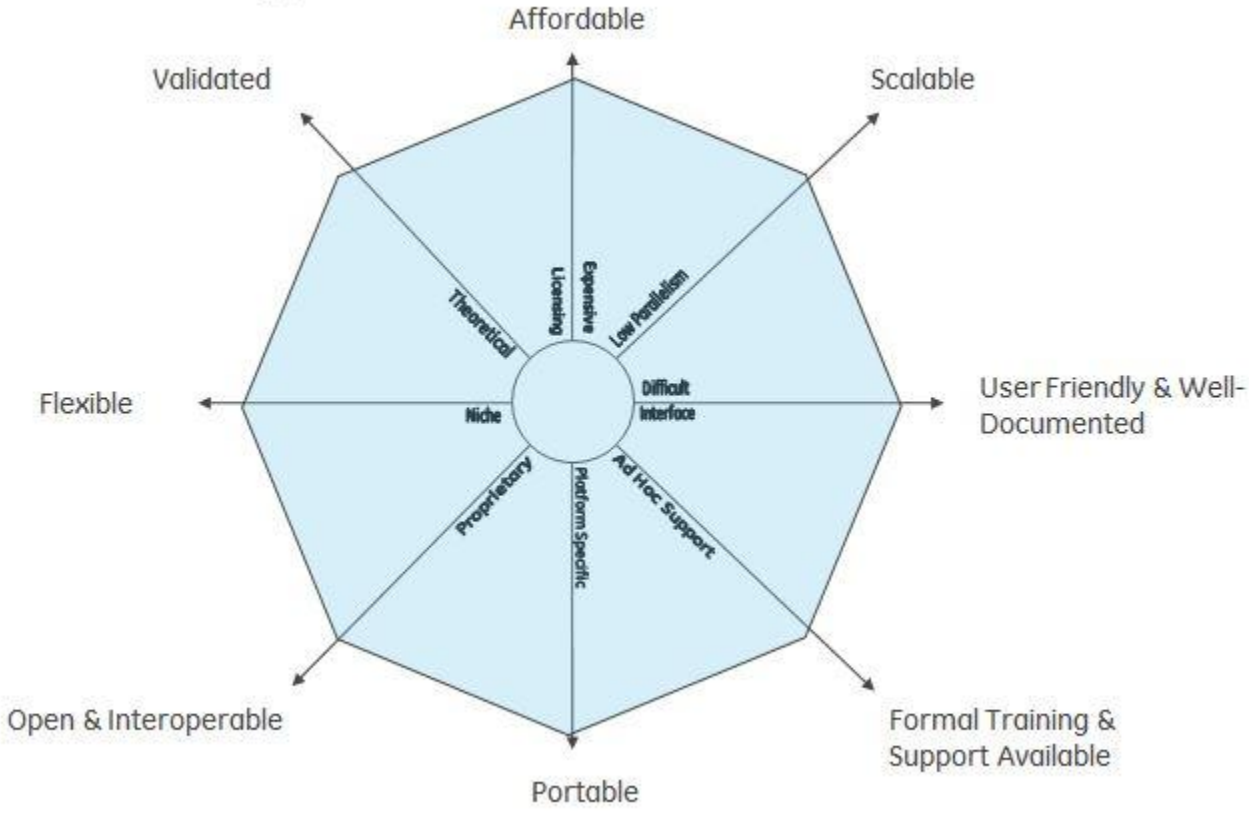
Open Source?


Affordable
Scalable
User Friendly & Well-Documented
Formal Training & Support Available
Portable
Open & Interoperable
Flexible
Validated
Hoc Support

Commercial Software?



How do we get here?



Open Source in the Industrial Internet

Risk level for distribution of run-time code (not source) using OS software under some common licenses

License	Incorporated		Linkable		Called		Dev Tool ONLY
	Modified	Unmodified	Modified	Unmodified	Modified	Unmodified	(internal use)
Public Domain [if confirmed]	Blue	Blue	Blue	Blue	Blue	Blue	Blue
MIT License	Green	Green	Green	Green	Green	Green	Blue
BSD-style	Green	Green	Green	Green	Green	Green	Blue
Creative Commons – Attribution (CC BY)	Green	Green	Green	Green	Green	Green	Blue
MS-PL (Microsoft Public License)	Green	Green	Green	Green	Green	Green	Blue
Python Software Foundation License	Green	Green	Green	Green	Green	Green	Blue
Artistic License 2.0 / Perl License	Yellow	Green	Yellow	Green	Green	Green	Blue
Apache License 2.0	Yellow	Green	Yellow	Green	Green	Green	Blue
Mozilla Public License	Orange	Yellow	Yellow	Green	Green	Green	Blue
Eclipse Public License (EPL)	Orange	Orange	Orange	Yellow	Green	Green	Blue
Common Public License (CPL)	Orange	Orange	Orange	Yellow	Green	Green	Blue
Creative Commons – No-Deriv's (CC BY-ND)	Red	Red	Red	Yellow	Red	Green	Blue
LGPL 2.1 / 3 **	Red	Red	Orange	Yellow	Yellow	Green	Green
Creative Commons – Share-Alike (CC BY-SA)	Red	Red	Red	Yellow	Red	Green	Green
GPL 2 / GPL 3	Red	Red	Red	Red	Orange	Yellow	Yellow
AGPL (Affero GPL)	Red	Red	Red	Red	Red	Red	Yellow
Creative Commons – all others	Red	Red	Red	Red	Red	Red	Red
No license identifiable	Red	Red	Red	Red	Red	Red	Orange

Industry Friendly

Not Friendly

** Current rating for LGPL licenses is based on unresolved ambiguity in interpretation of certain terms

Key:

Incorporated	OS code co-mingled with proprietary code (statically linked / linked at compile)
Linkable	OS code linked to proprietary code only at run-time (dynamically linked / not linked at distribution)
Called	OS code external to proprietary code even at run-time (no shared memory / separate process/thread)
Dev Tool	OS code is used in developing proprietary code (does not become part of distributed code / compilers)

Risk level:

Blue	Least risky
Green	
Yellow	
Orange	
Red	Most risky

Policy

- Legal & Compliance
  - Protection of corporate IP/Intellectual Property
  - Legal protection to practice/Indemnity - provenance
  - Regulatory implications (FDA, FAA, NHTSA, FRA, EPA, SEC, FTC, etc.)
- Restrictions on government code release competing with ommercial code
- Certification (EPA, FDA, etc.) - acceptance by agencies as stimulus to industrial adoption
  - Advantages: reduce costs, reduce risk, reduce time-to-market, improve regulatory oversight
  - Industrial pay-into sustainability of models to reduce costs of (compliance, certification, etc.)
  - Who owns and maintains these codes? Who / where are these run?

Risks

- Assurance, Risk & “Optics”
  - Provenance of data and code/libraries
  - Security (Data Protection + Malicious Access/Modification)
  - Address the stigma of “Corporate Welfare” vs. the subsidy of “Academic Benefactors” (university incubators, etc.)
- Orphaned software from government investments
- Licensing scalability (per-core licensing = HPC hostile)
  - Unlike physical manufacture - no incremental cost to vendor
  - Discourages use at the cutting edge

Promoting scalable software

- An opportunity to motivate ISVs would be to create a Top500 list (or Top10)
- Comparison of open/academic efforts to commercial codes.
  - Visibility into new approaches
  - Linking academic advances (professors/students) with commercial or (broadly standardized) open source
  - National Lab codes transitioning into commercial support
- It is noted that licensing often prohibits benchmarking
  - Vendor free to not submit, but curious by omission
- A body would be needed to curate test cases and provide test cases with adequate coverage of physics and multi-physics
  - DOE Office of Science / NSF / NIST? / The Platform / HPCWire / ?
  - Scalability Ranking for Critical Software Problem Solving
  - NIST for metrics - but cannot be the assessment or reporting vehicle

Economics

- Economics (ROI) of software and requisite hardware infrastructure (or hosted services)
  - Confidence in results (and confidence the results will have value) – to justify investment
  - Costs to employ within enterprise / across enterprise
    - Ability to use on problems of value (proprietary data)
    - Limits and/or obligations imposed in using
    - Costs to link with other systems (potentially internal/proprietary)
    - Ease of blending public/external data with private/internal data
- Types of software (commercial, in-house, open source) - interoperability, requirements

Economics (2)

- Potential Programs for Port & Scale
  - HPC4Mfg program (DOE/Livermore, Oak Ridge, Argonne) - sustainability elements?
  - DMDII (DOD/UI Labs/NNMI) - sustainability elements?
  - CORAL, etc. programs (NERSC, Livermore, ANL, ORNL) for scalability
    - funding to enable porting and scalability
    - access to talent to do so
    - no commercial codes currently in this code base

Economics (3)

- Reference Model Facilities
  - E.g., full physics, high-fidelity, validated, muiltiscale, multiphysics facility
  - Human Physiology: Virtualized human (system of organs, tissue, cells, *-omics)
  - Machinery: Full turbomachine (CFD, thermal, structural, chemical, materials, acoustics)
  - Government “gold standard” data sets available / maintained (possibly under license)
- Software enablement on commodity cloud environments
  - Software functional in cloud model
  - Hardware vendors - performant in cloud model