Project Management

Mike McKerns, Caltech

- Building the Project Plan
- Managing the Project Plan
- Results and Progress

Goals & Objectives

- The goal of DANSE is to build a software system for neutron scattering research that:

  - integrates the basic data reduction capabilities that are available today
  - enables new types of science in all major subfields of neutron scattering research
  - provides a coherent framework onto which software components can be added by scientists
  - is maintainable by the SNS software group before the end of the project

Identifying the Project Scope

- Planning begins: January, 2002
  - requirements collection with neutron scientists
  - software workshops & reviews
  - email polls & surveys
- First draft for five-year project plan: November 2003
  - identified project tasks
  - problems with estimation of effort, risk, & duplication of effort
- First draft for Scope Baseline (WBS): August 2004
  - risk assessment & mitigation plan
  - resource leveling, task dependencies, & external drivers
- Project Descoped ($16.7M – $11.9M): March 2006
  - identified critical tasks & scope contingency

Building a Scope Baseline (task-by-task)

Level 5 Task DURATION

Version 1.1a **Task Name** MMTK

Task Description

WBS 5.3.5.4

Task Risk Factor 1 (enter as an integer) Risk Multiplier

Task Damage Factor 30% (enter as decimal) 0.30

0.26 4.42 1.17 2.60

less Scope

Scope weeks

	Allocation as decimal	Annual Rate	Hourly Rate	Number of Resources
Sr. Scientist				1.00
Postdoc	100%			1.00
Programmer	100%			1.00
Tech Writer	50%			1.00
Grad Student				1.00
Undergrad	75%			1.00
Other	75%			1.00

History:
Sr Sci, Postdoc, Programmer, Tech Writer, Grad were 4%, 75%, 75%, 75%, 75%

		8.45	8.45	275.60	$13,355	less Scope

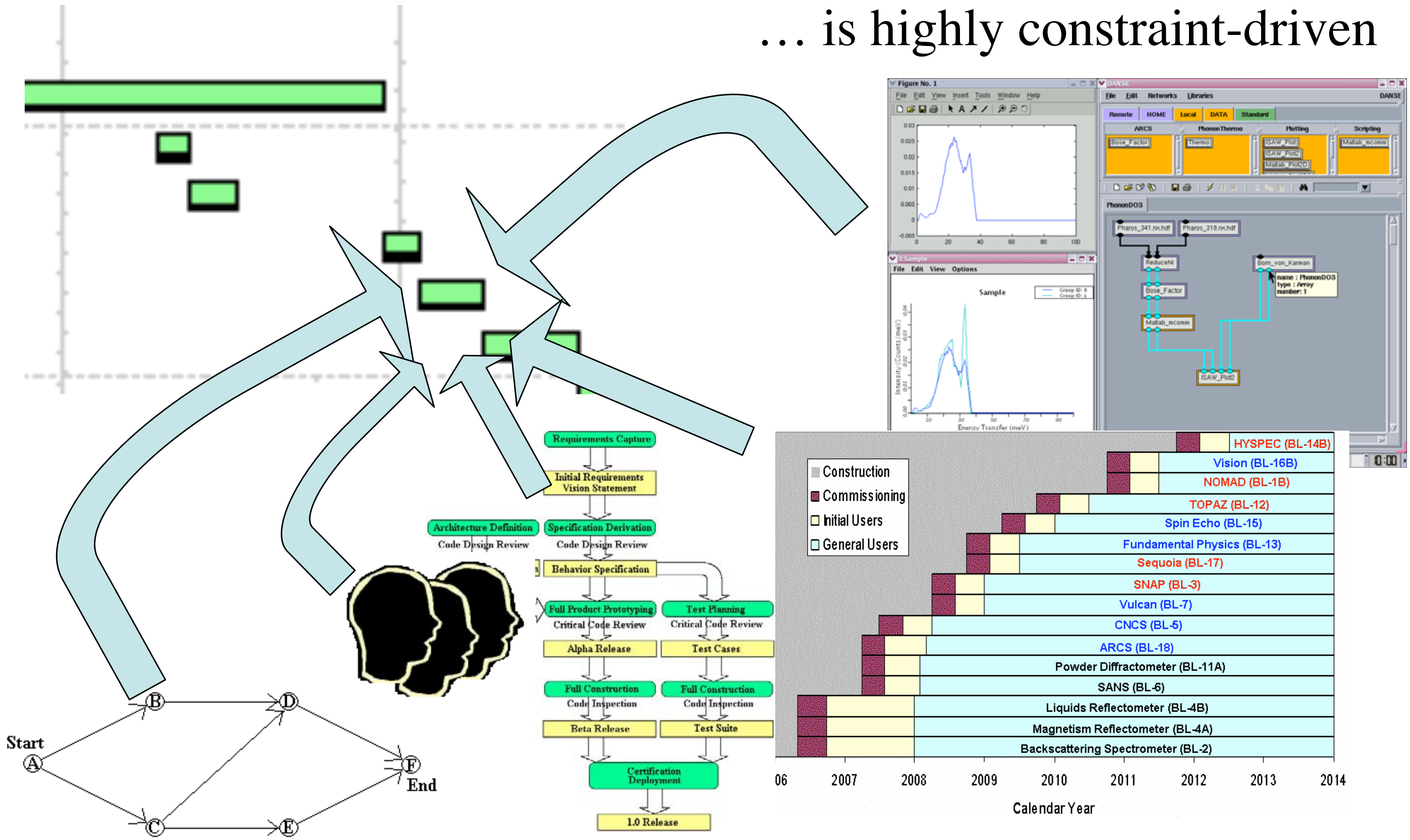
	Maximum Duration (weeks)	Minimum Duration (weeks)	Total Allocated Hours	Total cost
	8.45	8.45	275.60	$13,355

Scope Contingency

Task Name	Sr. Scientist	Postdoc	Programmer	Tech Writer	Grad Student	Undergrad	Other	Raw weeks	Risk Adj. Maximum Duration	Risk Adj. Minimum Duration	Risk Adj. Allocated Hours	Cost
Enter time in weeks	0.26	4.42	1.17	2.60								
Modify input file writer			0.4					0.40	0.52	0.52	20.80	$1,100
Write bindings to modules			0.1					0.10	0.13	0.13	5.20	$275
Modify output file reader			0.4					0.40	0.52	0.52	20.80	$1,100
Test/use case 1: energy minimization		0.2						0.20	0.26	0.26	10.40	$540
Test/use case 2: constrained molecular dynamics		0.2						0.20	0.26	0.26	10.40	$540
Test/use case 3: normal mode analysis		0.2						0.20	0.26	0.26	10.40	$540
Test/use case 4: operations on dynamic trajectories		0.2						0.20	0.26	0.26	10.40	$540
Test/use case 5: point charge fitting		0.2						0.20	0.26	0.26	10.40	$540
Test/use case 6: molecular surface calculations		0.2						0.20	0.26	0.26	10.40	$540
Write tutorial text		0.60						0.60	0.78	0.78	31.20	$1,620
Write tutorial code		0.60						0.60	0.78	0.78	31.20	$1,620
Write and integrate reference documentation				2				2.00	2.60	2.60	52.00	$1,700
Scientific review	0.20							0.20	0.26	0.26		
User testing and subsequent maintainance		1.00						1.00	1.30	1.30	52.00	$2,700

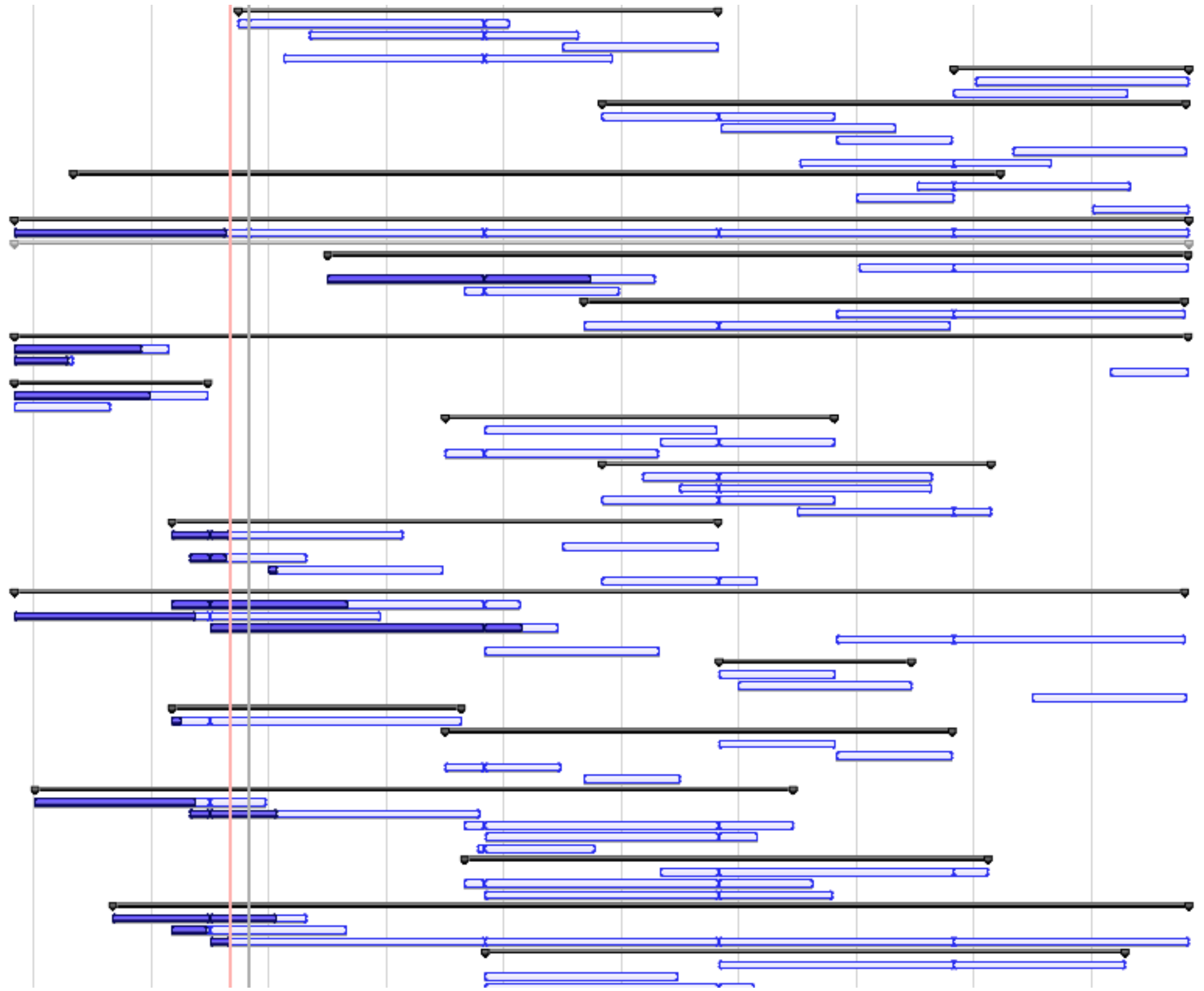
The Scheduling Process

… is highly constraint-driven



Five Year Software Construction Plan

50	9.2	Reflectometry 1D Modeling	5/15/07	5/28/09	0%
51	9.2.1.1	1D Models	5/15/07	7/8/08	0%
55	9.2.2.1	Basic 1D Analysis Launcher	9/3/07	10/24/08	0%
58	9.2.2.2	Extended 1D Analysis Launcher	10/1/08	5/28/09	0%
60	9.2.3	1D Analysis UI	7/25/07	12/16/08	0%
63	9.3.0	Experiment Planning	6/1/10	5/31/11	0%
64	9.3.0.1	Experiment Planning UI	7/5/10	5/31/11	0%
66	9.3.0.2	Experiment Planning Launcher	6/1/10	2/25/11	0%
68	9.3.1	Reflectometry 3D Modeling	12/1/08	5/27/11	0%
69	9.3.1.1	3D Models	12/1/08	11/26/09	0%
72	9.3.1.2	3D Model Builder	6/4/09	3/1/10	0%
74	9.3.1.3	Reflectometry Scattering Kernel	12/1/09	5/28/10	0%
76	9.3.1.4	3D Analysis UI	9/1/10	5/27/11	0%
78	9.3.1.5	3D Analysis Launcher	10/5/09	10/28/10	0%
81	9.3.2	Off-Specular Reflectometry	9/1/06	8/11/10	5%
94	9.3.3	Instrument Effects	4/5/10	3/1/11	0%
97	9.3.4	Bindings to OOMMF	1/1/10	5/31/10	0%
99	9.3.5	Bindings to GROMACS	1/3/11	5/31/11	0%
101	9.4	REFL Resources	6/1/06	5/31/11	18%
120	11.3.2	REFL Minority Student Research	6/1/06	5/31/11	18%
8	10	INS	6/1/06	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	5/1/08	12/26/08	0%
11	5.1.2	Advanced Data Reduction	11/3/08	5/25/11	0%
12	5.1.2.1	Advanced Data Reduction	12/1/09	5/25/11	0%
15	5.1.2.2	Reduction Corrections	11/3/08	5/25/10	0%
18	5.3.1	Instrument Simulation	6/1/06	5/30/11	62%
19	5.3.1.1	Monte-Carlo Instrument Simulation Framework	6/1/06	1/26/07	82%
21	5.3.1.2	Bindings to McStas	6/1/06	8/30/06	91%
23	5.3.1.4	Bindings to VITESS	1/31/11	5/30/11	0%
25	5.3.2	Sample Simulation	6/1/06	3/28/07	47%
26	5.3.2.1	Sample Simulation Framework	6/1/06	3/28/07	70%
28	5.3.2.2	Coherent Elastic Scattering Kernel	6/1/06	10/27/06	0%
30	5.3.3	Solid-State Simulation	4/1/08	11/26/09	0%
31	5.3.3.0	Solid-State Materials Simulation	6/2/08	5/27/09	0%
33	5.3.3.1	Bindings to Abinit	3/2/09	11/26/09	0%
38	5.3.3.2	Bindings to VASP	4/1/08	2/25/09	0%
39	5.3.4	Hartree-Fock Simulations	12/1/08	7/28/10	0%
40	5.3.4.0	Hartree-Fock Materials Simulation	2/2/09	4/27/10	0%
43	5.3.4.1	Bindings to GAMESS	4/1/09	4/26/10	0%
46	5.3.4.2	Bindings to NWChem	12/1/08	11/26/09	0%
49	5.3.4.3	Bindings to FIREBALL	10/1/09	7/28/10	0%
52	5.3.5	Molecular Dynamics Simulations	2/1/07	5/28/09	15%
53	5.3.5.0	Molecular Dynamics Materials Simulation	2/1/07	1/25/08	25%
56	5.3.5.1	Bindings to NAMD	10/1/08	5/28/09	0%
58	5.3.5.4	Bindings to MMTK	3/1/07	8/28/07	30%
61	5.3.5.5	Bindings to GULP	7/2/07	3/27/08	5%
63	5.3.6	Scattered Intensity	12/1/08	7/28/09	0%
66	5.4.1	Data Structures	6/1/06	5/25/11	42%
67	5.4.1.1	Reduction Data Structures	2/1/07	7/25/08	50%
71	5.4.1.2	Experiment Metadata Containers	6/1/06	12/21/07	50%
74	5.4.1.3	Common Array Manipulations	4/2/07	9/22/08	90%
77	5.4.1.4	Advanced Array Manipulations	12/1/09	5/25/11	0%
80	5.4.1.8	Database Accessor	6/2/08	2/26/09	0%
82	5.4.3	Basic Numerical Transformations	6/1/09	3/26/10	0%
83	5.4.3.0	Numerical Libraries	6/1/09	11/27/09	0%
85	5.4.3.2	Common Scattering Functions	7/1/09	3/26/10	0%
87	5.4.4	Error Propagation Algorithms	10/1/10	5/27/11	0%
89	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	4/1/08	5/28/10	0%
94	5.4.6.2	Bindings to Molden	6/1/09	11/27/09	0%
96	5.4.6.6	Bindings to VMD	12/1/09	5/28/10	0%
98	5.4.6.7	Molecular & Crystal Structure Format Translator	4/1/08	9/26/08	0%
101	5.4.6.8	Crystallographic Database Accessor	11/3/08	3/31/09	0%
103	5.4.7	Analysis Support Applications	7/3/06	9/23/09	20%
104	5.4.7.1	Histogram Viewer	7/3/06	6/26/07	70%
107	5.4.7.2	Instrument Builder	3/1/07	5/23/08	30%
110	5.4.7.3	Sample Builder	5/1/08	9/23/09	0%
114	5.4.7.4	Structure Builder	6/3/08	7/28/09	0%
117	5.4.7.5	Molecular Format Translator	5/22/08	11/18/08	0%
120	5.4.8	Sample Materials Applications	5/1/08	7/23/10	0%
121	5.4.8.1	Materials Simulation Launcher	3/2/09	7/23/10	0%
125	5.4.8.2	Sample Dynamics Launcher	5/1/08	10/23/09	0%
129	5.4.8.3	Structure Dynamics Launcher	6/2/08	11/23/09	0%
132	5.5.0	Maintenance & Updates	11/1/06	5/31/11	16%
133	5.5.0.a	Extensions to Distutils	11/1/06	8/28/07	85%
135	5.5.0.b	Geometric Shapes	2/1/07	10/29/07	20%
139	5.5.0.x	Maintenance & Updates	4/2/07	5/31/11	2%
144	10.1.0	INS Data Reduction	6/2/08	2/21/11	0%
145	10.1.0.0	Single Crystal Inelastic Reduction Drivers	6/1/09	2/21/11	0%
148	10.1.0.1	Polycrystal Inelastic Reduction Drivers	6/2/08	3/27/09	0%



The Project Plan Must be Agile

- The Project Plan must accommodate:
  - new developments in available software
  - the talents of current developers
  - refinement of ideas, scope, requirements, and design
- Regular refinements of schedule and tasks (6 months)
  - improve architectural design of common algorithms packages
  - create new tasks as code design becomes more extensible
- Baseline plan was good; however, not detailed enough…

- Difficulties encountered with planning occurring at task start
  - drive to design more reusable code lessened when “clock is ticking”
  - new requirements make Central Services & Common Algs schedule unstable

Solidify Initial Scope & Design…

- Ensure a clear definition of all deliverables
  - 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

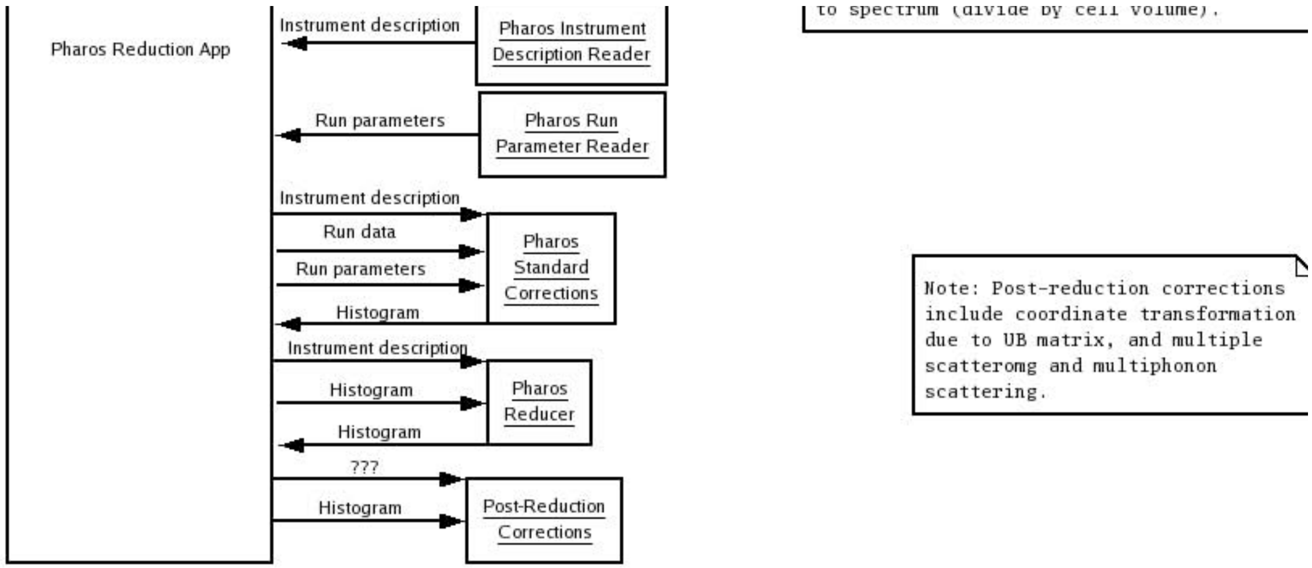
- Ensure identification of all tasks
  - list our objectives and our requirements for each subgroup
  - associate each requirement with at least one task
  - associate each deliverable with at least one task
  - identify if task is well-defined, digestible, and meets design requirements
  - identify if software has the desired interfaces

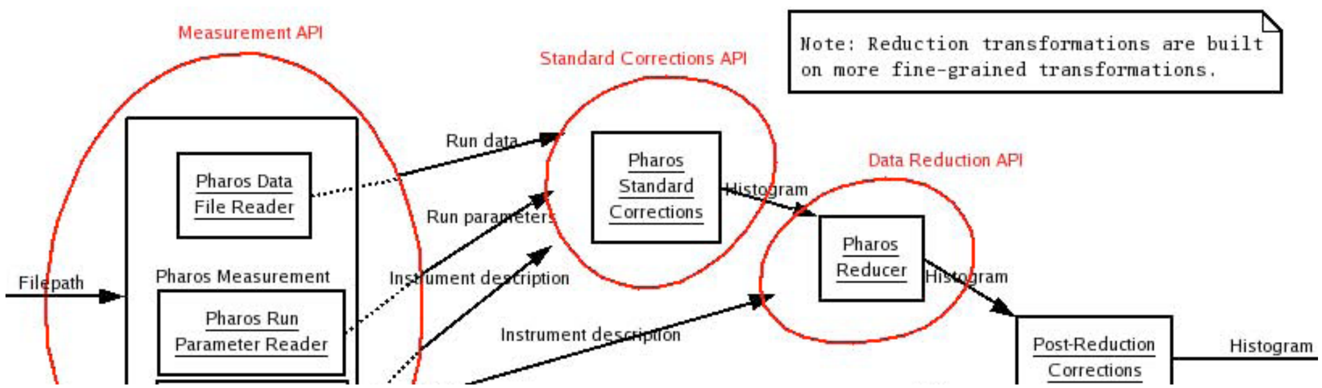
…then Refine the WBS

- Building tasks with narrower scope
  - creates additional tasks
  - 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

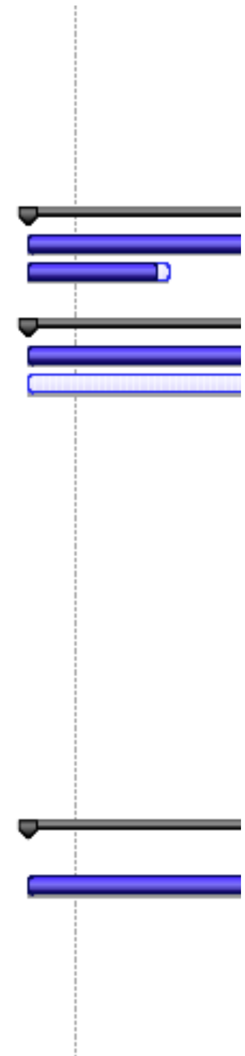
Capture High-level Specifications





Capture Good Engineering Design in WBS

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	5/1/08	12/26/08	0%
11	5.1.2	Advanced Data Reduction	11/3/08	5/25/11	0%
12	5.1.2.1	Advanced Data Reduction	12/1/09	5/25/11	0%
15	5.1.2.2	Reduction Corrections	11/3/08	5/25/10	0%
18	5.3.1	Instrument Simulation	6/1/06	5/30/11	62%
19	5.3.1.1	Monte-Carlo Instrument Simulation Framework	6/1/06	1/26/07	82%
21	5.3.1.2	Bindings to McStas	6/1/06	8/30/06	91%
23	5.3.1.4	Bindings to VITESS	1/31/11	5/30/11	0%
25	5.3.2	Sample Simulation	6/1/06	3/28/07	47%
26	5.3.2.1	Sample Simulation Framework	6/1/06	3/28/07	70%
28	5.3.2.2	Coherent Elastic Scattering Kernel	6/1/06	10/27/06	0%
30	5.3.3	Solid-State Simulation	4/1/08	11/26/09	0%
31	5.3.3.0	Solid-State Materials Simulation	6/2/08	5/27/09	0%
33	5.3.3.1	Bindings to Abinit	3/2/09	11/26/09	0%
36	5.3.3.2	Bindings to VASP	4/1/08	2/25/09	0%
39	5.3.4	Hartree-Fock Simulations	12/1/08	7/28/10	0%
40	5.3.4.0	Hartree-Fock Materials Simulation	2/2/09	4/27/10	0%
43	5.3.4.1	Bindings to GAMESS	4/1/09	4/26/10	0%
46	5.3.4.2	Bindings to NWChem	12/1/08	11/26/09	0%
49	5.3.4.3	Bindings to FIREBALL	10/1/09	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	25%
56	5.3.5.1	Bindings to NAMD	10/1/08	5/29/09	0%
58	5.3.5.4	Bindings to MMTK	3/1/07	8/28/07	30%
61	5.3.5.5	Bindings to GULP	7/2/07	3/27/08	5%
63	5.3.6	Scattered Intensity	12/1/08	7/29/09	0%
66	5.4.1	Data Structures	6/1/06	5/25/11	42%
67	5.4.1.1	Reduction Data Structures	2/1/07	7/25/08	50%
71	5.4.1.2	Experiment Metadata Containers	6/1/06	12/21/07	50%
74	5.4.1.3	Common Array Manipulations	4/2/07	9/22/08	90%
77	5.4.1.4	Advanced Array Manipulations	12/1/09	5/25/11	0%
80	5.4.1.8	Database Accessor	6/2/08	2/26/09	0%
82	5.4.3	Basic Numerical Transformations	6/1/09	3/26/10	0%
83	5.4.3.0	Numerical Libraries	6/1/09	11/27/09	0%
85	5.4.3.2	Common Scattering Functions	7/1/09	3/26/10	0%



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?

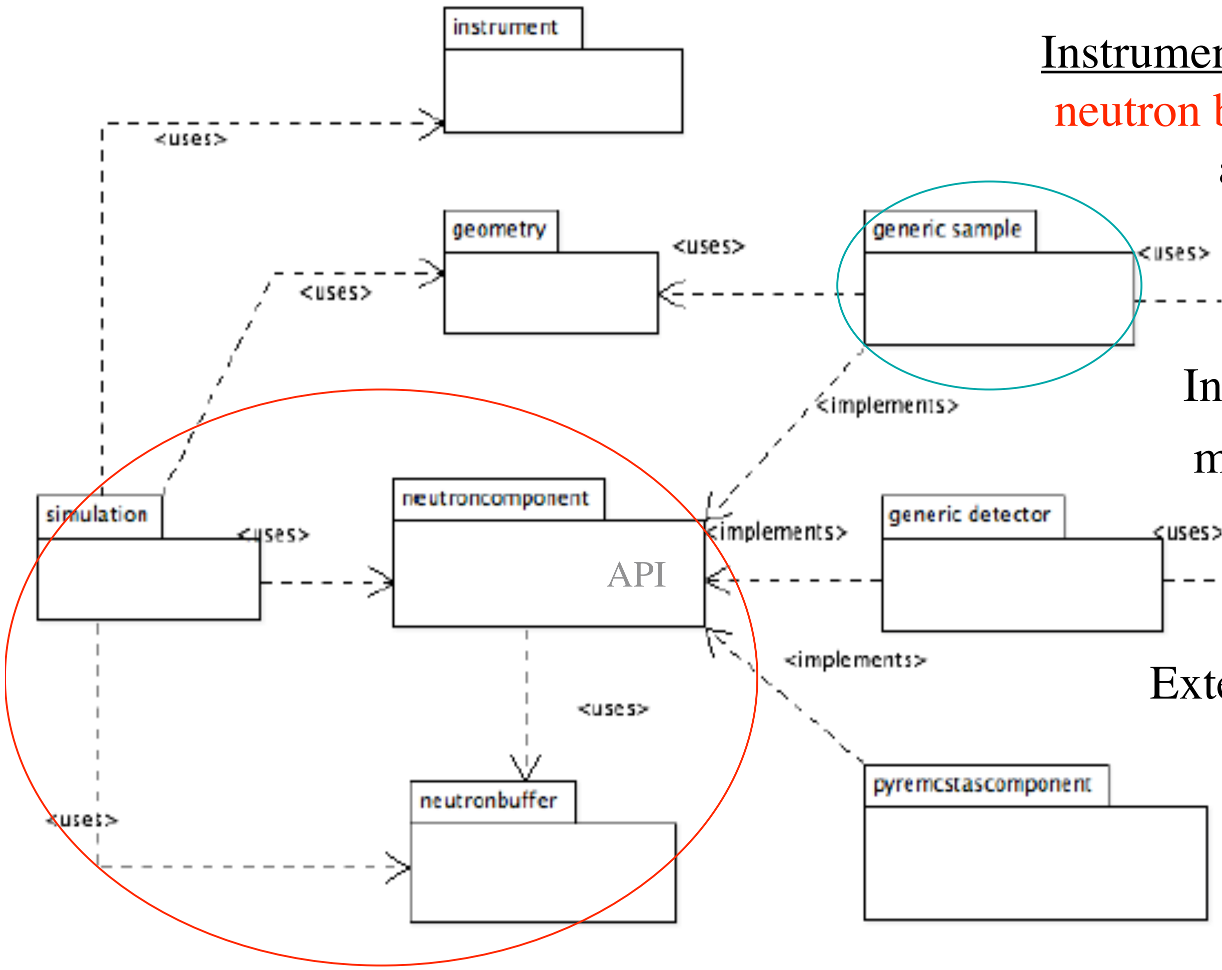
Michael McKerns 1/16/07 11:20 AM
Comment: Executable configure and launch.

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.

Ensure Well-defined Package Evolution

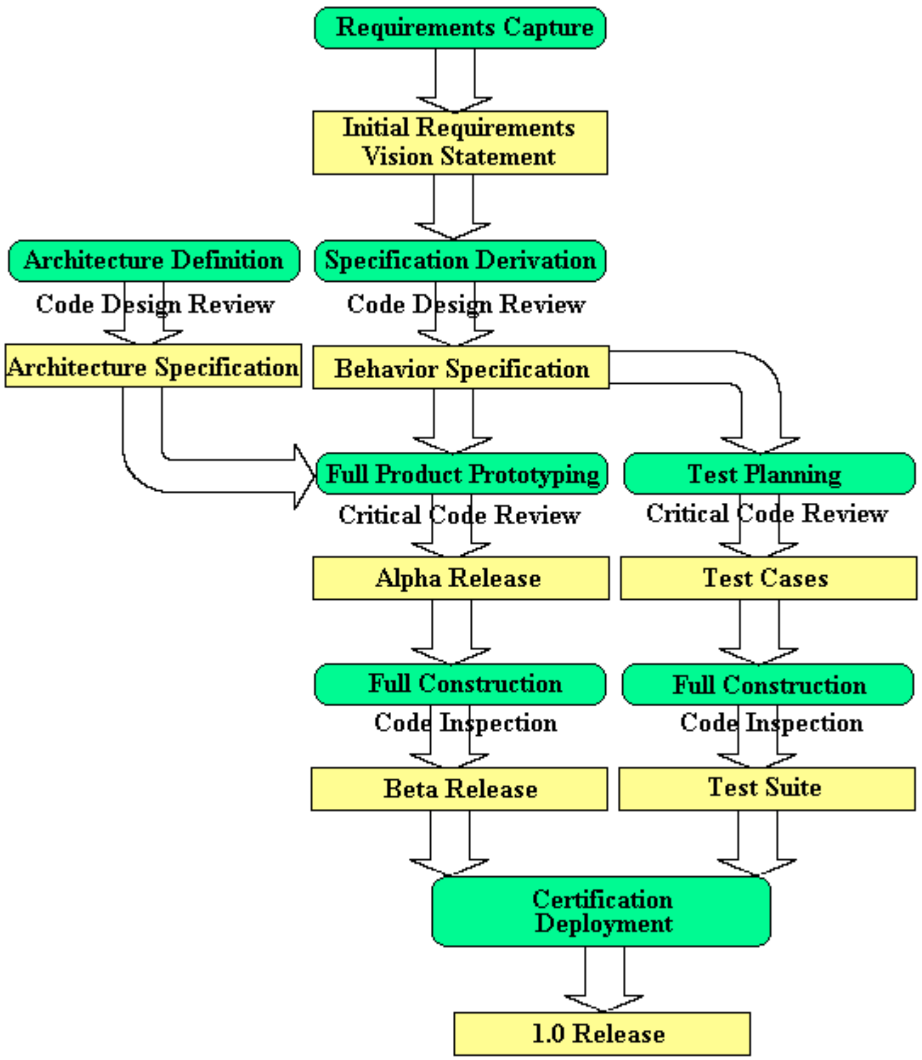


Instrument simulation is simulation engine, neutron buffer, and neutron components… and uses instrument and geometry

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

Extend by building neutron component with generic sample construction

Standardized Software Production Process



- review and quality control process integrated with development

- review and testing process enable efficient software engineering

- producing documentation and specifications early in the process minimizes rework and eases user manual generation

- full code construction and core documentation by beta release

Standardized Reviews (SQA Guidelines)

- Planning Review

  - vision statement summarizing purpose, functionality, and acceptance criteria

- Design Review

  - UML & auto-generated interface documentation for architecture specification

  - refinement of requirements and use cases into behavior specification

- Code Review

  - both component/application and test code inspected

  - automated test for quality standards, manual inspection for code functionality

- Reviews involve Architect, PM, QA, SubPI, developers

  - reviewers should be stakeholders in the code

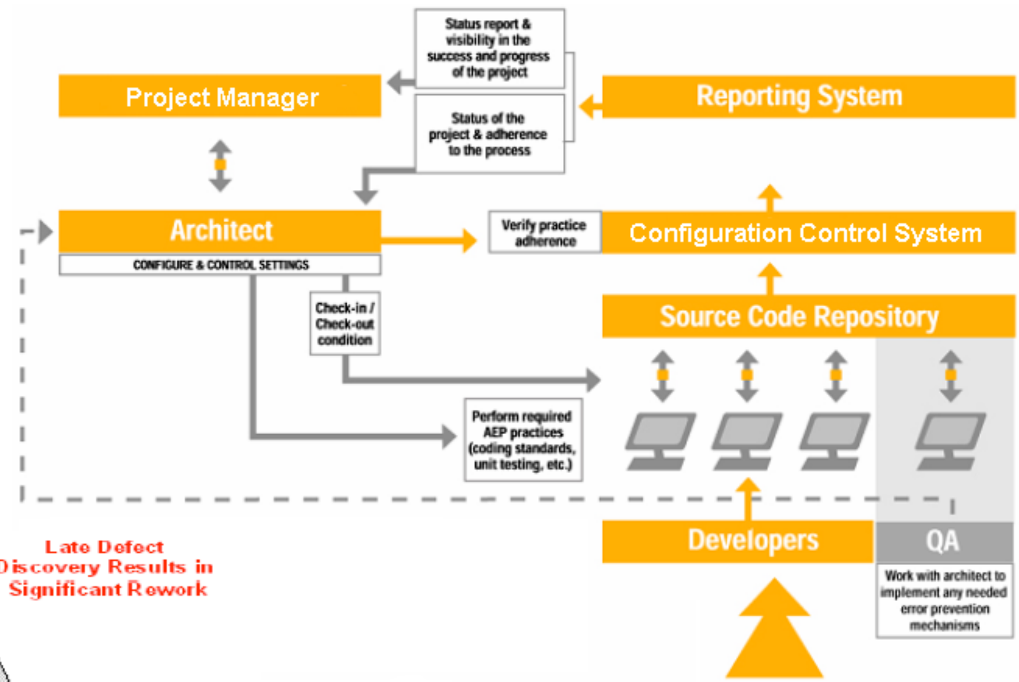
  - review action items entered to project tracking system

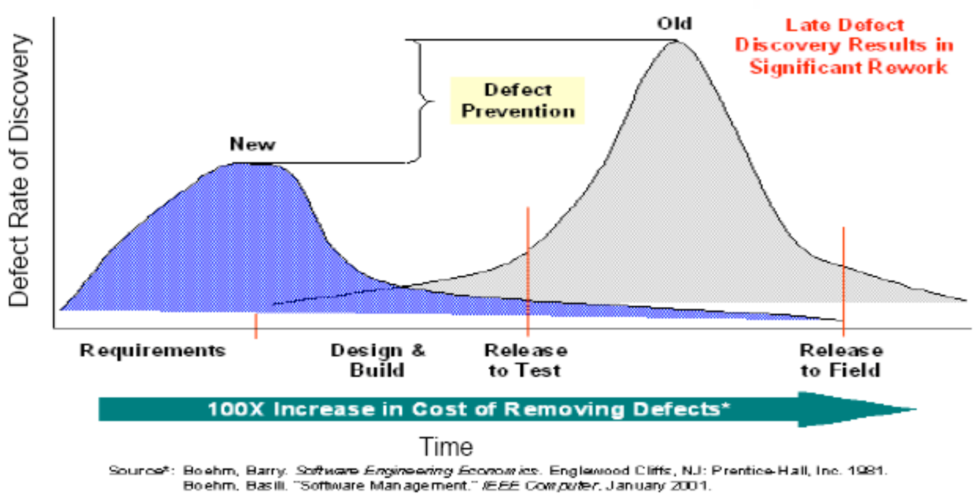
  - reviews are critical in developing the most commonly reused tasks

  - audits by PM, QA to verify progress and quality standards

Software Testing Process

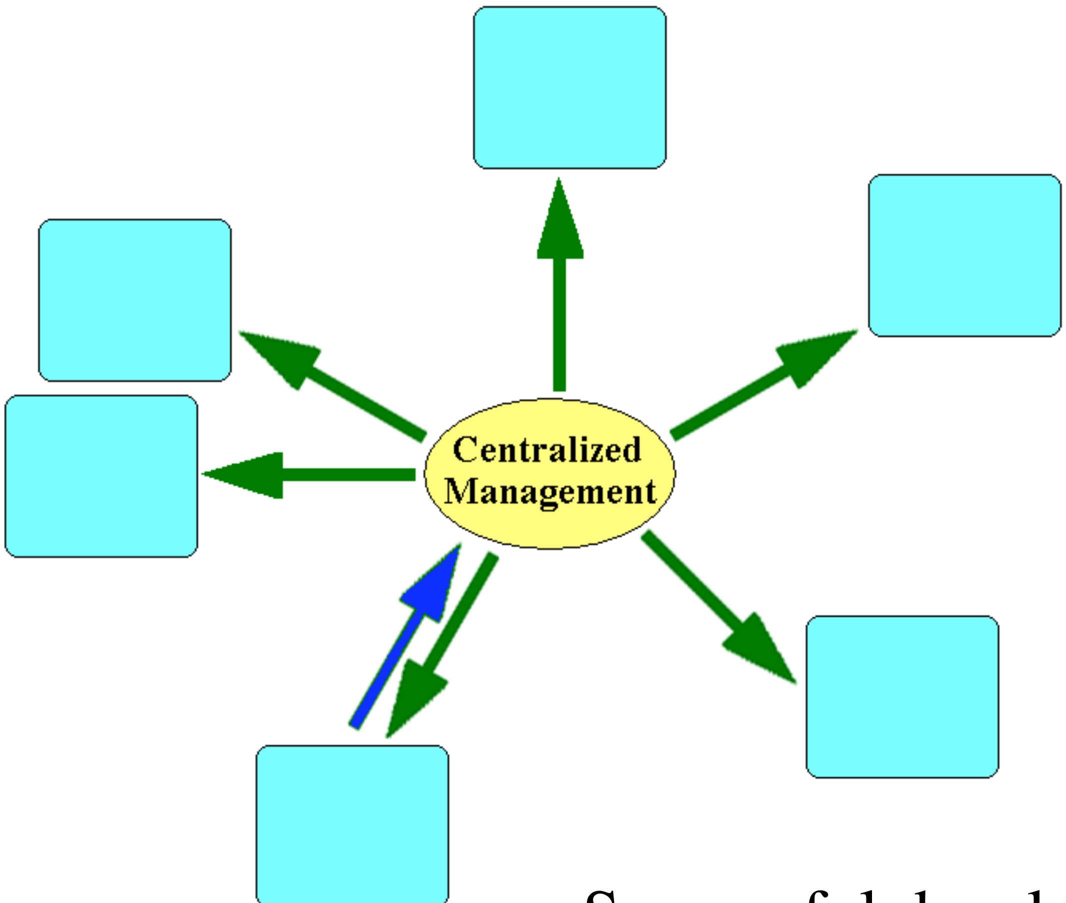
- automated builds tied to code repository & reporting system
- automated tests of process and standards adherence
- manual inspection of functional behavior





- tests developed concurrent to code
- rapid bug identification, notification
- overall product quality and developer efficiency increases

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

Tracking the Development Process

- Integrated tracking system monitors development & bugs
  - based on standard bug tracking system
  - bug reports from users and developers
  - WBS tasks added as feature requests
  - review action items entered as feature requests
  - error reports from automated builds and test harness
- Repository check-ins provide additional information
  - check-in messages reference to archived design documentation
  - finer-grain detail provided by repository 'diff' command
- EVM metrics will be tested for weighting task completion
  - tracking based on production process (design, review, test, release)
  - tracking based on functional requirements and specifications

Metrics Captured as Audit Checklist

Top three risks:	See Requirement document: - 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 submitted as tickets in the online bug and issue tracking system.

http://danse.us/trac/tickets/ticket/178

Earned Value Management

- Earned Value Management provides estimate of project progress and adherence to project plan
  - Sequenced tasks in WBS provide coupling of expected (scheduled) work completion to expected cost
  - Software Production Plan identifies basis for estimation of % complete, and is captured in the Statement of Work
  - Project Manager and Subproject Leader use project tracking tools to map work done to % complete
  - Monthly reports from subproject business systems provide actual costs

- EVM serves as a guide for project change decisions

Monthly Capture of EVM & Issues

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

Project is on-schedule, but…

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

Summary

- Project Plan is good, and always improving
- Project is on schedule
- Requirements gathering & scope clarification progressing
- Design and planning sessions have been successful
- Collaboration with the SNS & community has accelerated

End Presentation