Technical and Management Issues in Collaborative Software Construction

Brent Fultz
California Institute of Technology

- DANSE Project past and present
- How we work as a project team
- Technical and management integration
- Post-Construction phase of DANSE?

Why DANSE ?

New Science

Better Science

Ease of Use

Software Stability and Reuse

Support Early Operations of the SNS

DANSE focus is higher-level analysis tools

Approximately, the responsibilities of DANSE and SNS meet at S(Q,E) or I(Q)

SNS Data and Reduction } S(Q,E) { DANSE Analysis Modeling

Collaborative Development Requires:

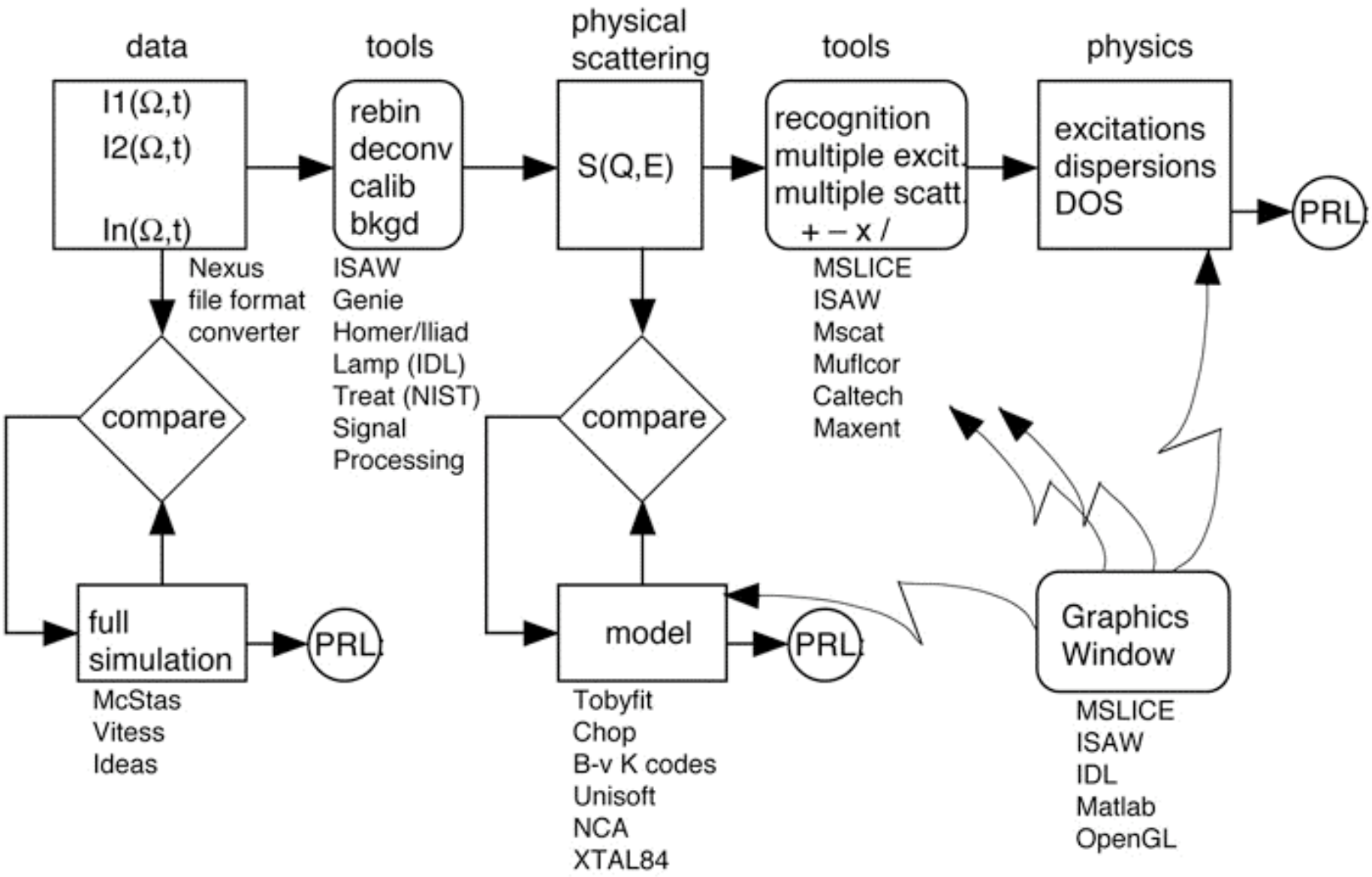
Shared Vision

Stable Expectations

Realistic Estimates

Architectural Coherence

Inelastic Roadmap of Sept. 2001 is Valid Today



People

The key resource.

- PI (1)
- Architect (1)
- Project Manager (1)
- Subproject leaders (4 physical scientists, 1 CS person)
- Developers (3 per subproject, in principle)
- Students
- Friends

Skills of DANSE Developers

- Postdocs and grad student developers use the software for their own work.
- Range of skills, usually weak in software engineering.
- Ideally, one professional developer per subgroup.

- Python is a big help in learning O-O programming.
- Weekly meetings use several UML diagrams and design patterns
- Documentation seems okay.
- Version control is fully accepted.
- Installation, builds, system skills are not for everyone.
- Pyre framework is helpful in forcing architectural discipline, but remains a challenge.

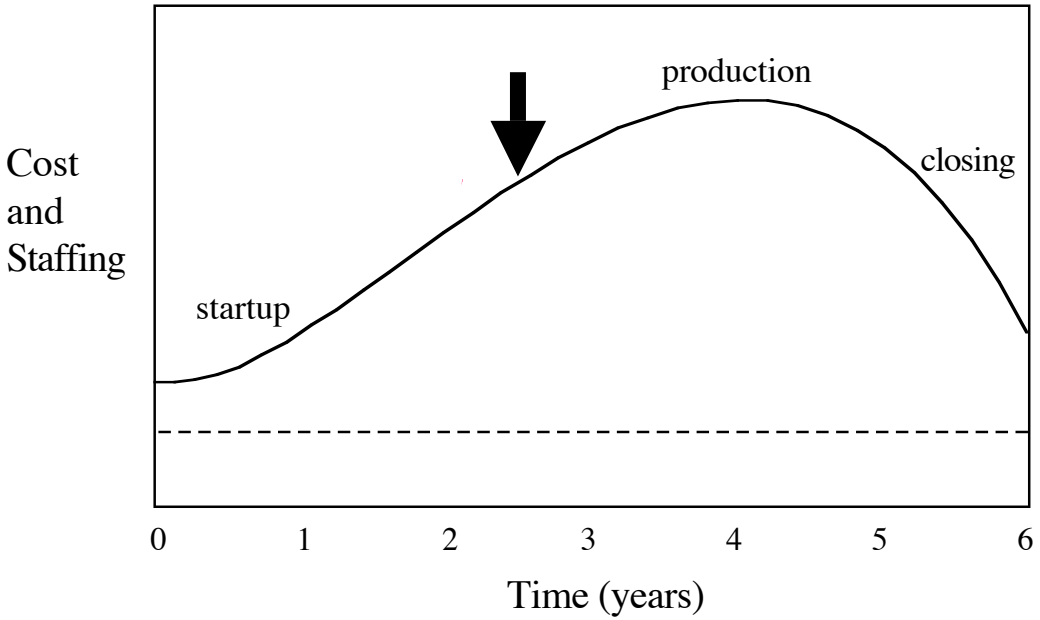
Interactions of People

Essential for vision and cohesion.

- Weekly Developers' Teleconference with vnc
- Weekly Management Teleconference
- Developers' Meetings (2 days each, 3 per year)
- Developer travel
- Phone calls with Project Manager or PI.

DANSE is a Project

- A project is temporary; produces a one-time product.
- 13 M$ over 6+ years.
- DANSE is 32% complete.
- For software development practices, what you see today is approximately what you get.
- Scope and features have been stable since design phase.
- Productivity is good. We can do what we promised.

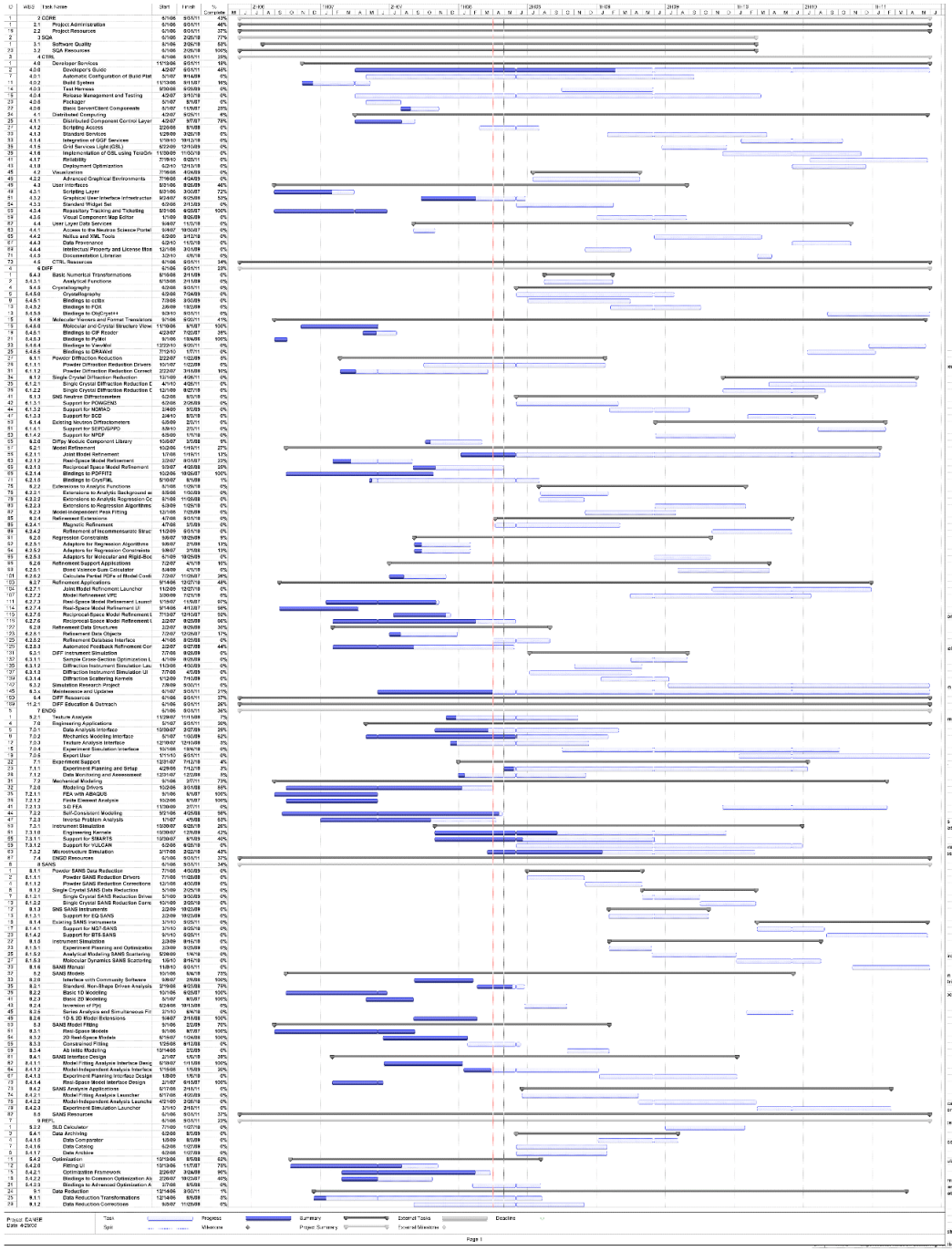


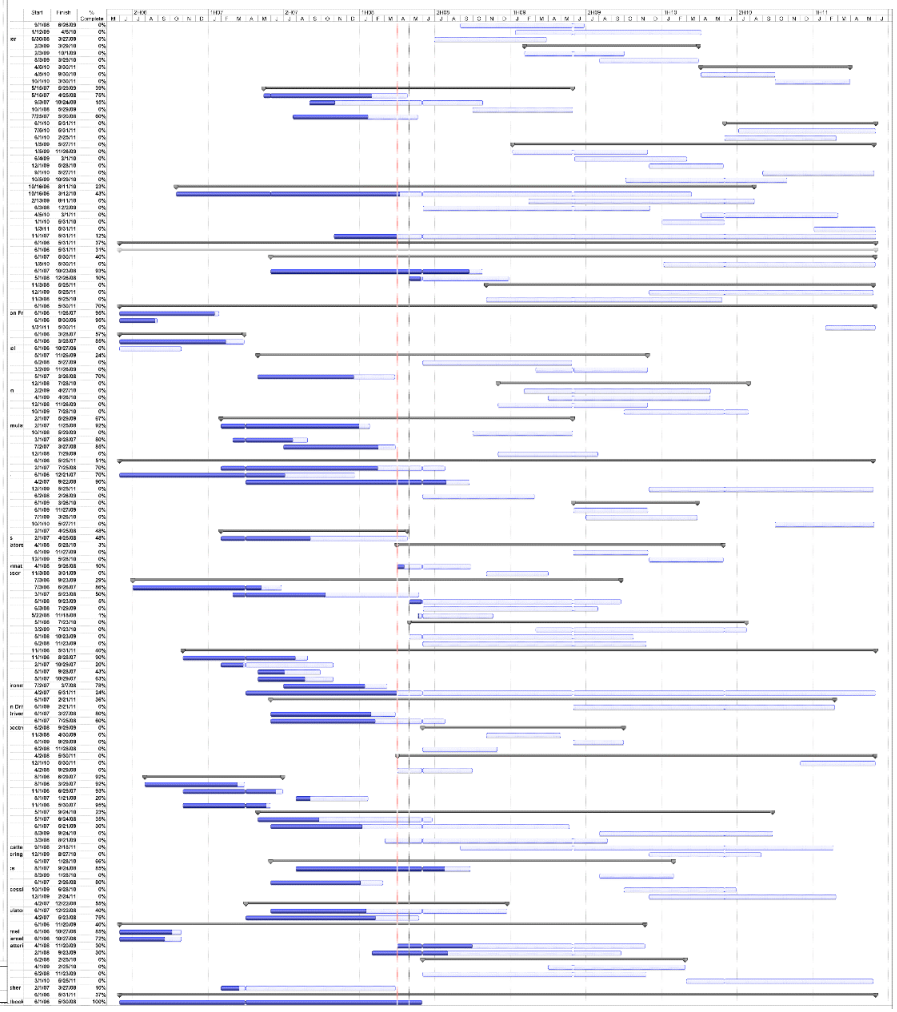
Earned Value Management

OMB Circular A-11 “Major IT investments also must be reported on your agency’s exhibit 53 ...”

- Scope set by subprojects and community.
- Developed detailed task list during design phase
- Estimates are critical!

Project Schedule





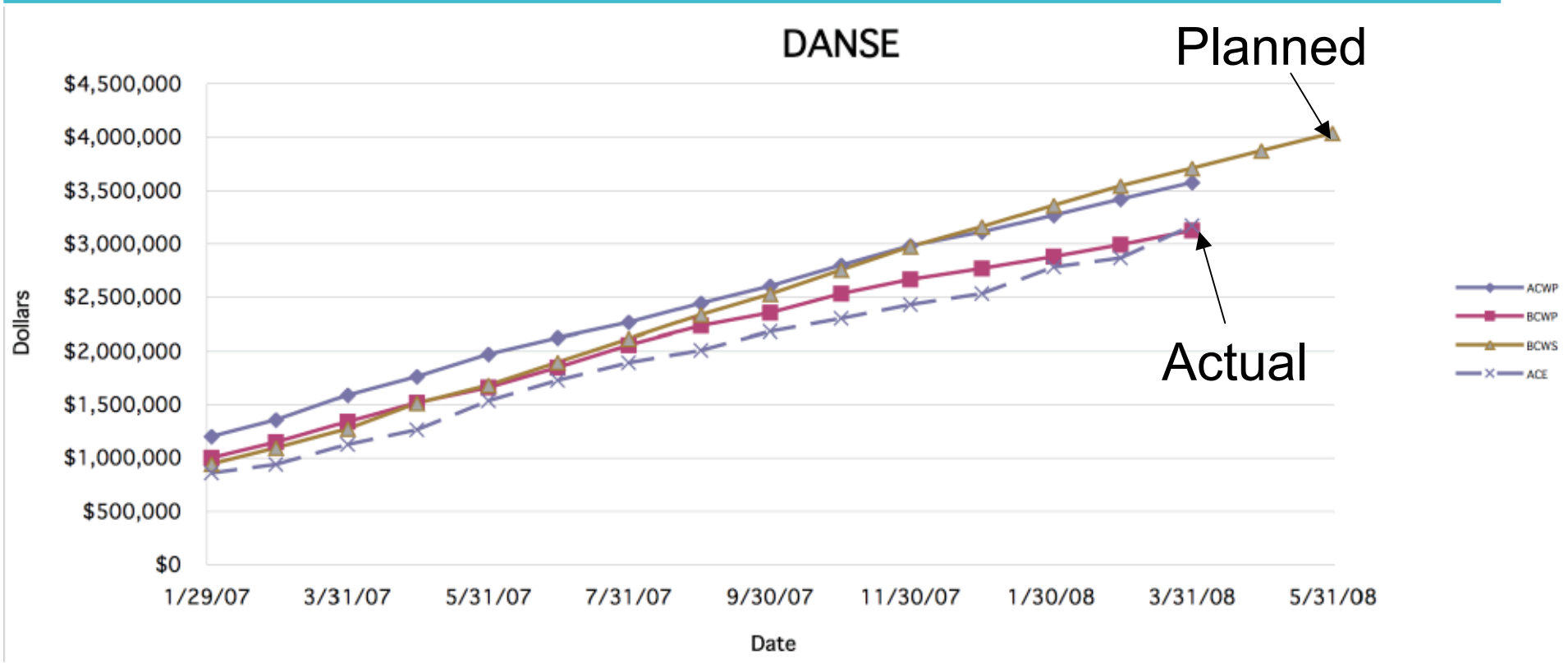
Good Estimates are Critical for Planning

- Prior experience of whole group was used for estimates.
- Eliminate risk -- less proven the approaches are deleted.
- Uncertainties become contingency.
- Original estimates were pretty good! There were changes in tactics, not strategy, and the baseline plan grows out of date.

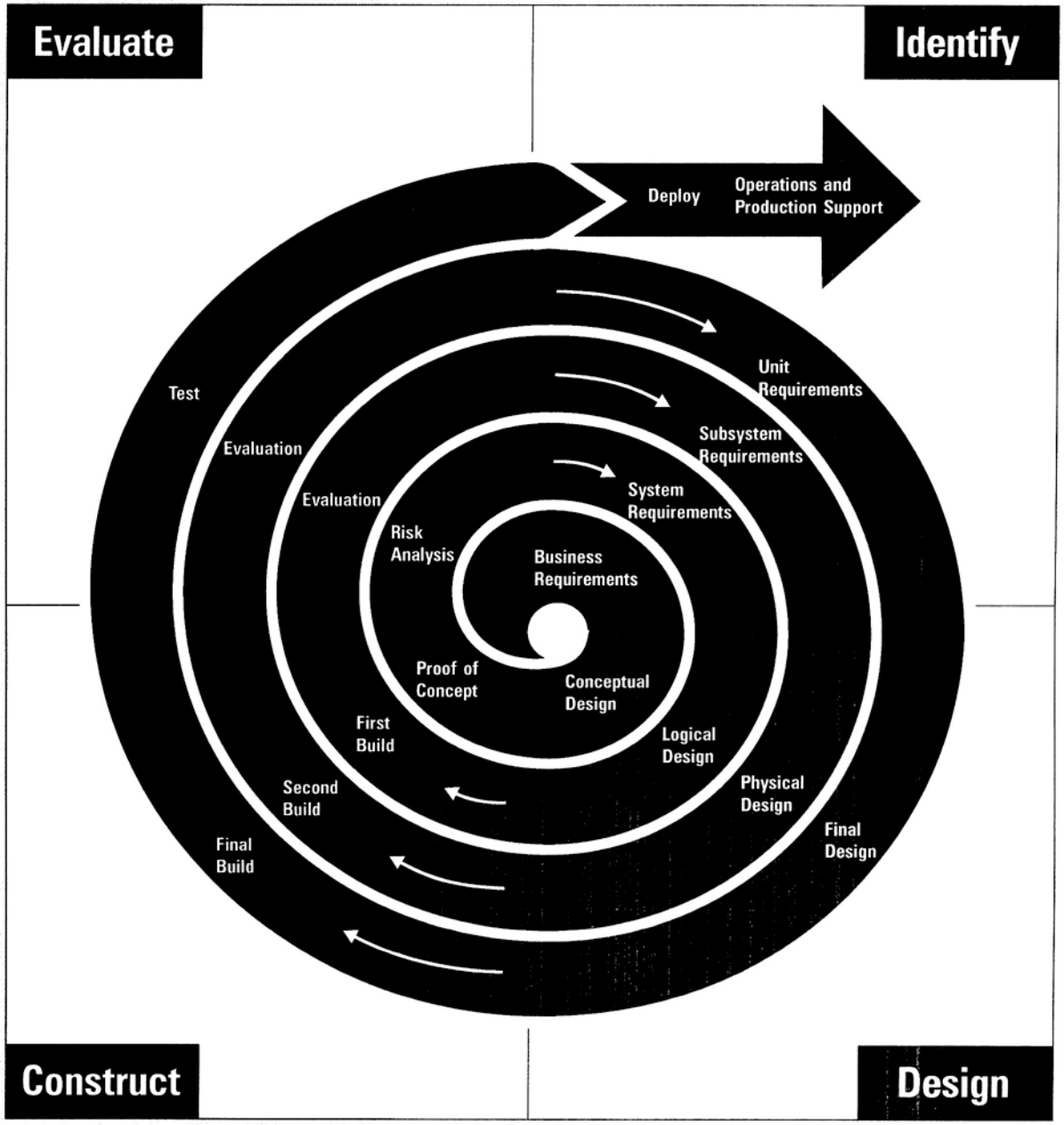
Follow the Project Plan

- P.I. and P.M. believe that this can be done.
- Statement of work tied to 6-month subproject budgets
- Project change requests separated from task changes are separated.
- For first year of tracking against baseline, measured progress ~15% behind.
- Rebaseline activity underway now. Takes more work than hardware rebaselining (e.g. ARCS project).

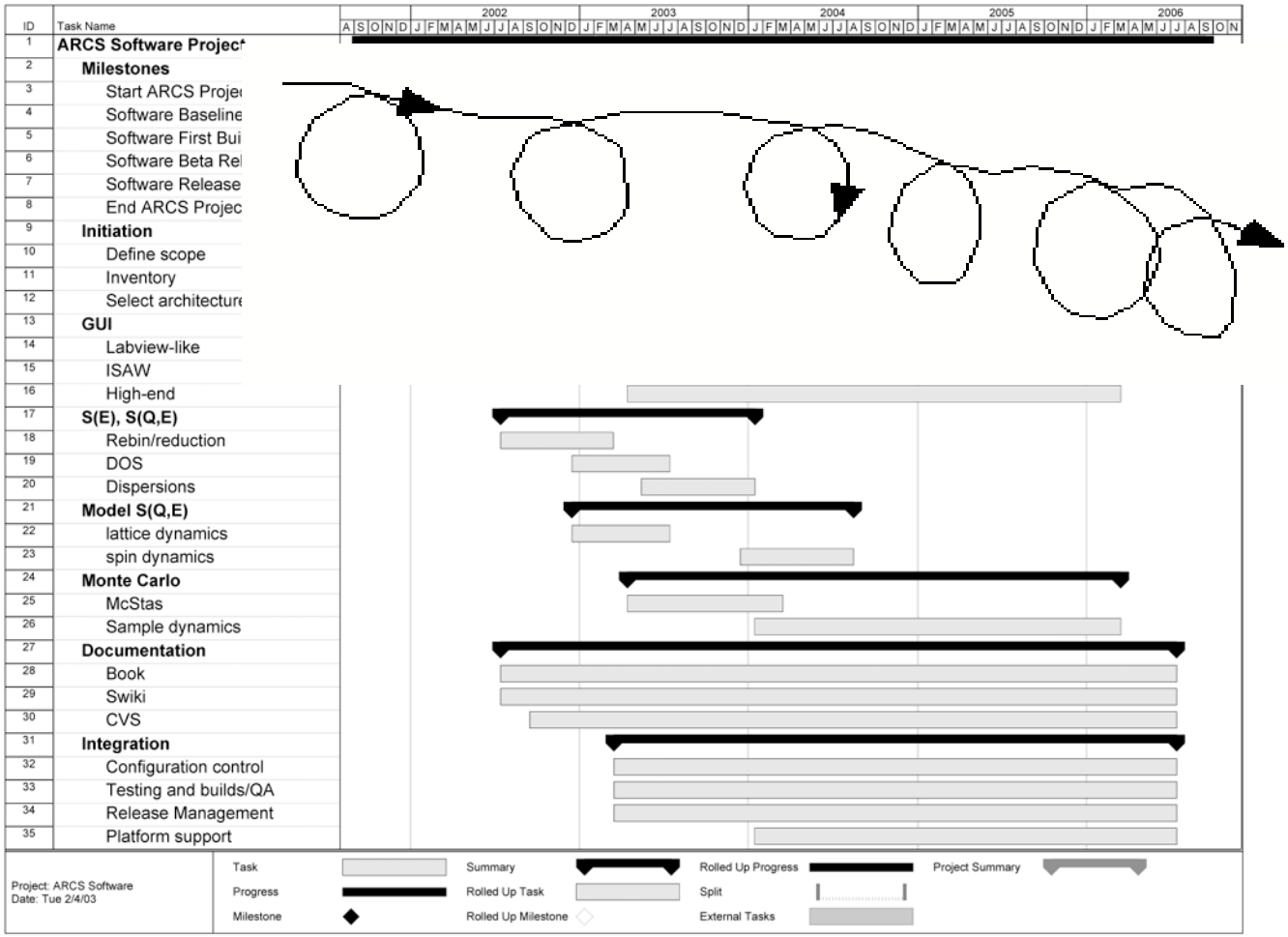
Budget and Earned Value



Software Development



Software Project Schedule

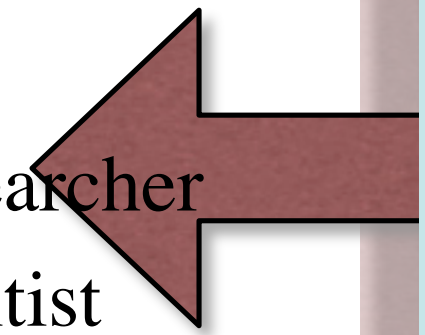


Release

- Subproject leaders showed good taste in planning and executing their release.
- Release includes:
  - Descriptive web pages (labeling of the box)
  - Documentation, especially Users Manual (Developers Manual, too)
  - Installation of software and easy browsing of source code

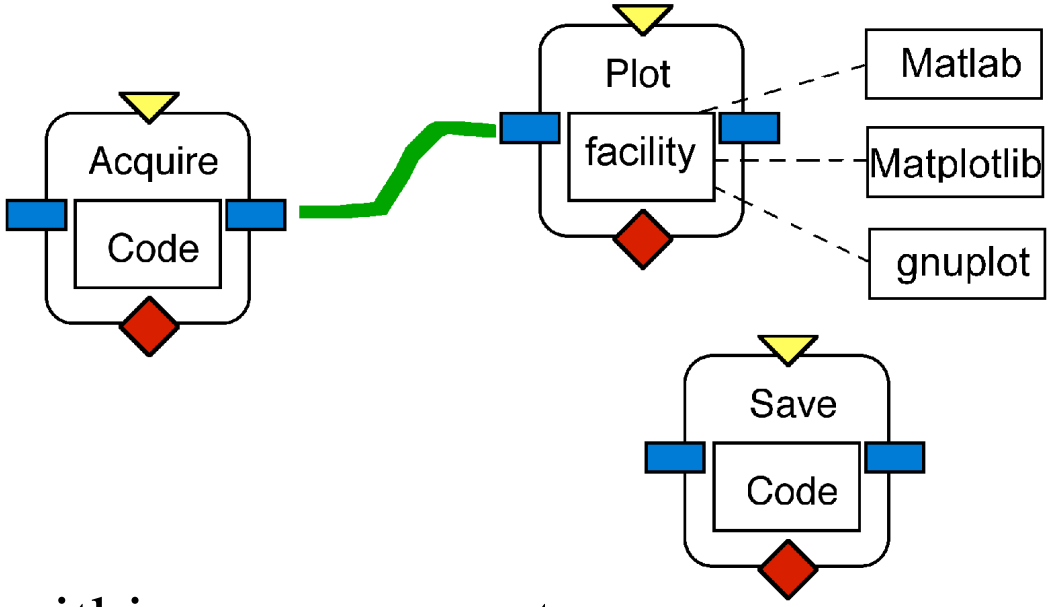
Users Differ in Expertise

1. Beginning Student
2. Senior Student or Postdoc
3. Young Scientist
4. Established Researcher
5. Instrument Scientist
6. Software Developer
7. System Maintainer



REQUIREMENT
One application must serve everybody. Support for different user interfaces

Pyre Runtime Component Framework



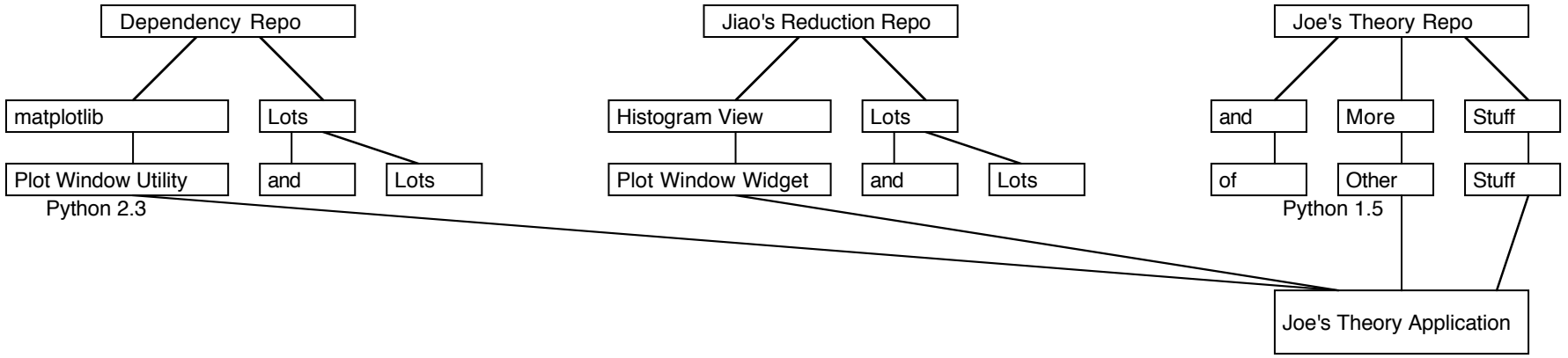
Encapsulate scientific code within a component.

A component inherits useful methods from the framework.

Standardizes interfaces, promotes O-O design.

Pyre is not a full developer toolkit. Some assembly required.

A “Releaser”



The releaser coordinates disparate source trees, applications, and components.

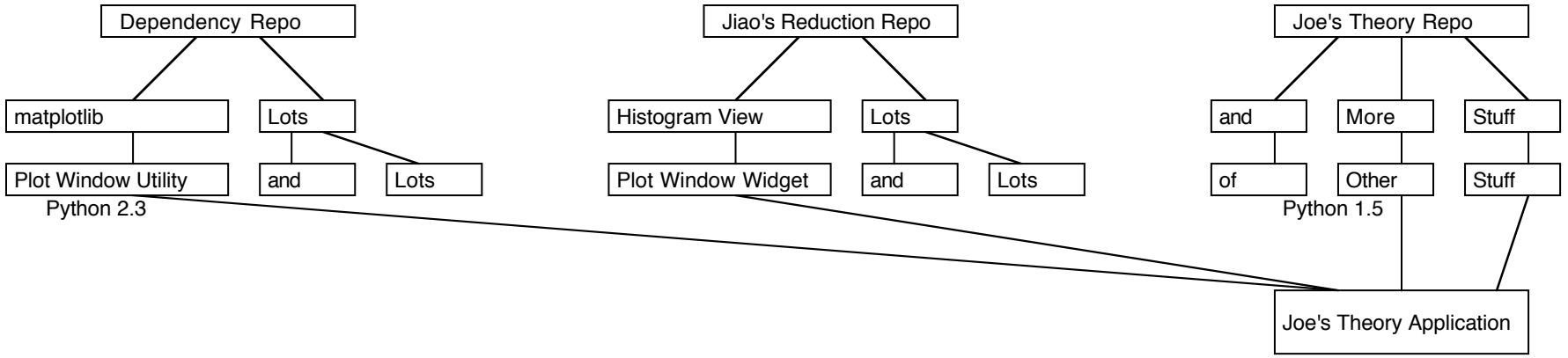
Manages complexity of dependencies.

Aggregates some core functionalities.

Unloads work from developers,

Helps with testing.

A “Build System”



Turns source code into executable files for a specific platform.

All platforms have “personality”.

Best to build on hardware with a controlled file system.

Virtual machines seem the way to go.

How best to integrate with the repositories and releaser?

“Process-Oriented” versus “Product-Oriented” Software Development and Quality Assurance

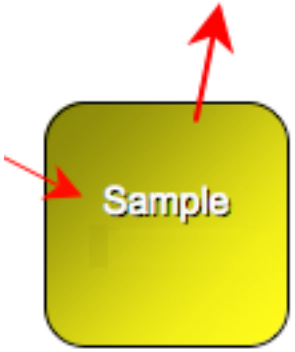
- Process-Oriented (“waterfall process”)
  - gather detailed requirements
  - establish procedures, and document that they are followed
  - success measured by testing against written requirements
  - best when customer and developer have contractual relationship

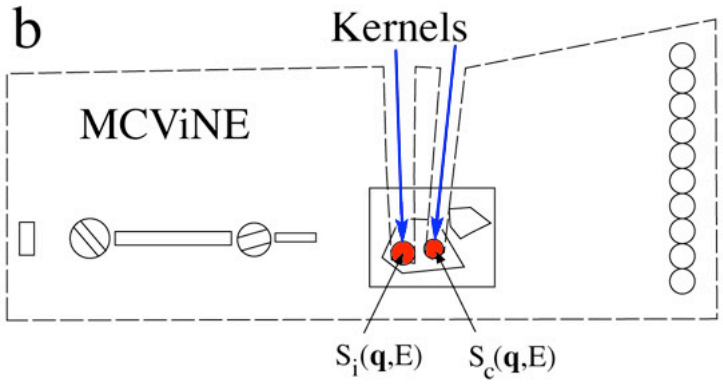
- Product-Orientation (“agile programming”)
  - gather requirements, develop and evaluate prototypes
  - iterative development, get user input on value and robustness, manuals
  - success measured by value to neutron scattering scientists
  - best when developers are customers

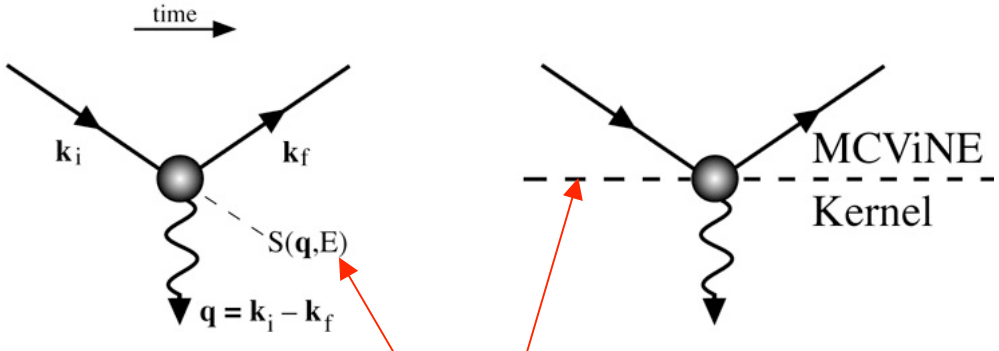
Internal Reviews and Inspections

- Version control system (Subversion, previously cvs)
- Repository inspection is very easy.
  - Please explore “Browse Source” from danse.us/trac/all
  - DANSE management uses this information
  - check-in messages useful, too
- Reviews of Flagship Applications were useful for developers (and managers).
- Collaborative style in Reviews help the developer. Avoid confrontations.

vnf Monte Carlo Instruments + Kernels







Interface is at S(Q,E)

P.I. Strategy for Architectural Integration

1. Rebaseline around Flagship Applications

2. Project-wide refactoring of code towards component design. Selling points will be vnf, PARK, web services, and gui generation tools.

3. Web services for very friendly users (late 2008)
   - Focus on cross-project interoperability through vnf.
   - A toolkit for scientific creativity.
   - Emphasis first on working prototype

Science Interoperability
Diffraction ∪ Inelastic

With software components, what will scientists build?

- depends on specific science needs
- depends on creativity

In 2008, integrate tools across materials physics with diffraction and inelastic scattering.

Science case of anharmonic phonons in Ni_3Fe nanocrystals.

Trips to the SNS

- Subgroup visits to the SNS (WBS 7 May 29,30, WBS10 June 9-12, WBS 8 in August)

  - discuss science

  - awareness of software needs by instrument scientists

  - expectations and timescales are important to synchronize

After DANSE Construction Ends in 2 Years

- “All that is human must retrograde if it does not advance.” E. Gibbon.

- Three big issues:
  - Maintenance of DANSE
  - User Support
  - Future Upgrades

Maintenance (Sine Qua Non!)

- People
  - Software engineers/developers
  - Probably best if co-located
- Technology Evolves
  - Operating Systems
  - Grids
  - Web Services
  - Frameworks
- Resources
  - Nobody maintains software as an enjoyable hobby.
  - Neutron facilities can help. How much?
  - Central Simulation Facility? Who pays?
  - Under another organization that has a science mission, e.g., ASISI?

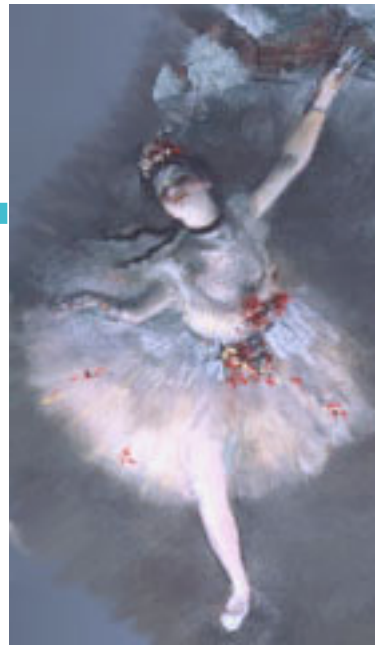
User Support (Increases User Acceptance)

- People
  - Engineers: Can I install SliceView on my PDP-8?
  - Scientists: Why can't I use a 500 eV plane wave cutoff for FeS? (Will this demand excessive memory if I use a fine k-space mesh?)
  - Physically dispersed group is probably okay.
- Technology Evolves
  - Web services
  - Grid questions
  - Help with new computing facilities
- Resources
  - Nobody does user support as an enjoyable hobby.
  - Is some of this more of a scientific collaboration than a service call?
  - Neutron facilities have had some success with this. Plans for more?
  - Far beyond present ability of NSSA or SHUG.

Upgrades of DANSE (versus Retrograde)

- People
  - Engineers: New systems for high performance computing.
  - Scientists: New computational science of neutrons and materials
  - Physically dispersed group is okay.
- Technology Evolves
  - Multicore systems.
  - Web services and Grids
  - Where to with framework integration?
- Resources
  - This could be part of a science research effort.
  - We need to assess the holes in the scientific coverage of DANSE.
  - Unfunded community efforts move slowly.
  - ASISI?

Summary



- Getting the right people is difficult.
- DANSE Team has been together for 6 years now.
- Shared scientific vision is stable.
- Base projections on demonstrated productivity.
- Architectural coherence is difficult for isolated development efforts.
- Web services are an integration strategy
  - add tools to vnf
  - no installation issues for users and later developers
- Plan beyond Construction -- uncharted waters

End.