



Management of Change

Mike McKerns, Caltech

- Project Status
 - Schedule, Cost, and Earned Value
 - Issues and a Path Forward
- Project Management Infrastructure
 - Agility of the WBS
 - Software Quality and Metrics of Completion
 - Integrated Development and Management

Project Growth Profile

Please inquire to Mike McKerns
or Brent Fultz for access to
information contained on this slide.

- year 1: infrastructure and training
- year 2: initial DANSE releases
- year 3: project-wide integration
- year 4-5: project robustness

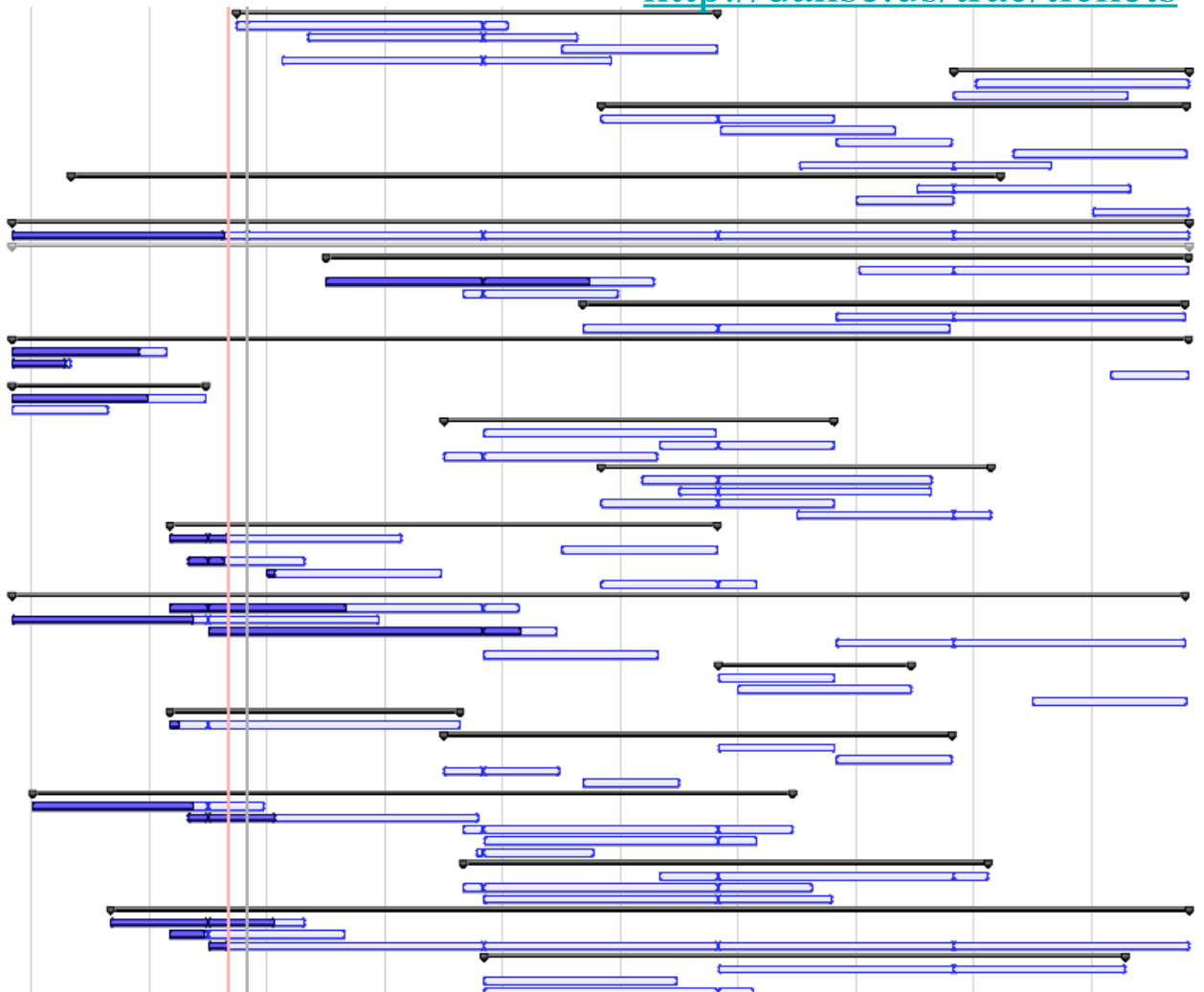
Project Organization Reflected in WBS

WBS 2: Project Infrastructure	McKerns, Fultz
WBS 3: Software Quality	McKerns, Aivazis
WBS 4: Central Services	Aivazis
WBS 5: Common Algorithms	*
WBS 6: Diffraction	Billinge
WBS 7: Eng. Diffraction	Ustundag
WBS 8: SANS	Butler
WBS 9: Reflectometry	Kienzle
WBS 10: Inelastic Scattering	Fultz
WBS 11: Education & Outreach	*

Five Year Software Construction Plan

<http://danse.us/trac/tickets>

50	8.2	Reflections 1D Modeling	5/15/07	5/29/09	0%
51	8.2.1.1	1D Models	5/15/07	7/8/09	0%
55	8.2.2.1	Basic 1D Analysis Launcher	8/5/07	10/24/08	0%
58	8.2.2.2	Extended 1D Analysis Launcher	10/1/06	5/29/09	0%
60	8.2.3	1D Analysis UI	7/25/07	12/16/08	0%
63	8.3.0	Experiment Planning	8/1/10	5/31/11	0%
64	8.3.0.1	Experiment Planning UI	7/5/10	5/31/11	0%
66	8.3.0.2	Experiment Planning Launcher	8/1/10	2/25/11	0%
68	8.3.1	Reflections 3D Modeling	12/1/06	5/27/11	0%
69	8.3.1.1	3D Models	12/1/06	11/26/09	0%
72	8.3.1.2	3D Model Builder	8/4/06	2/1/10	0%
74	8.3.1.3	Reflections Scattering Kernel	12/1/06	5/29/10	0%
76	8.3.1.4	3D Analysis UI	8/1/10	5/27/11	0%
78	8.3.1.5	3D Analysis Launcher	10/5/09	10/29/10	0%
81	8.3.2	CHI-Specular Reflectometry	8/1/06	8/11/10	4%
94	8.3.3	Instrument Effects	4/5/10	3/31/11	0%
97	8.3.4	Bindings to ODMIP	1/1/10	5/31/10	0%
99	8.3.5	Bindings to GROMACS	1/5/11	5/31/11	0%
101	8.4	REFL Resources	8/1/06	5/31/11	18%
120	11.3.2	REFL Minority Student Research	8/1/06	5/31/11	16%
8	10	INS	8/1/06	5/31/11	16%
1	5.1.1	Basic Data Reduction	10/1/07	5/30/11	32%
2	5.1.1.0	Single Crystal Data Reduction	1/5/10	5/30/11	0%
5	5.1.1.1	Polycrystal Data Reduction	10/1/07	2/20/09	80%
8	5.1.1.2	Measurement	5/1/06	12/26/08	0%
11	5.1.2	Advanced Data Reduction	11/3/06	5/25/11	0%
12	5.1.2.1	Advanced Data Reduction	12/1/06	5/25/11	0%
15	5.1.2.2	Reduction Corrections	11/3/06	5/25/10	0%
16	5.3.1	Instrument Simulation	8/1/06	5/30/11	62%
19	5.3.1.1	Monte-Carlo Instrument Simulation Framework	8/1/06	1/28/07	82%
21	5.3.1.2	Bindings to McStas	8/1/06	8/30/06	91%
23	5.3.1.4	Bindings to VITESS	5/31/11	5/30/11	0%
25	5.3.2	Sample Simulation	8/1/06	3/28/07	47%
26	5.3.2.1	Sample Simulation Framework	8/1/06	3/28/07	70%
28	5.3.2.2	Coherent Elastic Scattering Kernel	8/1/06	10/27/06	0%
30	5.3.3	Solid-State Simulation	4/1/06	11/26/09	0%
31	5.3.3.0	Solid-State Materials Simulation	8/2/06	5/27/09	0%
33	5.3.3.1	Bindings to Abinit	3/2/06	11/26/09	0%
38	5.3.3.2	Bindings to VASP	4/1/06	2/25/09	0%
39	5.3.4	Harmon-Fock Simulations	12/1/06	7/28/10	0%
40	5.3.4.0	Harmon-Fock Materials Simulation	2/2/06	4/27/10	0%
43	5.3.4.1	Bindings to GANESS	4/1/06	4/26/10	0%
48	5.3.4.2	Bindings to NWChem	12/1/06	11/26/09	0%
49	5.3.4.3	Bindings to FIREBALL	10/1/06	7/28/10	0%
52	5.3.5	Molecular Dynamics Simulations	2/1/07	5/29/09	15%
53	5.3.5.0	Molecular Dynamics Materials Simulation	2/1/07	1/25/08	26%
58	5.3.5.1	Bindings to NAMD	10/1/06	5/29/09	0%
58	5.3.5.4	Bindings to MMTK	3/1/07	8/28/07	50%
61	5.3.5.5	Bindings to GULP	7/2/07	3/27/08	0%
83	5.3.6	Scattered Intensity	12/1/06	7/29/09	0%
86	5.4.1	Data Structures	8/1/06	5/25/11	42%
87	5.4.1.1	Reduction Data Structures	2/1/07	7/25/08	90%
71	5.4.1.2	Experiment Metadata Containers	8/1/06	12/21/07	50%
74	5.4.1.3	Common Array Manipulations	4/2/07	8/23/08	90%
77	5.4.1.4	Advanced Array Manipulations	12/1/06	8/23/08	90%
80	5.4.1.8	Database Accessor	8/2/06	2/26/09	0%
82	5.4.3	Basic Numerical Transformations	8/1/06	3/26/10	0%
83	5.4.3.0	Numerical Libraries	8/1/06	11/27/09	0%
85	5.4.3.2	Common Scattering Functions	7/1/06	3/26/10	0%
87	5.4.4	Error Propagation Algorithms	10/1/10	5/27/11	0%
88	5.4.5	Crystallography	2/1/07	4/25/08	3%
90	5.4.5.6	Crystal Structure Container Class	2/1/07	4/25/08	3%
93	5.4.6	Molecular Viewers and Format Translators	4/1/06	5/28/10	0%
94	5.4.6.2	Bindings to Wolden	8/1/09	11/27/09	0%
96	5.4.6.6	Bindings to VMD	12/1/06	5/28/10	0%
98	5.4.6.7	Molecular & Crystal Structure Format Translator	4/1/06	9/26/08	0%
101	5.4.6.8	Crystallographic Database Accessor	11/3/06	3/10/09	0%
103	5.4.7	Analysis Support Applications	7/3/06	8/23/09	26%
104	5.4.7.1	Histogram Viewer	7/3/06	8/26/07	70%
107	5.4.7.2	Instrument Builder	2/1/07	5/23/08	30%
110	5.4.7.3	Sample Builder	5/1/06	8/23/09	0%
114	5.4.7.4	Structures Builder	8/3/06	7/29/09	0%
117	5.4.7.5	Molecular Format Translator	5/22/06	11/16/08	0%
120	5.4.8	Sample Materials Applications	5/1/06	7/23/10	0%
121	5.4.8.1	Materials Simulation Launcher	3/2/06	7/23/10	0%
125	5.4.8.2	Sample Dynamics Launcher	5/1/06	10/23/09	0%
129	5.4.8.3	Structure Dynamics Launcher	8/2/06	11/23/09	0%
132	5.4.9	Maintenance & Updates	11/1/06	5/31/11	16%
133	5.5.0.a	Extensions to Dietdile	11/1/06	8/28/07	88%
138	5.5.0.b	Geometric Shapes	2/1/07	10/28/07	20%
139	5.5.0.c	Maintenance & Updates	4/2/07	5/31/11	2%
144	10.1.0	INS Data Reduction	8/2/06	2/21/11	0%
145	10.1.0.0	Single Crystal Inelastic Reduction Drivers	8/1/06	3/21/11	0%
148	10.1.0.1	Polycrystal Inelastic Reduction Drivers	8/2/06	3/27/09	0%



WBS & Earned Value Reporting

- half-year subproject funding increments tied to a SOW
- SOW built from a half-year slice of the WBS
- submit (contractual) monthly reports on scheduled WBS tasks
- earned value metrics derived from reported cost and activity

WBS	Task	Type	Account	Year / Month	Direct Costs	Indirect Costs
7.2.1.1	Modular integration of finite element analysis (FEA) software – AI	LA	ENGDIF	2007/04	\$ 1,402	\$ 423
7.2.1.2	Input/output parameters for FEA	LA	ENGDIF	2007/04	\$ 1,402	\$ 423
7.2.2	Self-Consistent Modeling	LA	ENGDIF	2007/04	\$ 1,261	\$ 270
7.2.3	Inverse Problem Analysis	LA	ENGDIF	2007/04	\$ 1,261	\$ 270
7.3.2	Microstructure Simulation	LA	ENGDIF	2007/04	\$ 2,200	\$ 467
7.3.X	Maintenance and Updates	LA	ENGDIF	2007/04	\$ 961	\$ 254
7.4.1	Local administrative support	LA	ENGDIF	2007/04		
7.4.2	Computing hardware and software	MS	ENGDIF	2007/04	\$ 147	\$ 69
7.4.3	Travel	TR	ENGDIF	2007/04		
7.4.4	Communications	MS	ENGDIF	2007/04		
7.4.X	Subawards	SA	ENGDIF	2007/04		
11.3.5	Minority Student Research Position	LA	ENGDIF	2007/04		

Budget Period: Jun 1, 2006 to May 31, 2007 (NCE)
Reporting Period: Apr 1, 2007 to Apr 30, 2007

TOTALS \$ 8,634 \$ 2,176

WBS	Task	Account	Baseline Start Date	Baseline End Date	Projected % Complete at End y2p1	Prior Reported % Complete	Actual % Complete
7.2.1.1	Modular integration of finite element analysis (FEA) software – AI	ENGDIF	09/02/10	10/28/10	100%	69%	70%
7.2.1.2	Input/output parameters for FEA	ENGDIF	10/02/10	12/13/10	100%	65%	68%
7.2.2	Self-Consistent Modeling	ENGDIF	09/22/10	04/26/12	70%	29%	32%
7.2.3	Inverse Problem Analysis	ENGDIF	12/31/10	04/10/12	65%	19%	21%
7.3.2	Microstructure Simulation	ENGDIF	05/02/11	02/23/14	18%	0%	7%

Budget Period: Jun 1, 2006 to May 31, 2007 (NCE)
Reporting Period: Apr 1, 2007 to Apr 30, 2007

Project Status (year 1)

Please inquire to Mike McKerns
or Brent Fultz for access to
information contained on this slide.

Project Status (year 2)

Please inquire to Mike McKerns
or Brent Fultz for access to
information contained on this slide.

Schedule Lag: Issues

Please inquire to Mike McKerns
or Brent Fultz for access to
information contained on this slide.

Schedule Lag: Issue Resolution

- Current issues are “one-time” instead of “ongoing”
- Good hires to make progress on key software infrastructure
 - runtime component framework
 - global optimization framework
 - Monte Carlo instrument simulation framework
 - UI generator and web services
- Rebaseline the WBS
 - no reduction of scope
 - reschedule around flagship applications and release cycle
 - account for work recently done on tasks scheduled in out years

Begin with High-level Definition & Design

- Ensure identification of all tasks
 - list science objectives and workflow requirements at the subgroup-level
 - assemble science and workflow into target release products (deliverables)
 - each deliverable must be associated with a task
 - identify if task is clearly states software and science requirements
 - identify if software has the desired interfaces
- Ensure a clear definition of all deliverables & interfaces
 - identify all of the project deliverables (especially the flagship applications)
 - break deliverables into well-defined conceptual components
 - specify information to exchange between the components
 - draw UML diagram of components and data objects for each deliverable

Capture Vision in WBS Dictionary

5.3.3.0 Solid-State Materials Simulations: *Ab initio solid-state calculations of electronic structure are practical on systems with 100 atoms or so in a periodic unit cell. These quantum mechanical calculations provide electron wavefunctions within the local density approximation and its extensions. At minimum, we will support calculations of optimized atom positions, electron densities, densities of state, dynamical matrices, and total energy and forces.*

5.3.3.0.x API Implemented as Abstract Classes

5.3.3.0.x Materials Simulation Data Objects

5.3.3.0.x Result Analysis and Steering

5.3.3.1-4 Bindings to Common Solid-State Materials Simulation Packages: *We will provide bindings to Abinit, VASP, CPMD, and WIEN2K.*

5.3.3.n.x Input/Output File Parsers

5.3.3.n.x Full Simulation Driver

5.3.3.n.x Replace Parsers w/ Data Object Bindings

5.3.3.n.x Replace Full Simulation Driver w/ Drivers for each Step?

5.3.4.0 Hartree-Fock Materials Simulations: *Hartree-Fock codes with Gaussian basis sets are also practical for systems of 100 atoms or so. These codes are quite mature, and can provide both the positional information required for diffraction studies and the vibrational eigenfrequencies required for inelastic scattering. At a minimum, we will support calculations of optimized atom positions, electron densities, densities of state, dynamical matrices, and total energy and forces.*

5.3.4.0.x API Implemented as Abstract Classes

5.3.4.0.x Materials Simulation Data Objects

5.3.4.0.x Result Analysis and Steering

5.3.4.1-3 Bindings to Common Hartree-Fock Materials Simulation Packages: *We will provide bindings to GAMESS, NWChem, and FIREBALL.*

5.3.4.n.x Input/Output File Parsers

5.3.4.n.x Full Simulation Driver

Michael McKerns 1/16/07 11:25 AM

Comment: At minimum, we will deliver components that launch a full simulation through the bindings, and an API for these calls. At maximum, we will deliver bindings to individual simulation steps, with the launching and configure done in python... and includes a UI and some results analysis. Also, convenience functions.

Michael McKerns 1/18/07 2:22 PM

Comment: Group into: 1) SCF calculation (total energy, electron density, forces), 2) geometry optimization, 3) phonon properties (density of states, dynamical matrix). Should calculation of phonon properties should be in it's own separate task? ... I think so. Should a calculation of G(R) or G(R,t) be available here or as a separate task?

Michael McKerns 1/16/07 4:06 PM

Comment: Correct/complete? This list depends on if we choose the max or min implementation of this task. What specific API are needed? What about handling pseudopotentials?

Michael McKerns 1/16/07 11:20 AM

Comment: Executable configure and launch.

Michael McKerns 1/16/07 4:02 PM

Comment: Depends on min or max implementation, as in 5.3.3.0. Max implementation (drivers at each step) is not likely; it's be most useful for geometry optimization, but probably not worth the effort.

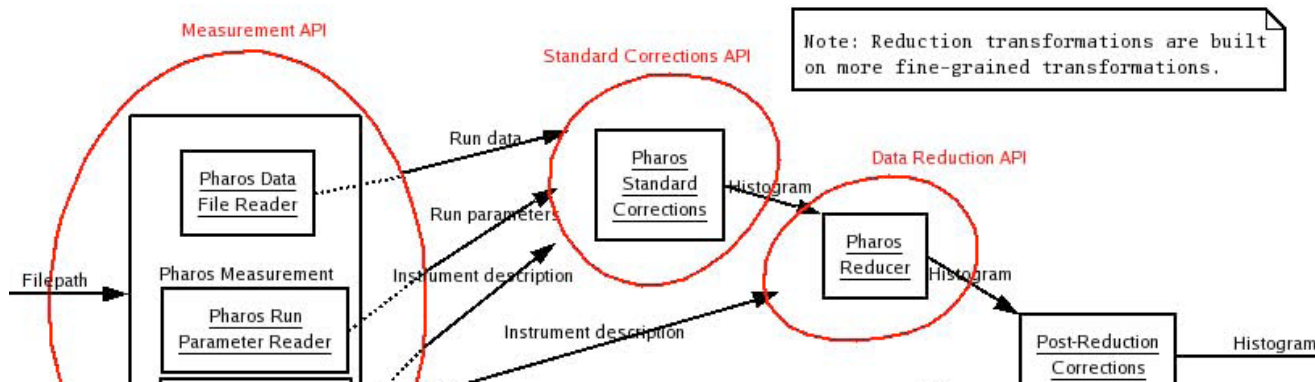
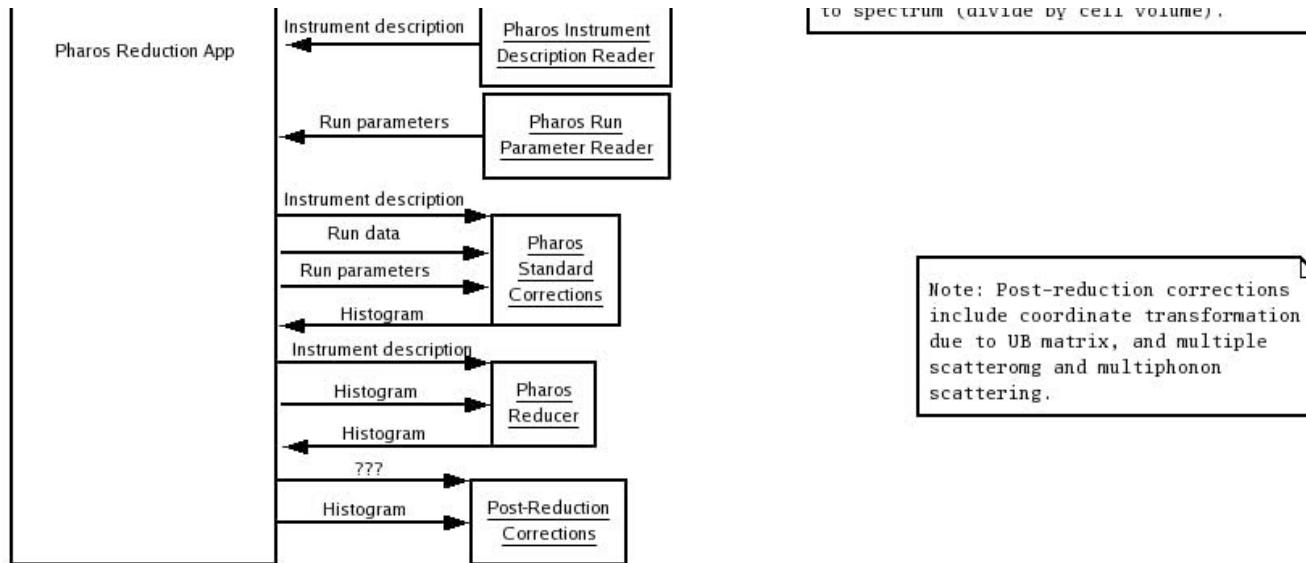
Michael McKerns 1/16/07 11:25 AM

Comment: Same as 5.3.3.0.

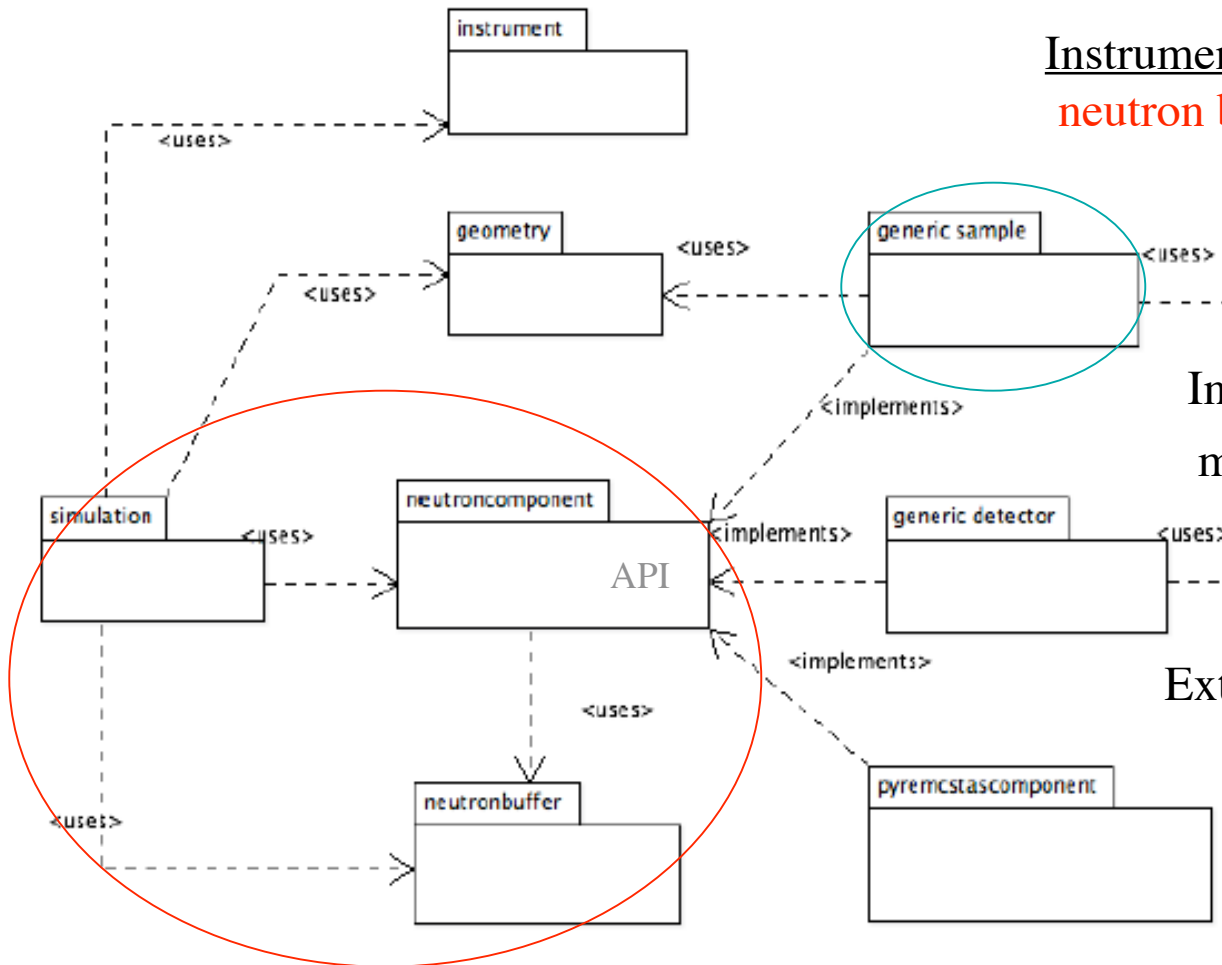
Michael McKerns 1/16/07 11:28 AM

Comment: Same as 5.3.3.0.

Capture High-level Specifications



Use Good Software Engineering Design



Instrument simulation is **simulation** engine, **neutron buffer**, and **neutron components**... and uses instrument and geometry

Initially, McStas components provide monitors, guides, samples, detectors

Extend by building neutron component with **generic sample** construction

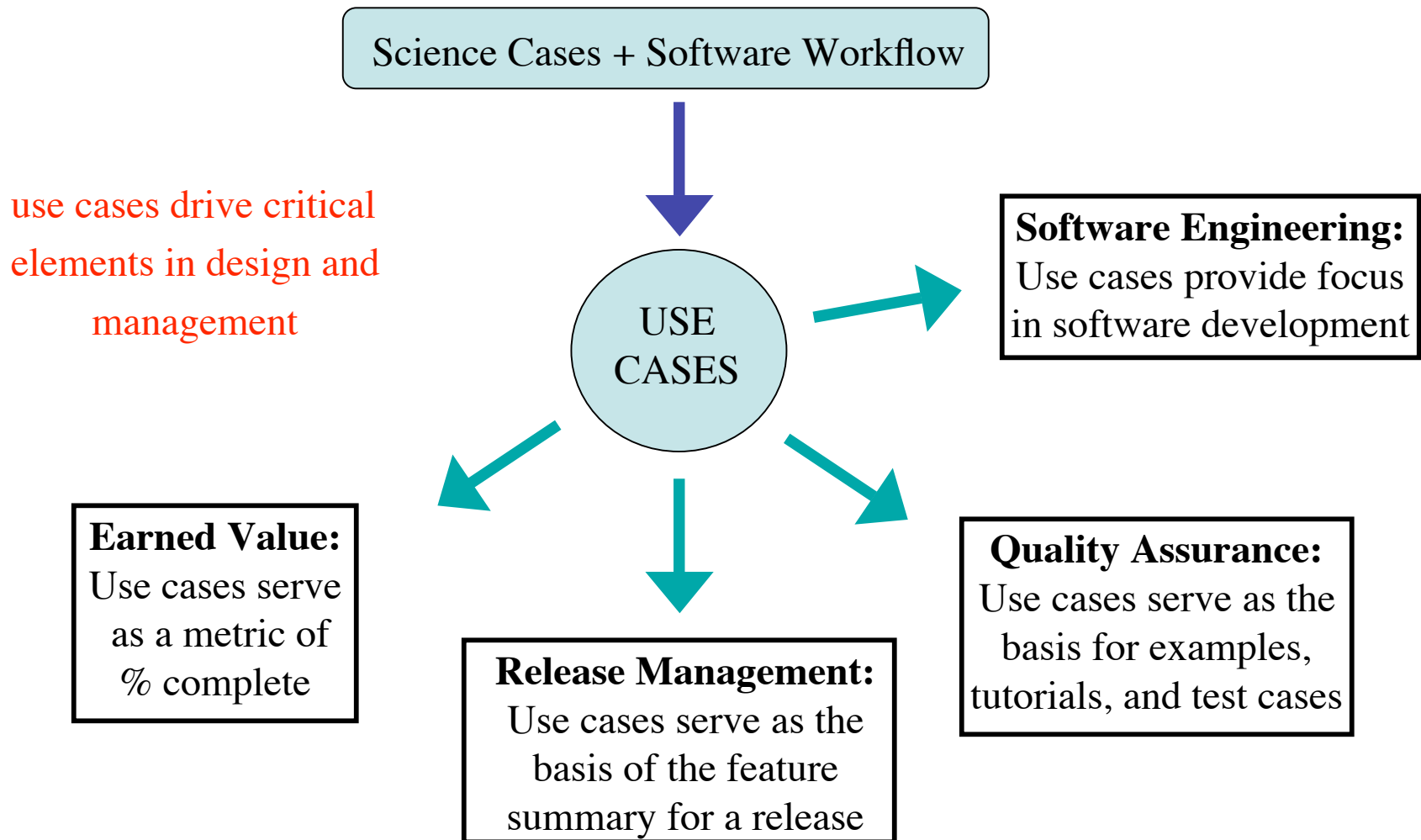
Capture Software Engineering in the WBS

- Building tasks with narrower scope
 - creates additional tasks (but not scope)
 - tasks tend to be shorter duration with better-defined scope
 - code tends to be more reusable
 - development is more efficient and there is less rework
 - creates less surprises
- Schedule tasks once WBS dictionary is updated
 - developers focus on more than one task
 - more tasks run simultaneously, thus durations can be lengthened
 - more akin to development in a small dynamic group
 - less well-defined tasks must be scheduled later in the project

Outlook for Updating the WBS

- Current WBS is good, but not complete enough...
 - Needs to capture flagship applications and higher-level tasks
- **“Functionality” is stable, but requirements are dynamic**
 - Science requirements are well understood
 - Software requirements are still being defined
- Path Forward:
 - We have science objectives well in hand (i.e. science scope is stable)
 - We need to schedule the WBS to facilitate planned project-wide integration
 - We need to better organize the WBS around release products
 - We need to add more emphasis on software workflow to the WBS

Centralized Role of Use Cases



Software Quality & Metrics of Completion

- Estimates of complete based on:
 - completion of science use cases
 - completion of software use cases
 - verification by a collection of testable data objects
 - status of accompanying documentation
- Completion based on review of release products:
 - released code (i.e. applications, libraries, modules)
 - user's guide and developer's guide
 - release website

Metrics and the Review Process

Top three risks:	See Requirement document: <ul style="list-style-type: none">- Speed- Parameter access- Use of models from other modules
Top three open issues:	Deployment strategy (to be discussed with central services)

General Observations

Overall, there is a lot of good content. Task status is clear and on schedule. The high-level architecture diagrams look solid. A more detailed high-level behavior diagram for the in-scope application would help clarify the system boundary and may also better define the API. However, there are a few critical elaboration details missing; thus, several action items have been created, and must be closed before the elaboration phase is declared complete. Data structures have been identified as "model objects"; more details will be needed in the construction phase.

Actions

ID	Action Item	Assigned To	Due By
1	Inform reviewers/testers of their roles (http://danse.us/trac/tickets/ticket/175)	Butler	05/07/2007
2	List in-scope 1D models (http://danse.us/trac/tickets/ticket/176)	Butler	05/07/2007
3	Provide evidence of prototype application (http://danse.us/trac/tickets/ticket/177)	Butler	05/07/2007
4	Generate inheritance diagram for each in-scope component (http://danse.us/trac/tickets/ticket/178)	Butler	05/07/2007
5	Capture behavior specifications for each in-scope	Butler	05/07/2007

Action items are a result of a review or audit. Reviews are closed when all action items are closed.

Action items can be treated as tickets in the online bug and issue tracking system.

Submitting a Ticket

<http://danse.us/trac/all/query>

{1} Active Tickets (70 matches)

- List all active tickets by priority.
- Color each row based on priority.
- If a ticket has been accepted, a "*" is appended after the owner's name

Ticket	Summary	Component	Version	Milestone	Type	Owner
#182	Complete description of scripting task on wiki	admin		4.3.1:inception	task	aivazi
#172	clarify software production requirements	admin			task	mmck
#170	Non-functional requirements need clarification.	boxmin		5.4.2.1:inception	defect	kienzl
#169	Identification of acceptance criteria	boxmin		5.4.2.1:inception	defect	kienzl
#168	Identify task timeline	boxmin				
#167	Identify risks and risk mitigation plan.	boxmin				
#166	Identify the system boundary, actors, and interfaces within each application.	boxmin				
#165	Provide a clear definition of task scope.	boxmin				
#164	Name target testers	boxmin				
#163	Typo in pyre file FileLockingNT.py	Devel				
#162	Crystal Structure Class	simul				
#161	multiple histograms with same name cause trouble	histog				
#159	histogram review #3 action item	histog				
#158	Action items for Histogram Design Review # 2	histog				
#157	histogram design review #1 action items	histog				
#156	Complete 9.1.1 "Data Transformations"	reflec				
#155	Complete 7.2.3 "Inverse Problem Analysis"	engdi				
#154	Complete 6.2.3 "Model Independent Peak Fitting"	diffra				
#153	Complete 5.3.1.2 "McStas"	mcsta				
#152	Complete 3.2.2 "Build System"	mk				
#149	supplementary specifications for reduction package	reduc				
#148	use cases for reduction package	reduc				
#147	risk mitigation plan for reduction package	reduc				
#146	roles and drivers for reduction package	reduc				
#145	clarify requirements for reduction package	reduc				
#144	define scope for reduction package	reduc				
#141	identify roles and drivers for instrument apps	instru				
#138	clarify requirements for measurement	meas				
#137	Define use cases for measurement	meas				
#135	Complete 5.?.?.? "Geometry"	geom				
#134	Complete 5.?.?.? "Reduction"	reduc				
#133	Complete 5.?.?.? "Instrument"	instru				
#132	Complete 5.?.?.? "Measurement"	meas				
#131	Complete 5.?.?.? "Histogram"	histog				
#130	suggestions for INS reduction plotting	reduc				

Create New Ticket

Your email or username:

Short summary:

Type:

Full description (you may use WikiFormatting here):

Ticket #57 (enhancement)

Reduction -- Working with funky detectors.

Status: new

Reported by:	max	Assigned to:	
Priority:	major	Milestone:	
Component:	reduction	Version:	
Keywords:	reduction	Cc:	

In no particular order:

- Generally, we use the vanadium data and some knowledge of pressure of the detectors to try and correct for their not all possibly add to this the shading of some of the detectors from that's what does it?)
- Currently, it's very difficult to pick bad detectors to be removed. In fact, it's often difficult to know which detectors are bad until therefore propose the following:

Once the data has been reduced, you open an intensity plot of s phi space that appears to have been marred by a bad detector. detectors, which may then be displayed to the user -- possibly i (six to f) (dot to f) and (dot six). He may then pick the culprit

Integrated Change Tracking System

<http://danse.us/trac/all/timeline>

an integrated environment benefits developers and managers

The screenshot displays two browser windows from the Trac system. The left window shows the 'Timeline - All Develop' page, which lists a series of tickets with their dates, times, authors, and statuses. The right window shows a 'diff' view for the file 'HowToCreateAReleaser', comparing version 2 and version 3. The diff view includes a table of line numbers and a list of changes, such as adding a new section for 'release.py' and modifying 'deps.py'.

Timeline

05/22/08:

- 13:10 Ticket #361 (defect) closed by juhas (in fixed: Fixed in r2097.
- 11:59 Ticket #299 (defect) closed by juhas (in fixed: Done in r2096.
- 07:22 Ticket #8 (task) closed by sylee (in FEA Invalid: will be added in v.0.5

05/20/08:

- 23:14 Ticket #361 (defect) created by juhas (i pdfgul project becomes invalid for huge occupancy
- 13:04 Ticket #78 (defect) closed by youngshin

Changes between Version 2 and Version 3 of HowToCreateAReleaser

Author: linjiao (IP: 131.215.220.161)
Timestamp: 05/22/08 11:46:53 (12 hours ago)
Comment: --

Unmodified Added Removed Modified

v2	v3	
30	30	Now, several files need to be touched:
31	31	
32		1. deps.py:
	32	1. Please give this release an identifier by modifying release.py
	33	- Fill in release information: name and version
	34	1. For external dependencies, please adjust deps.py:
33	35	- Fill in dependencies dictionary (All external dependencies should have an entry :
34	36	- Choose the appropriate installer, and implement new installer if necessary (more [wiki:InstallerForExternalSoftware Installer for external software]).
35		1. release.py
36		- Fill in release information: name and version

Browseable Code & Document Source

Browser window: /diffraction (log) - diffraction - Trac
 http://dev.danse.us/trac/diffraction/log/diffraction

Legend: Added (green), Modified (yellow), Copied or renamed (blue)

View changes: ...

	Rev	Chgset	Date	Author	Log Message

Browser window: All Development Tracs - Trac
 http://dev.danse.us/trac/all/browser

<http://danse.us/trac/all/browser>

Browser window: Changeset 2095 - diffraction - Trac
 http://dev.danse.us/trac/diffraction/changeset/2095

Changeset 2095

Timestamp: 05/16/08 15:46:12 (1 week ago)
Author: farrowch

Message: Added cumulative Rw to pdfit2 and pdfgui.


Files:

- diffraction/diffpy/diffpy/Structure/Structure/expansion/makeCuboctahedron.py (1 diff)
- diffraction/diffpy/diffpy/pdfit2/pdfit2/PdfFit.py (1 diff)
- diffraction/diffpy/diffpy/pdfit2/pdfit2module/bindings.cc (2)
- diffraction/diffpy/diffpy/pdfit2/pdfit2module/misc.h (1 d)
- diffraction/diffpy/diffpy/pdfit2/tests/ExceptionsTest.py (1 d)
- diffraction/diffpy/diffpy/pdfit2/tests/TestPdfFit.py (1 diff)
- diffraction/diffpy/diffpy/pdfgui/pdfgui/control/fitdataset.py (2)
- diffraction/diffpy/diffpy/pdfgui/pdfgui/control/fitting.py (2)
- diffraction/diffpy/diffpy/pdfgui/pdfgui/control/plotter.py (2)
- diffraction/diffpy/diffpy/pdfgui/tests/testdata/lcmo.ddp (2)

Legend: Unmodified Added Removed Modified Copied Moved

diffraction/diffpy/diffpy/Structure/Structure/expansion/makeCuboctahedron.py

r2003	r2095	0	--	A Structure instance
14	14	0	--	A Structure instance
15	15	dist	--	Distance from center to nearest face
16	16	axis	--	Tuple defining the z-axis of the cuboctahedron (0,0,1).
17	17			
18	16			
19	17			Returns a new structure instance



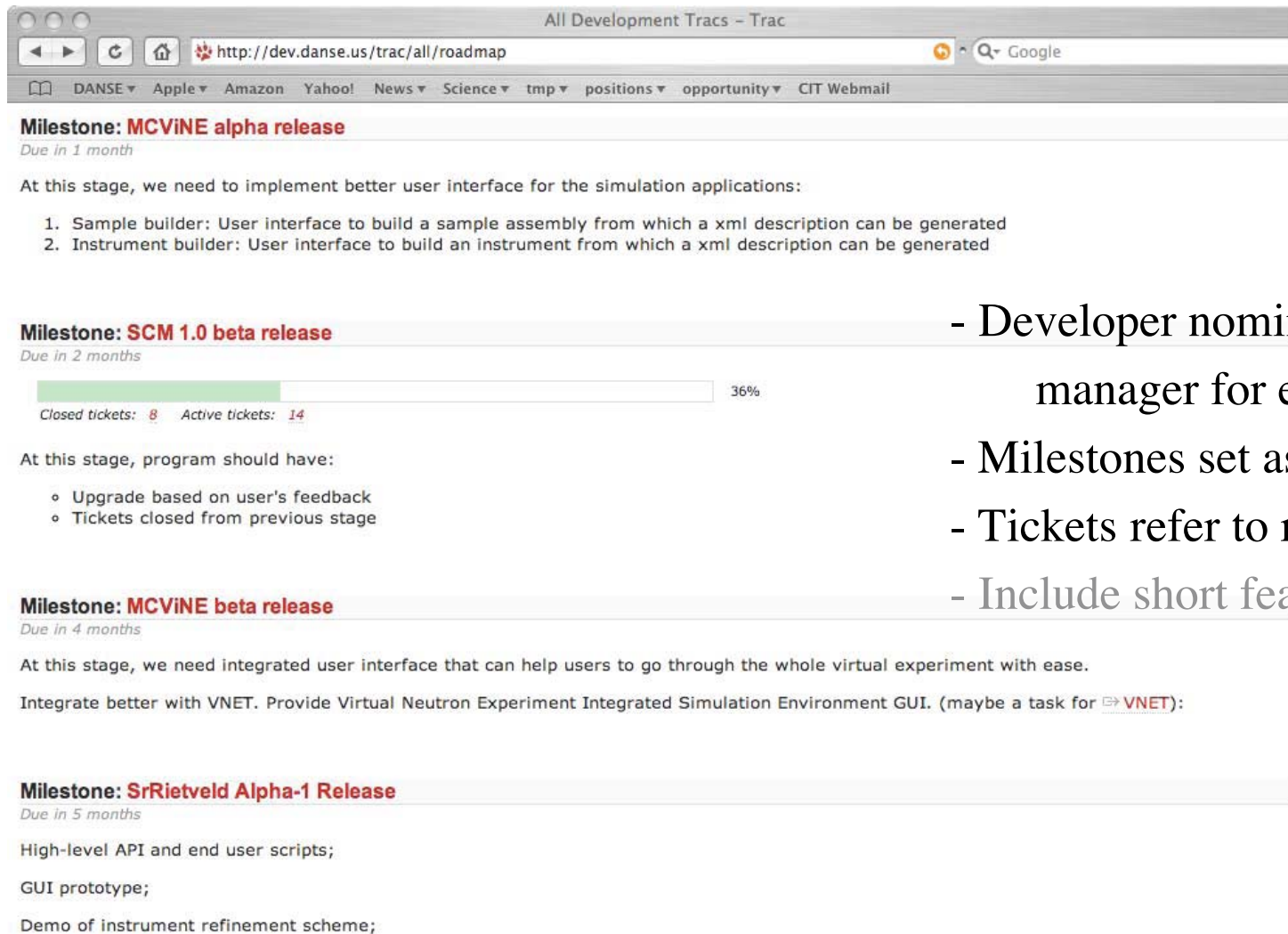
Wiki | Timeline | Roadmap | View

All Project Source

Name	Size	Rev	Age	Last Change
AbInitio		87	3 weeks	delaire:
Build Inelastic		271	1 day	linjiao: utils/installers/
common		36	6 days	brandon: rearranging
ctrl		1277	1 day	linjiao: docbook rules
DANSE tickets		1	0 days	No Author: Empty Rep
diffraction		2100	9 hours	wdzhou: some minor r
DrChops		21	1 week	linjiao: expose param
DynModels		71	1 month	mmckerns: core of op

Roadmap for Release Management

<http://danse.us/trac/all/roadmap>



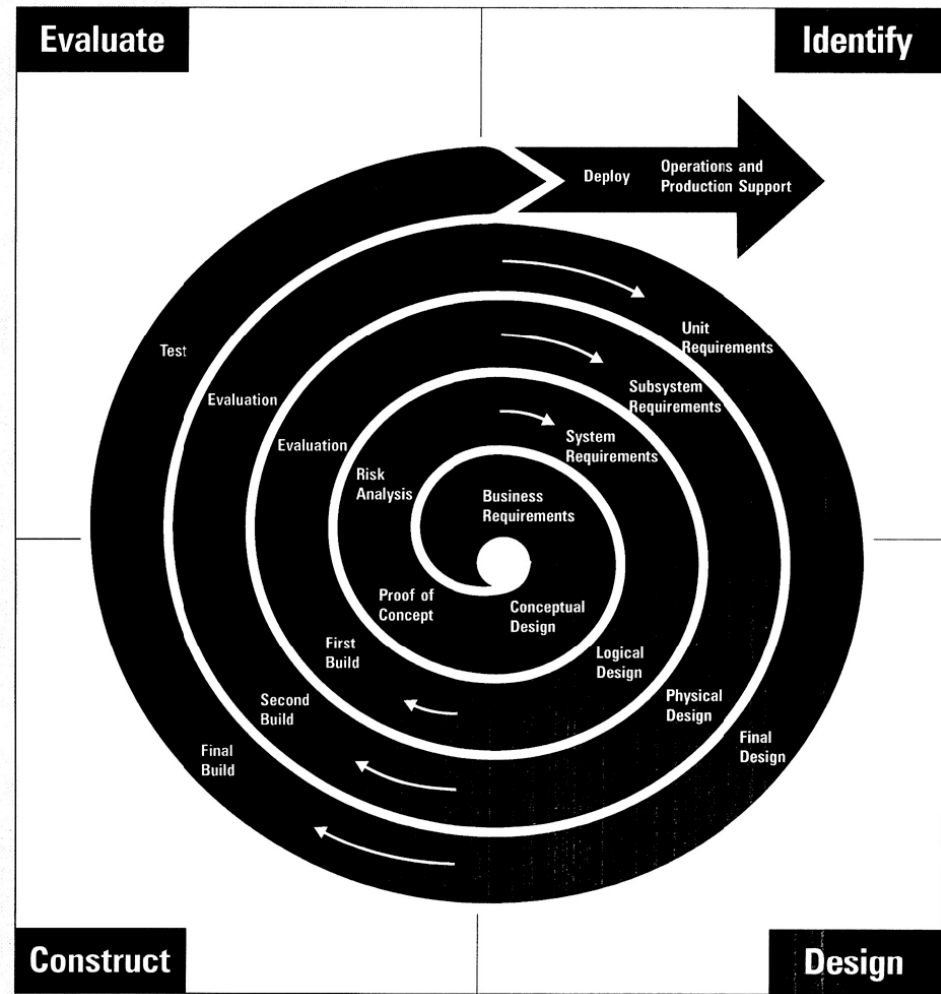
The screenshot shows a web browser window titled "All Development Tracs - Trac" with the URL "http://dev.danse.us/trac/all/roadmap". The browser's address bar and search engine (Google) are visible. The page content is organized into milestones:

- Milestone: MCVINE alpha release**
Due in 1 month
At this stage, we need to implement better user interface for the simulation applications:
 1. Sample builder: User interface to build a sample assembly from which a xml description can be generated
 2. Instrument builder: User interface to build an instrument from which a xml description can be generated
- Milestone: SCM 1.0 beta release**
Due in 2 months
A progress bar shows 36% completion. Below it, "Closed tickets: 8" and "Active tickets: 14" are listed.
At this stage, program should have:
 - Upgrade based on user's feedback
 - Tickets closed from previous stage
- Milestone: MCVINE beta release**
Due in 4 months
At this stage, we need integrated user interface that can help users to go through the whole virtual experiment with ease.
Integrate better with VNET. Provide Virtual Neutron Experiment Integrated Simulation Environment GUI. (maybe a task for [VNET](#)):
- Milestone: SrRietveld Alpha-1 Release**
Due in 5 months
High-level API and end user scripts;
GUI prototype;
Demo of instrument refinement scheme;

- Developer nominated as release manager for each package
- Milestones set as release versions
- Tickets refer to release milestones
- Include short feature summary

DANSE uses Good Development Practices

- Early design & definition
- Use cases & workflow
- Prototyping
- Iterative code growth
- Minimize refactoring
- Plan for “plug-in” development
- Utilize abstraction
- Document all interfaces
- Monitor and contribute to 3rd party solutions



Benefit of Good Development Practices

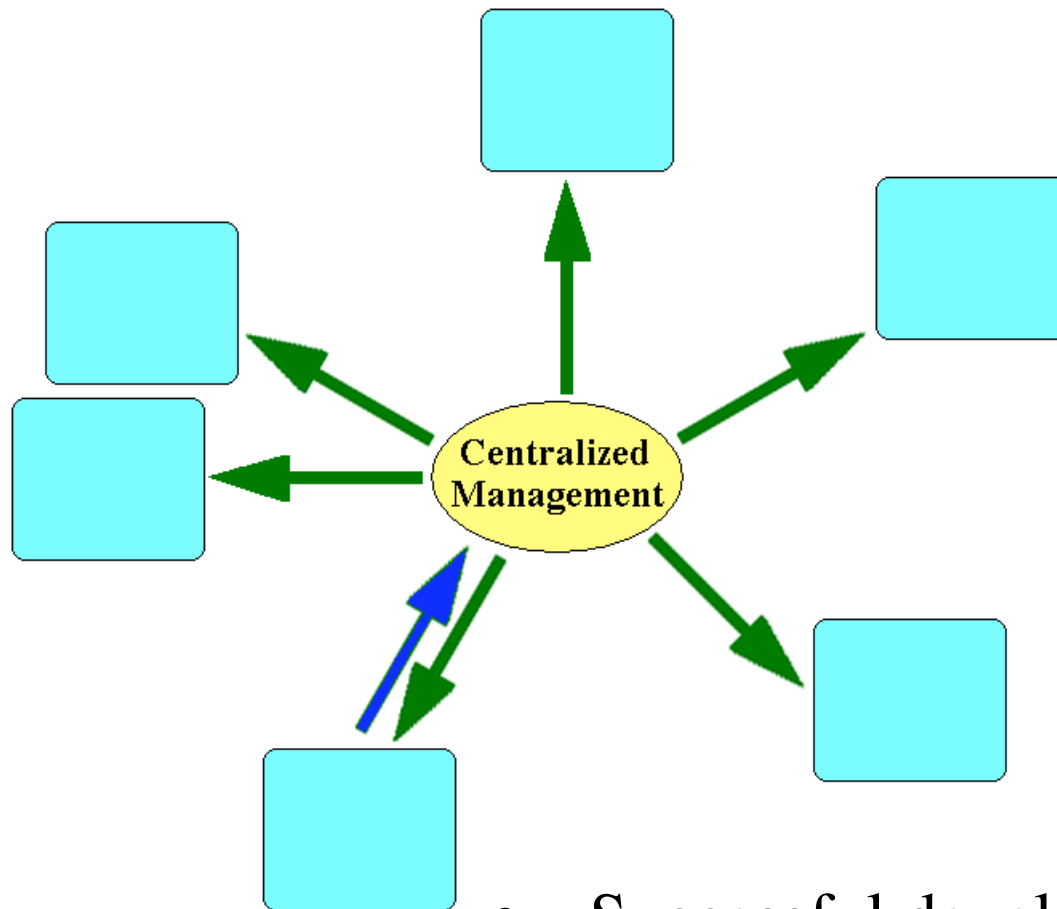
- Better software engineering enables greater reusability and flexibility, which in turn enables DANSE to be more efficient in building robust and durable software that will enable new science.

Summary: Risk in Software Projects

- Misperceptions of specifications are the biggest source of risk in software projects
- Good specifications and software design in the WBS (revisited as appropriate -- at least every 6 months)
- Subproject leaders know what science they want, our developers write use cases to flesh-out the specifications
- Reviews, audits, & quality assurance further minimizes risk
- Adopt commercial practices for scientific software development and management

End Presentation

Resource Management Tools



- centralized code repository
- email ticketing system
- systematic tracking
- structured design process
- build/testing infrastructure
- release management
- standardization
- quality assurance
- reporting mechanisms
- review & design steering

- Successful development with distributed resources requires centralized management.

Communication Processes

- **Coordination and Planning Mechanisms**
 - mailing lists and ticketing system
 - weekly technical and management meetings; [breakout sessions](#)
 - control boards
- **Production Mechanisms**
 - [versioning repositories](#)
 - standardized development and release process
 - 1-on-1 VNC sessions
- **Configuration Control and Reporting Mechanisms**
 - [repository tracking](#)
 - feature and bug tracking
 - automated builds and testing