

Sustaining the software capabilities long term

Address Solutions as part of software.

Act on “Hard challenges are not technical” bringing in the right help.

Ensure Community-wide ownership of architecture and usability.



.. \$100M+/year on Track2 and Track1 ...how much ..on software development by award winners, by others..

*Want a # for CI ? >15 communities 10-30 s/w FTEs each >70M/year -
- but not software “development”*

End-to-end

Usability; “Right place” to put functionality and interfaces; Maturity/readiness risk assessment; Integration, operations, scalability testing; Security auditing/risk assessment; Error and fault information, translation and propagation; Diagnosis/troubleshooting; Documentation and training; Life-cycle planning (maturity management as well as evolution);

Solutions

Co-design of applications, infrastructure & fabric – user driven requirements, CS owned architecture, joint evolution & metrics. Watch for maturity issues.

With Communities

Separate h/w from software funding. Include application community “chits” to spend on external s/w developments. Reward reuse.

Central institute distributing support for software to be supported when current developers leave and support is negotiated across the communities who depend on it.

.. . evaluation criteria should be put in place for software development and maintenance? ... "close collaboration with a scientific user community" and "middleware should be general and usable by many different observatories and projects"

Require funding for maintenance, testing, deployment, support. Identify target, early adopter, user communities up front.

Repeat regularly risk assessment, evaluation of use usability, design, performance, security, re-training. Publish results.

Deployment and support of specific user communities part of deliverables/schedule.

Include thick application software. Identify and motivate functionality to move out of narrower application domain into commons (in the same community, outside the community)

Program for analysis of S/W projects patterned on VOSS ?

Individual and collective motivation: Organizational structure, scope, and scaling: Organizational life cycles: Production and innovation: Management, Governance, and Leadership: Measurement and assessment: Units and frameworks of analysis-both social and technical: Comparative performance:

.. Is there a (virtual) payment system that would allow other NSF awardees to indicate their desire for continued support of well-utilized software that they depend upon but do not develop themselves?

Address realities that support inevitably includes evolution (response to faults, extension to new requirements).

Community & agency investment in:

Forum for principles of what components should include/exclude. As important to understand what should Not be in software as what should be in. How we decide where the boundaries are. How we minimize complexity. Where there are and recommendations to deal with overlap & gaps. How and how long to accommodate legacy.

Planning for, identification of need for next generation of fabric & infrastructure.

Awareness of software aging and how to measure and evaluate: “Span of changes,” increases over time (changes to mature s/w can be more risky.); “Breakdown of modularity; ”Fault potential”.

Connection between data preservation and sustaining software. Address cost/benefit as well as potential of newer initiatives. Understanding of end-of-life – not just “drop dead”.

And must maintain Leadership in grand challenges and opportunities.

.. . Many existing elements of the science cyberinfrastructure were developed before other elements of the CI were in place. These existing elements have their own ad hoc approaches already built into many lines of code. How do we incent/fund projects to replumb themselves into the emerging CI, and how do we insure that new projects use the CI parts that are in place?"

Address Solutions as part of software by user driven development through collaborations across users, application developers, middleware, fabric.

Act on "Hard challenges are not technical" by sponsoring broader involvement in software programs.

Ensure Community-wide ownership of architecture, principles and usability through sponsored discussion, brainstorming, fora deliverables.