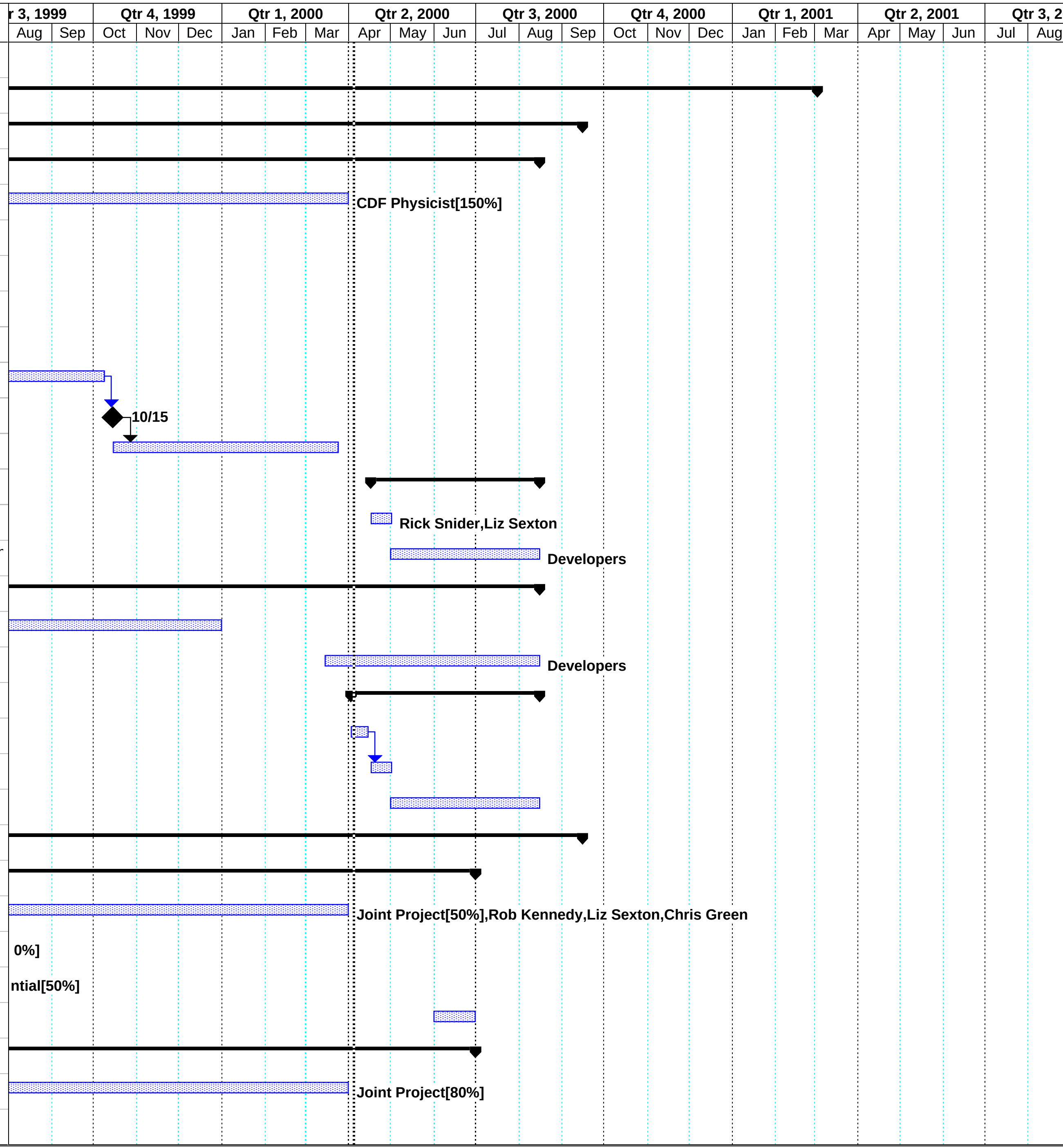
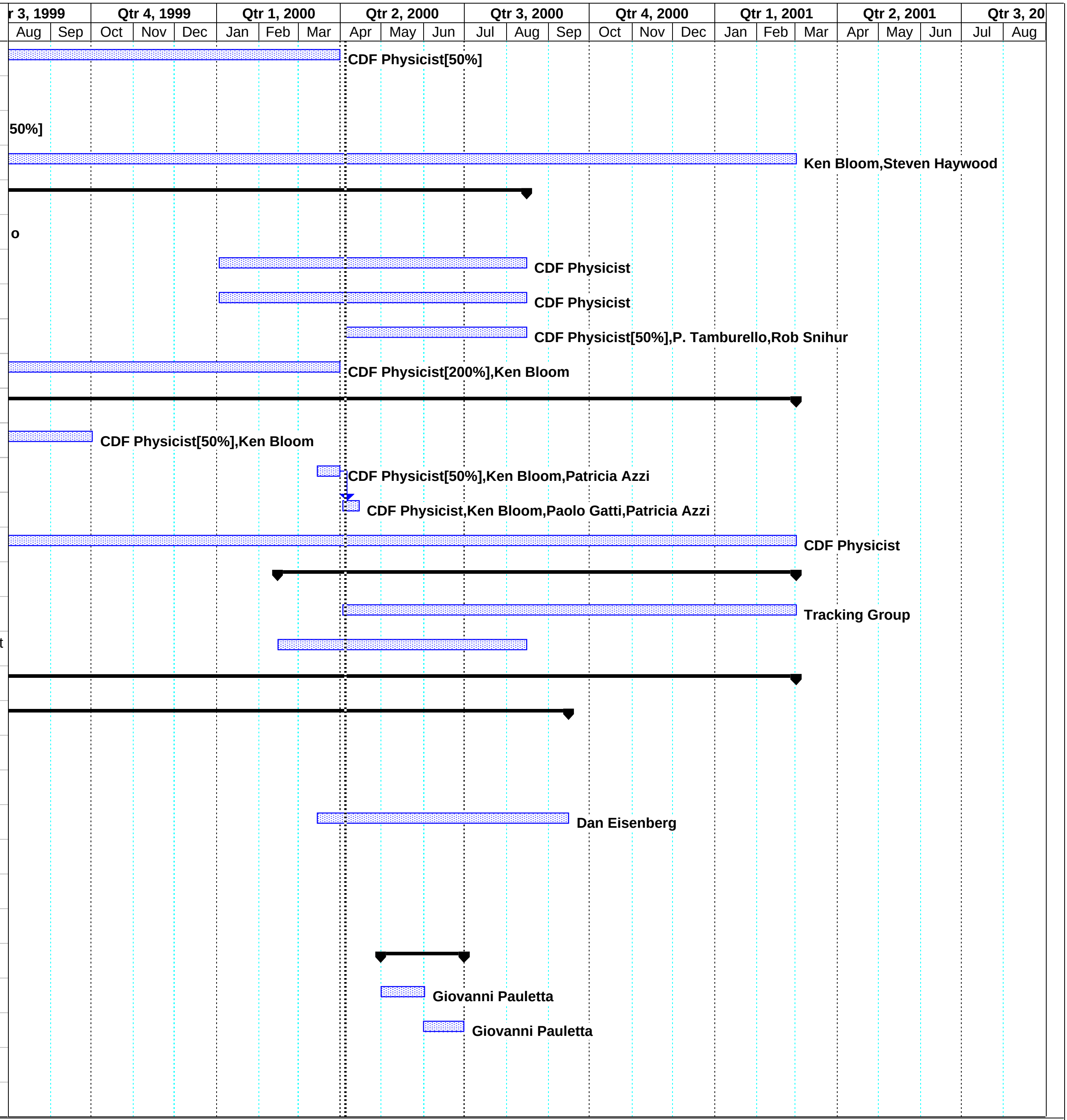
ID	WBS	Task Name
1	1	Rest of CDF
2	**2**	**CDF Computing and Software Upgrade**
3	**2.1**	**Code Integration**
4	**2.1.1**	**Production Executable**
5	2.1.1.1	Management
6	2.1.1.2	First integration
7	2.1.1.3	October Executable
8	2.1.1.4	Second Integration
9	2.1.1.5	Full prototype
10	2.1.1.6	Integration/Optimization
11	2.1.1.7	Mock Data Challenge
12	2.1.1.8	Second Optimization
13	**2.1.1.9**	**Error Logging and Handling**
14	2.1.1.9.1	Policy and Guide for Error Logging
15	2.1.1.9.2	Implementing Error Handling in Reconstructior
16	**2.1.1.10**	**Communications with Database**
17	2.1.1.10.1	Example of Talking to Database
18	2.1.1.10.2	Use by Tracking, Muons, Calorimetry, etc.
19	**2.1.1.11**	**Access to Alignment**
20	2.1.1.11.1	Formation of Alignment Group
21	2.1.1.11.2	Definition of Infrastucture and Constants
22	2.1.1.11.3	Code Infrastructure
23	**2.1.2**	**Infrastructure**
24	**2.1.2.1**	**C++ Support**
25	2.1.2.1.1	Compliers and Debuggers
26	2.1.2.1.2	Class Browser
27	2.1.2.1.3	EDG_decode
28	2.1.2.1.4	Use of ISOCXX portability package
29	**2.1.2.2**	**Analysis Tools**
30	2.1.2.2.1	PAT ROOT support
31	2.1.2.2.2	Formatted Bank Dump

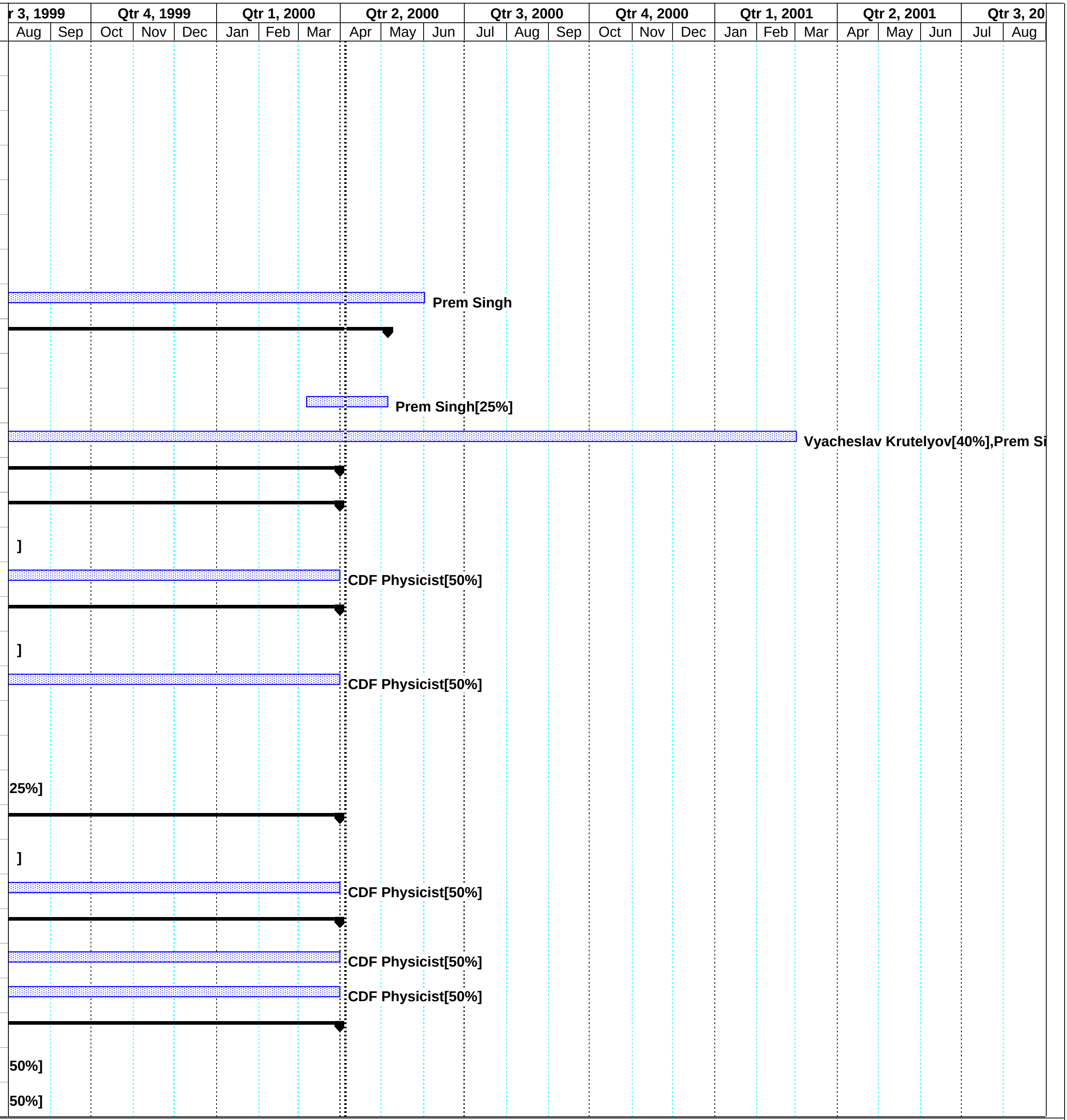


ID	WBS	Task Name	Resources
32	2.1.2.2.3	Build Job++	**Student[50%]**
33	2.1.2.2.4	Zoom: heptuple, error log	
34	**2.1.2.3**	**Library Maintenance**	**CDF Physicist[25%]**
35	**2.1.2.4**	**Redesign Geometry**	
36	2.1.2.4.1	Design/implementation	**Chris Green,Jim Kowalkowski,Marc Patterno**
37	2.1.2.4.2	Database/Persistence	
38	2.1.2.4.3	Extrapolation Interface	**CDF Physicist[50%]**
39	**2.1.3**	**Production Runtime Configuration (L4)**	
40	2.1.3.1	Prototype production parser	**Kirsten Tollefson[30%]**
41	2.1.3.2	Prototype production table	**Rick Snider**
42	2.1.3.3	Prototype Input validation control	**Mark Lancaster[30%]**
43	2.1.3.4	Data File Catalog Interface	**CDF Physicist[30%]**
44	2.1.3.5	Optimization	**CDF Physicist**
45	**2.1.4**	**Validation**	
46	2.1.4.1	Validation Modules	
47	**2.2**	**Event Reconstruction**	
48	2.2.1	Luminosity	**CDF Physicist,Luminosity Group**
49	**2.2.2**	**Tracking**	
50	**2.2.2.1**	**High Level track Class**	
51	2.2.2.1.1	Basic Design	**],Rick Snider,Joe Boudreau**
52	2.2.2.1.2	Remake for EDM design	**CDF Physicist[25%],Rick Snider,David Waters**
53	2.2.2.1.3	CDFTrack in new EDM	
54	2.2.2.1.4	Re-implement CDFTrack persistency	**Rick Snider,Dave Waters**
55	**2.2.2.2**	**EventVertex**	
56	2.2.2.2.1	Fast Event Vertex	
57	2.2.2.2.2	VXPrim 3D Vertex	**Karlsruhe**
58	2.2.2.2.3	Generic Vertex Fitting	
59	2.2.2.2.4	constrained mass fitting (CTVMFT)	
60	**2.2.2.3**	**Silicon Reconstruction**	
61	2.2.2.3.1	Z side clustering and charge depo	**Matt Herndon[10%],Rick Snider[10**
62	2.2.2.3.2	Standalone algorithm development	

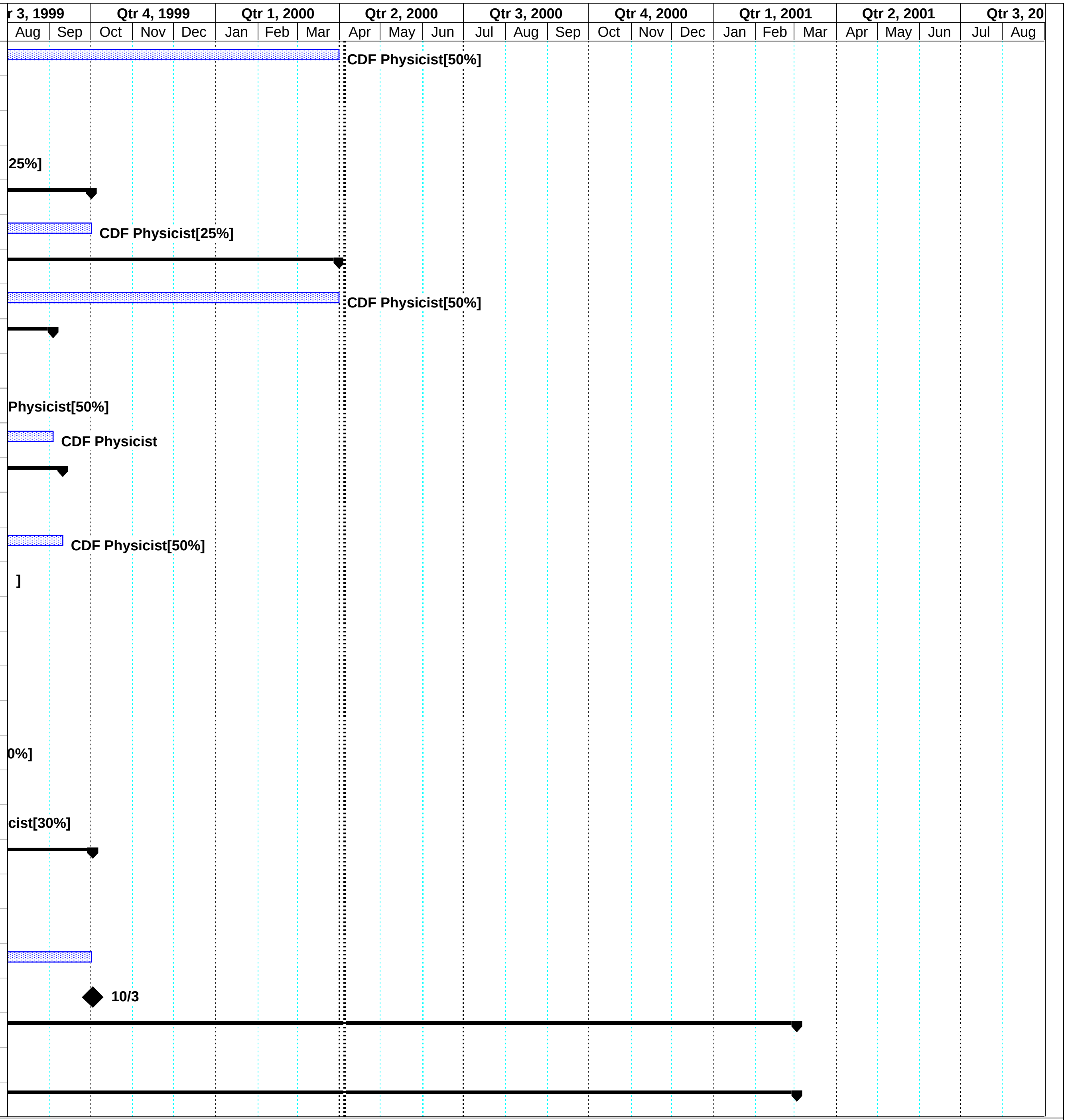
ID	WBS	Task Name
63	2.2.2.3.3	Reorganize geometry
64	2.2.2.3.4	Alignment Hooks
65	2.2.2.3.5	Material Integrator
66	2.2.2.3.6	Optmization
67	**2.2.2.4**	**COT Reconstruction**
68	2.2.2.4.1	Reorganize geometry
69	2.2.2.4.2	R-phi Algorithm development
70	2.2.2.4.3	Stereo algorithm development
71	2.2.2.4.4	Port Run I fitter
72	2.2.2.4.5	Optmization
73	**2.2.2.5**	**COT-into-SVX Reconstruction**
74	2.2.2.5.1	Integrate outside-in code
75	2.2.2.5.2	Integrate silicon z-hit finding
76	2.2.2.5.3	Full integration & testing
77	2.2.2.5.4	Optmization
78	**2.2.2.6**	**multi-algorithm, multi-fit, integration**
79	2.2.2.6.1	picking the best track for a set of hits
80	2.2.2.6.2	multiple fit infrastructure (CDFTrack needs list
81	**2.2.3**	**Muons**
82	**2.2.3.1**	**CMU**
83	2.2.3.1.1	D to E
84	2.2.3.1.2	E to S
85	2.2.3.1.3	E to S in C++
86	**2.2.3.2**	**CMP**
87	2.2.3.2.1	D to E
88	2.2.3.2.2	E to S
89	**2.2.3.3**	**CSP**
90	2.2.3.3.1	D to E
91	2.2.3.3.2	E to S
92	**2.2.3.4**	**CMX**
93	2.2.3.4.1	D to E



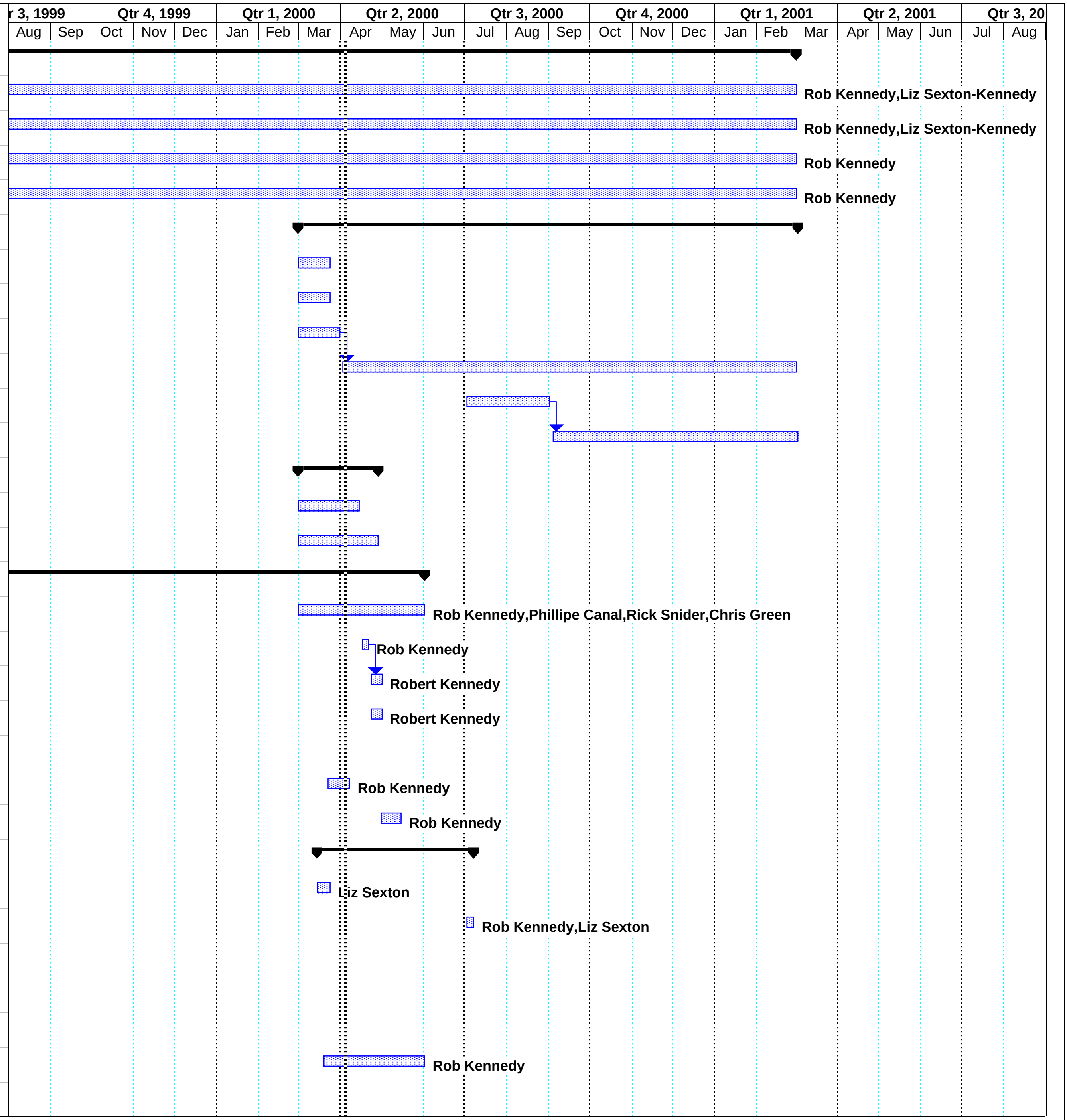
ID	WBS	Task Name
94	2.2.3.4.2	E to S
95	**2.2.3.5**	**CSX**
96	**2.2.3.6**	**IMU Muons**
97	2.2.3.6.1	D to E
98	2.2.3.6.2	E to S
99	**2.2.3.7**	**CMLNK**
100	2.2.3.7.1	Implementation
101	2.2.3.8	MuonLink
102	**2.2.3.9**	**High Level Muon Class**
103	2.2.3.9.1	Design
104	2.2.3.9.2	Implementation (CDFMuon)
105	2.2.3.10	Validation
106	**2.2.4**	**Calorimetry**
107	**2.2.4.1**	**Calorimetry**
108	2.2.4.1.1	Mods for geom, EDM, calibration
109	2.2.4.1.2	Integration/Optmization
110	**2.2.4.2**	**Jetclu**
111	2.2.4.2.1	Mods for geom, EDM, calibration
112	2.2.4.2.2	Integration/Optimization
113	**2.2.4.2.3**	**High Level Jet Class**
114	2.2.4.2.3.1	Design
115	2.2.4.2.3.2	Preliminary Implementation
116	**2.2.4.3**	**Emclst**
117	2.2.4.3.1	Mods for geom, EDM, calibration
118	2.2.4.3.2	Integration/Optimization
119	**2.2.4.4**	**Stpana**
120	2.2.4.4.1	Central strips
121	2.2.4.4.2	Plug Algorithm Development
122	**2.2.4.5**	**Electron**
123	2.2.4.5.1	Central Algorithm Development
124	2.2.4.5.2	Plug Algorithm Development



ID	WBS	Task Name
125	2.2.4.5.3	Integration/Optimization
126	**2.2.4.5.4**	**High Level Electron Class**
127	2.2.4.5.4.1	Design
128	2.2.4.5.4.2	Preliminary Implementation
129	**2.2.4.6**	**Metser**
130	2.2.4.6.1	Algorithm Development
131	**2.2.5**	**Level 3 Modules**
132	2.2.5.1	Management
133	**2.2.5.2**	**Reconstruction**
134	2.2.5.2.1	Regional Reco Framework
135	2.2.5.2.2	Regional COT tracking
136	2.2.5.2.3	Regional silicon tracking
137	**2.2.5.3**	**Framework**
138	2.2.5.3.1	Path/Prereq/Output
139	2.2.5.3.2	Monitors
140	2.2.5.3.3	DAQ interface
141	**2.2.5.4**	**Databases**
142	2.2.5.4.1	L3 Trigger Table
143	2.2.5.4.2	L3 Calibration
144	**2.2.5.5**	**Code Building**
145	2.2.5.5.1	Code management Tools
146	2.2.5.5.2	Build Tools
147	2.2.5.5.3	Build Platform
148	**2.2.5.6**	**Executable**
149	2.2.5.6.1	Null Exe
150	2.2.5.6.2	Non-null Exe
151	2.2.5.6.3	DAQ Integrated Exe
152	2.2.5.6.4	Final Benchmarking
153	**2.3**	**Event Data Model**
154	2.3.1	TRYBOS
155	**2.3.2**	**Event Data Model (CDFEDM2)**



ID	WBS	Task Name
156	**2.3.2.1**	**Management and Support**
157	2.3.2.1.1	coordinate software development
158	2.3.2.1.2	follow-up on build and validation failures
159	2.3.2.1.3	maintain web page
160	2.3.2.1.4	consult with users on CDFEDM2
161	**2.3.2.2**	**Documentation**
162	2.3.2.2.1	EDM for Novices
163	2.3.2.2.2	Basic Tools for EDM and Documentation
164	2.3.2.2.3	examples, datafiles, put into validation
165	2.3.2.2.4	revise and announce
166	2.3.2.2.5	programmer reference guide
167	2.3.2.2.6	maintain and test the guide
168	**2.3.2.3**	**Build Procedures**
169	2.3.2.3.1	Fix arch_spec_rootcint.mk
170	2.3.2.3.2	Adapt official arch_spec_root*.mk
171	**2.3.2.4**	**Optimize EDM2 for Sequential I/O**
172	2.3.2.4.1	Profiling and Benchmarking on MDC1 Data
173	2.3.2.4.2	Rework postread()
174	2.3.2.4.3	Speed up Object Selection
175	2.3.2.4.4	Use Memory Managers
176	2.3.2.4.5	Making postread() optional
177	2.3.2.5	Optimize EDM2 for Data Logger
178	2.3.2.6	Optimize EDM2 for Splitter
179	**2.3.2.7**	**Support I/O drop/keep lists in EDM2**
180	2.3.2.7.1	Initial minimal implementation
181	2.3.2.7.2	More general API
182	2.3.2.8	Support Object Compression in EDM2
183	2.3.2.9	List of Class Names in Data file in EDM2
184	2.3.2.10	Templated Find and Append Methods
185	2.3.2.11	indexedLink classes
186	2.3.2.12	Adapt EDM2 to new RCP System



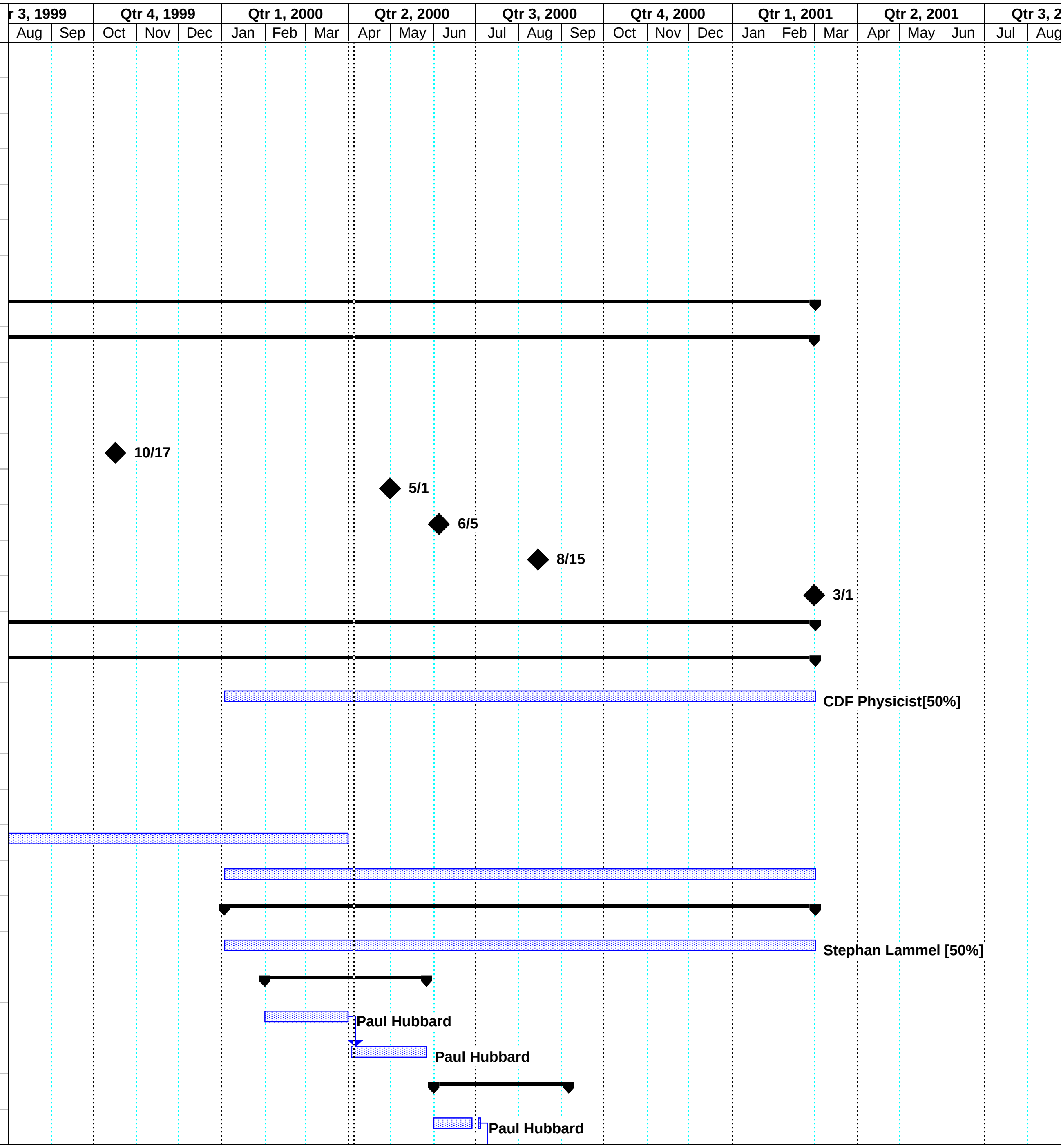
ID	WBS	Task Name	Resources
187	2.3.2.13	Adapt EDM2 code to use ErrorLogger	
188	2.3.2.14	Improve linking to new objects	
189	2.3.2.15	Improve robustness	
190	2.3.2.16	Improve error handling	
191	**2.3.3**	**ROOT Event Model**	
192	2.3.3.1	Implement first sequential event model	
193	2.3.3.2	ROOT data browsing/interaction	
194	2.3.3.3	first multi-branch event model	**Rob Kennedy**
195	2.3.3.4	Optimize multi-branch event model	**Rob Kennedy**
196	2.3.3.5	branches and class-branch assigments	
197	2.3.3.6	Split-Root Files	
198	**2.3.4**	**Event Data Format**	
199	**2.3.4.1**	**Management and Support**	
200	2.3.4.1.1	Determine which objects lack support	
201	2.3.4.1.2	definition and implementation of objects	
202	2.3.4.1.3	Consulting	
203	2.3.4.2	Header Banks/objects	
204	**2.3.4.3**	**Raw Data, Calibration, Pedestal Banks**	
205	2.3.4.3.1	CDF 4152 maintenance and support	**David Dagenhart**
206	2.3.4.3.2	Bank accessor functions	**David Dagenhart,Rich Glosson,Simona Rolli**
207	2.3.4.3.3	Bank simulation functions	
208	2.3.4.3.4	Bank channel map functions	
209	2.3.4.4	Trigger Banks	
210	2.3.4.5	Simulation objects	
211	2.3.4.6	Reconstruction objects	
212	2.3.4.7	Higher-Level Physics objects	
213	**2.4**	**Databases and Constants**	
214	2.4.1	Management	**Rick St. Denis,Mark Lancaster**
215	**2.4.2**	**Calibration DB**	
216	2.4.2.1	Shakedown + Document: release exaples	
217	**2.4.2.2**	**Table definitions (java) with det. Grps**	

ID	WBS	Task Name	Resource
218	2.4.2.2.1	Calorimeter	
219	2.4.2.2.2	COT	
220	2.4.2.2.3	SVX	
221	2.4.2.2.4	Muons	
222	2.4.2.2.5	Luminosity Monitor	
223	2.4.2.2.6	TOF	
224	2.4.2.2.7	CES	
225	**2.4.2.3**	**Calibration Hooks**	
226	2.4.2.3.1	Reconstruction	]
227	2.4.2.3.2	L3	]
228	**2.4.3**	**Conditions Databases**	
229	2.4.3.1	Development	Joint Project Request[50%]
230	2.4.3.2	Trigger	CDF Physicist[50%]
231	2.4.3.3	Detector Conditions	CDF Physicist[50%]
232	2.4.3.4	Luminosity	CDF Physicist[50%]
233	2.4.3.5	Beams	Paul Dervent
234	**2.4.3.6**	**Alignment**	
235	2.4.3.6.1	Develop Algorithms	CDF Physicist[200%]
236	2.4.3.6.2	Implement	CDF Physicist[200%]
237	2.4.3.6.3	COT	
238	2.4.3.6.4	SVX	
239	2.4.3.6.5	Muons	
240	2.4.3.6.6	CES	
241	**2.4.3.7**	**Constants**	
242	2.4.3.7.1	Develop Algorithms	CDF Physicist[200%]
243	2.4.3.7.2	Implement	CDF Physicist[200%]
244	2.4.3.7.3	COT	
245	2.4.3.7.4	SVX	
246	2.4.3.7.5	Muons	
247	2.4.4	Production/Integration/Development Databases	
248	2.4.5	Export of Database to Remote Institutions	

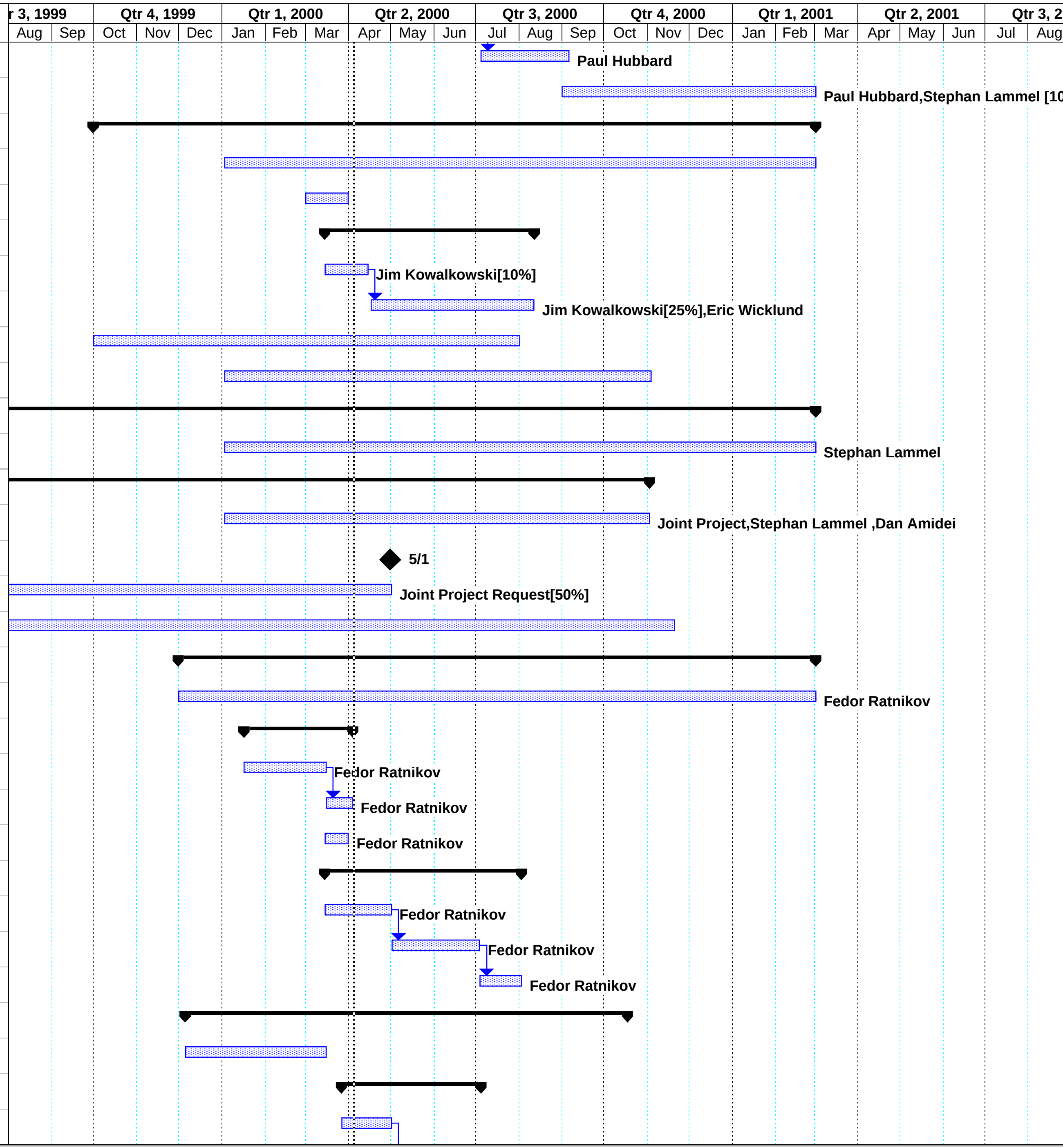
ID	WBS	Task Name	Resources
249	2.4.6	Database Administration	
250	2.4.7	Database Replication	
251	2.4.8	Database Backup and Restore	
252	**2.5**	**Event Display**	
253	2.5.1	Develop Proposal	
254	2.5.2	ROOT SVX/COT prototype	
255	2.5.3	OI SVX prototype in AC++	
256	2.5.4	Platform Design	
257	2.5.5	First Release	
258	2.5.6	Platform maintenance	CDF Physicist
259	**2.5.7**	**Integrate Plug-Ins**	
260	2.5.7.1	COT	CDF Physicist[25%]
261	2.5.7.2	SVX	CDF Physicist[25%]
262	2.5.7.3	Muons	CDF Physicist[25%]
263	2.5.7.4	Central Cal	CDF Physicist[25%]
264	2.5.7.5	Plug Cal	CDF Physicist[25%]
265	**2.6**	**Code and Configuration Management**	
266	**2.6.1**	**Release Management**	
267	2.6.1.1	CVS Repository management	CDF Physicist[50%],Joint Project[50%]
268	2.6.1.2	SoftRelTools	
269	2.6.1.3	Continuing Offline Reorganization	
270	2.6.1.4	Creation of frozen releases	
271	2.6.1.5	New Flavors	
272	**2.6.2**	**Distribution Management**	
273	2.6.2.1	Field Support	Joint Project
274	2.6.2.2	UPS,UPD, roll into IRIX	
275	2.6.2.3	User Access and Registration	
276	2.6.2.4	New Flavors	
277	**2.6.3**	**Validation Management**	
278	2.6.3.1	Validation of Frozen Releases	CDF Physicist[50%]
279	2.6.3.2	Daily Error Reporting	

ID	WBS	Task Name	Resources
280	2.6.3.3	Improve depth of checking	CDF Physicist
281	2.6.3.4	New Flavors	
282	**2.7**	**Analysis Framework**	
283	2.7.1	Generally	CDF Physicist[50%],Joint Project Request
284	**2.7.2**	**FrameMods**	
285	2.7.2.1	F77Interface	
286	2.7.2.2	APPFileInputModule	
287	2.7.2.3	APPDataSetInputModeule	
288	2.7.2.4	APPGenInputModule	
289	2.7.2.5	APPLevel3InputModule	
290	**2.7.2.6**	**APPFileOutputModule**	
291	2.7.2.6.1	Bank List Control	
292	2.7.2.6.2	Connection to Filters	
293	2.7.2.7	APPVegasInputModule	
294	2.7.2.8	APPMixInputModule	
295	**2.7.3**	**Utility Modules**	
296	2.7.3.1	InitModule	
297	2.7.3.2	Framework Bank Dump	
298	2.7.3.3	Error Logger	
299	2.7.3.4	Calibration Manager	
300	2.7.3.5	GeometryManager	
301	2.7.3.6	Histogram Manager	
302	**2.7.4**	**MenuHandler**	
303	2.7.4.1	Basic classes implemented	
304	**2.7.4.2**	**Interpreter Interface**	
305	2.7.4.2.1	GUI	
306	**2.7.5**	**Filtering**	
307	2.7.5.1	enable/disable by path	
308	2.7.5.2	connect path<-->stream	
309	2.7.5.3	statistics	
310	2.7.6	Run-time Module cloning	

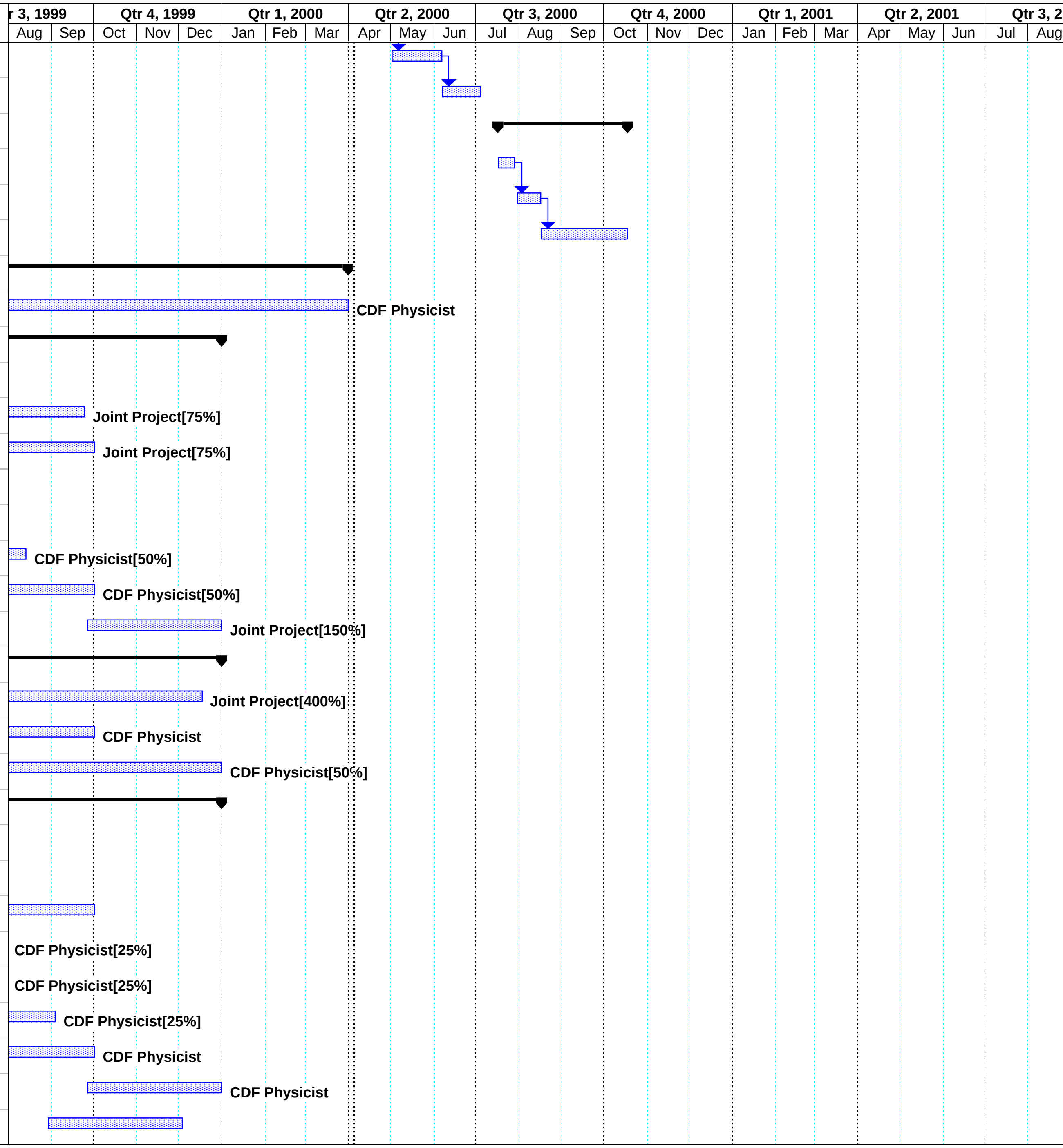
ID	WBS	Task Name
311	2.7.7	Begin-run Begin-File behaviour
312	2.7.8	Processing statistics
313	2.7.9	Persistant Processing History
314	2.7.10	Automatic optional logging of Absparm
315	**2.7.11**	**Histograms**
316	2.7.11.1	ntuple example
317	2.7.11.2	Fortran API implemented
318	**2.8**	**Data Handling**
319	**2.8.1**	**Milestones**
320	2.8.1.1	October Demonstration
321	2.8.1.2	Analysis Prototype available
322	2.8.1.3	Mock Data Challenge 1
323	2.8.1.4	Testing for Mock Data Challenge 2
324	2.8.1.5	Mock Data Challenge 2
325	2.8.1.6	Engineering Run
326	2.8.1.7	First Data
327	**2.8.2**	**Resource Management**
328	**2.8.2.1**	**Batch System**
329	2.8.2.1.1	Management
330	2.8.2.1.2	Prototype
331	2.8.2.1.3	Batch Requirements
332	2.8.2.1.4	Choose Batch implementation
333	2.8.2.1.5	Batch Customization
334	2.8.2.1.6	Batch Integration
335	**2.8.2.2**	**Disk Inventory Manager**
336	2.8.2.2.1	Management
337	**2.8.2.2.2**	**Basic**
338	2.8.2.2.2.1	Design
339	2.8.2.2.2.2	Code and Debug
340	**2.8.2.2.3**	**Full version**
341	2.8.2.2.3.1	Design



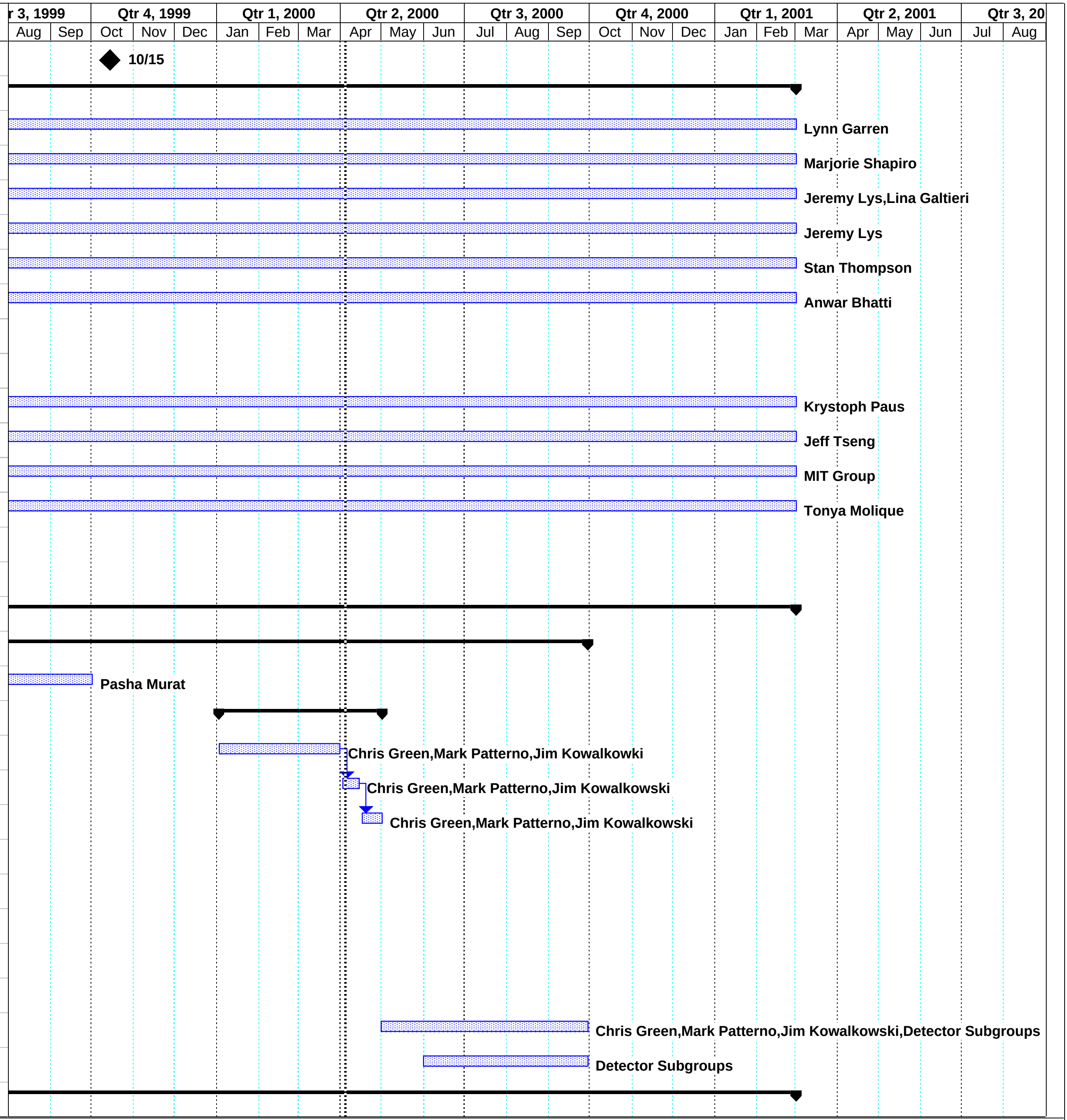
ID	WBS	Task Name
342	2.8.2.2.3.2	Code and Debug
343	2.8.2.2.4	Integration
344	**2.8.3**	**CDF Data Catalog**
345	2.8.3.1	Manage
346	2.8.3.2	MDC1 Fixes
347	**2.8.3.3**	**Tables**
348	2.8.3.3.1	Finalize Tables
349	2.8.3.3.2	Finalize Implementation
350	2.8.3.4	Dataset Browsing
351	**2.8.3.5**	**Operations and Maintenance**
352	**2.8.4**	**Mass Storage**
353	2.8.4.1	Management
354	**2.8.4.2**	**Tape systems**
355	2.8.4.2.1	Serial Media Working Group
356	2.8.4.2.2	First Serial Media RFP
357	2.8.4.2.3	Tape software(OCS, mt-tools, FTT, daemon)
358	2.8.5	Analysis Systems
359	**2.8.6**	**I/O (DHMod)**
360	2.8.6.1	Management
361	**2.8.6.2**	**Input Module**
362	2.8.6.2.1	Input Module Implementation
363	2.8.6.2.2	Input Module Tests
364	2.8.6.2.3	User Guide
365	**2.8.6.3**	**Output Module**
366	2.8.6.3.1	Design
367	2.8.6.3.2	Implmentation
368	2.8.6.3.3	User Guide
369	**2.8.6.4**	**Operations**
370	2.8.6.4.1	MDC1 Operations
371	**2.8.6.4.2**	**Mock Data Challenge 2**
372	2.8.6.4.2.1	Preparation



ID	WBS	Task Name
373	2.8.6.4.2.2	Testing
374	2.8.6.4.2.3	Operations
375	**2.8.6.4.3**	**Engineering Run**
376	2.8.6.4.3.1	Test Data Path
377	2.8.6.4.3.2	Rate Test
378	2.8.6.4.3.3	Operations
379	**2.9**	**Production Systems (Farms)**
380	2.9.1	Management
381	**2.9.2**	**Hardware prototype**
382	2.9.2.1	Establish
383	2.9.2.2	Maintain
384	2.9.2.3	Connectivity
385	2.9.2.4	Specify I/O hardware
386	2.9.2.5	Specify Worker Node Acquisition
387	2.9.2.6	Integrate tape I/O
388	2.9.2.7	Integrate farms + I/O control
389	2.9.2.8	Integration of worker nodes
390	**2.9.3**	**Farm Control**
391	2.9.3.1	Establish/Run/Maintain FBS
392	2.9.3.2	I/O Control within farm
393	2.9.3.3	Output control code
394	**2.9.4**	**CDF production**
395	2.9.4.1	Run CDF Production Executable
396	2.9.4.2	Multiple Executable with FBS
397	2.9.4.3	Farms Executable prototype
398	2.9.4.4	Calibration acces
399	2.9.4.5	Trigger DB access
400	2.9.4.6	CDF catalog access
401	2.9.4.7	Prototype version
402	2.9.4.8	First release
403	2.9.4.9	Monte carlo production



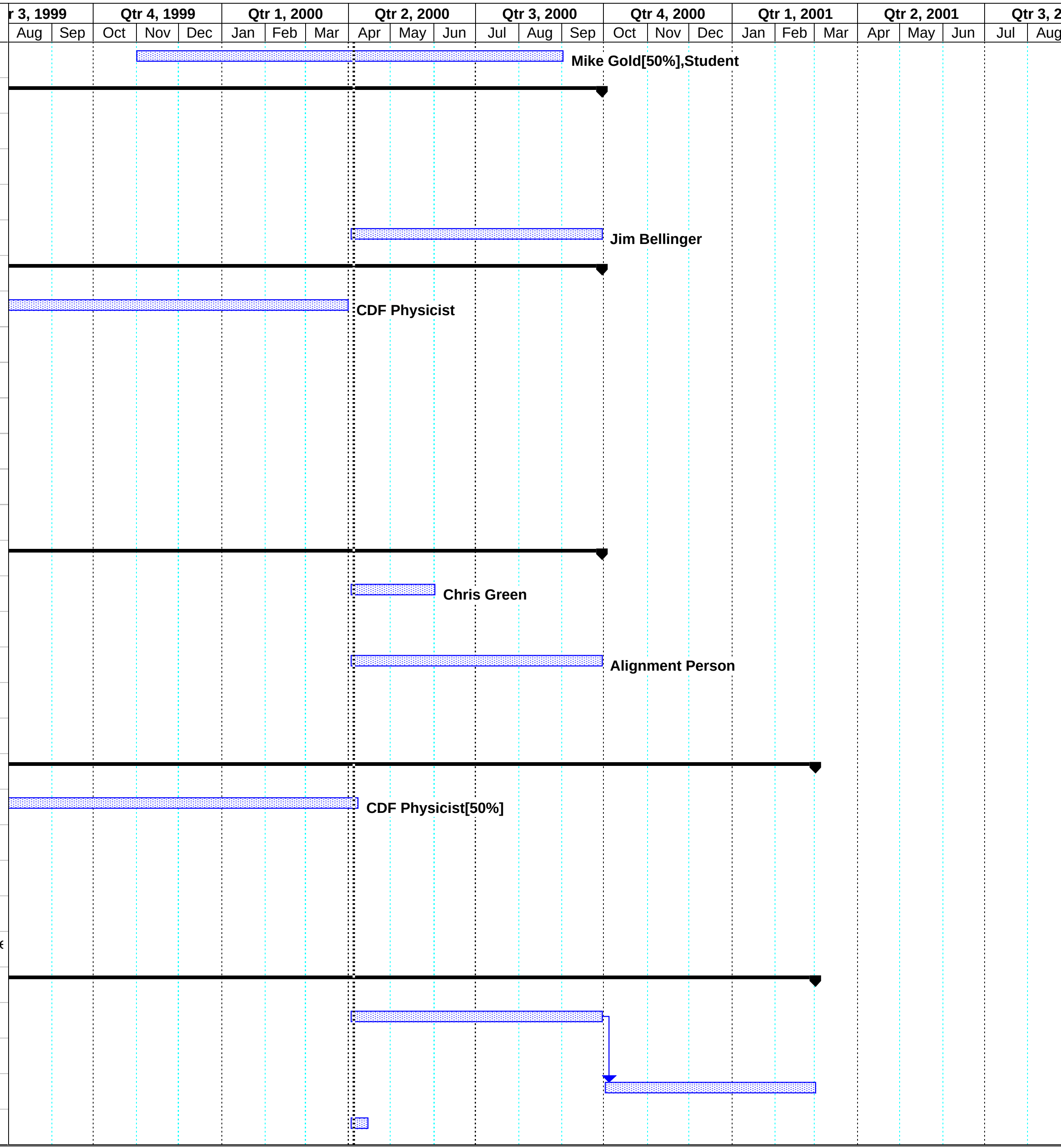
ID	WBS	Task Name
404	2.9.4.10	Mock data challenge
405	**2.10**	**Event Generators**
406	2.10.1	StdHep
407	2.10.2	Fakev
408	2.10.3	Isajet
409	2.10.4	Herwig
410	2.10.5	Pythia
411	2.10.6	MBR
412	2.10.7	Vecbos
413	2.10.8	DyRad
414	2.10.9	Bgenerator
415	2.10.10	QQ
416	2.10.11	EVTGen
417	2.10.12	Tauola
418	2.10.13	Wgrad
419	2.10.14	Herprt
420	**2.11**	**Detector Simulation**
421	**2.11.1**	**Infrastructure**
422	2.11.1.1	Old
423	**2.11.1.2**	**New**
424	2.11.1.2.1	Geant Initialisation using New Geometry
425	2.11.1.2.2	Prototype with 1 subdetector
426	2.11.1.2.3	completion of new simulation infrastructure
427	2.11.1.3	Random Numbers
428	2.11.1.4	Geometry database
429	2.11.1.5	Physics Param Tools
430	2.11.1.6	Track Embedding
431	2.11.1.7	API Digitization Framework
432	2.11.1.8	Digitization Rewrite
433	2.11.1.9	Bank Creation Routines
434	**2.11.2**	**Calorimeter**



ID	WBS	Task Name	
435	**2.11.2.1**	**user interface**	
436	2.11.2.1.1	talk to	**Chris Green,Liz Sexton**
437	**2.11.2.2**	**I/O**	
438	2.11.2.2.1	CPRD made Run 1 format)	**ti,Pierre Savard,Michael Riveline**
439	2.11.2.2.2	PPRD made	**KEK**
440	2.11.2.2.3	PESD made	**zuki**
441	2.11.2.2.4	D banks for Trigger Simulation	**rd**
442	**2.11.2.3**	**GFLASH**	
443	2.11.2.3.1	generally	**ard**
444	**2.11.2.3.2**	**validation**	
445	2.11.2.3.2.1	Cental EM scale	
446	2.11.2.3.2.2	Central Hadron Tuned (high energy)	
447	2.11.2.3.2.3	Plug EM tuned	
448	2.11.2.3.2.4	Plug Hadron Tuned (High Energy)	
449	**2.11.2.4**	**QFL**	
450	2.11.2.4.1	Cental EM scale	
451	2.11.2.4.2	Central Hadron Tuned (high energy)	
452	2.11.2.4.3	Plug EM tuned	
453	2.11.2.4.4	Plug Hadron Tuned (High Energy)	
454	2.11.2.5	Detailed central, plug in Geant module	**CDF Physicist[50%]**
455	**2.11.2.6**	**Geometry**	
456	2.11.2.6.1	Update Coarse Geometry (GFLASH)	**Pierre Savard**
457	2.11.2.6.2	Update Detailed Geometry (GEANT)	**John Mayer**
458	2.11.2.7	Miniplug	**Lena Gerchtein**
459	**2.11.3**	**Central Muons**	
460	2.11.3.1	generally	**CDF Physicist[40%],George Velev**
461	2.11.3.2	Muon Steel Geometry	
462	2.11.3.3	Magnetic Field tuned	
463	2.11.3.4	Validate Stub quality	**Prem Singh**
464	2.11.3.5	Test track-stub matching	**Prem Singh**
465	2.11.3.6	Update geom to DetectorModel	**Mike Gold**

Timeline: r 3, 1999 (Aug, Sep) · Qtr 4, 1999 (Oct, Nov, Dec) · Qtr 1, 2000 (Jan, Feb, Mar) · Qtr 2, 2000 (Apr, May, Jun) · Qtr 3, 2000 (Jul, Aug, Sep) · Qtr 4, 2000 (Oct, Nov, Dec) · Qtr 1, 2001 (Jan, Feb, Mar) · Qtr 2, 2001 (Apr, May, Jun) · Qtr 3, 20 (Jul, Aug)

ID	WBS	Task Name
466	2.11.3.7	fix bugs in digitization
467	**2.11.4**	**Intermediate Muons**
468	2.11.4.1	Fortran Version
469	2.11.4.2	Magnetic Field Map
470	2.11.4.3	Study Stub quality
471	2.11.4.4	Move geometry to C++
472	**2.11.5**	**Silicon**
473	2.11.5.1	generally
474	**2.11.5.2**	**User interface**
475	2.11.5.2.1	Talk-to submenu
476	**2.11.5.3**	**signal simulation**
477	2.11.5.3.1	Tune cluster widths(Run1 cluster sim)
478	2.11.5.3.2	Add noise
479	2.11.5.3.3	Z-clusters
480	**2.11.5.4**	**Geometry**
481	2.11.5.4.1	Update geom to DetectorModel
482	2.11.5.4.2	Interface to Geometry DB
483	2.11.5.4.3	misalignment hooks
484	**2.11.5.5**	**I/O**
485	2.11.5.5.1	MSVX bank creation
486	**2.11.6**	**COT**
487	2.11.6.1	generally
488	**2.11.6.2**	**user interface**
489	2.11.6.2.1	talk-to
490	**2.11.6.3**	**Signal simulation**
491	2.11.6.3.1	Physics effects: hit res, afterpulsing, noiuse, de
492	**2.11.6.4**	**Geometry**
493	2.11.6.4.1	fix hit generation (change from fortran to C++)
494	2.11.6.4.2	Put correct material in active COT
495	2.11.6.4.3	wire sag
496	2.11.6.4.4	outer cylinder



ID	WBS	Task Name	Resources
497	2.11.6.4.5	Geometry DB and Misalignments	
498	2.11.6.4.6	unified geometry	
499	2.11.6.4.7	misalignment hooks	
500	2.11.7	TOF	Mathew Jones,Penn Group
501	2.11.8	CLC	Florida Group,Darin Acosta
502	**2.11.9**	**Validation**	
503	2.11.9.1	Validation	Pasha Murat
504	**2.11.10**	**Documentation**	
505	2.11.10.1	Geometry Classes	
506	2.11.10.2	Users Guide	
507	2.11.10.3	Validation note: Calorimeter	
508	2.11.10.4	Validation note: Central Muons	
509	2.11.10.5	Performance note: Run 2 Tracking	
510	**2.12**	**Trigger Simulation**	
511	2.12.1	Management	Simona Rolli[50%]
512	2.12.2	XFTsim	Brian Winer[50%]
513	2.12.3	SVTsim	
514	2.12.4	Calsim	]
515	2.12.5	Cmusim	CDF Physicist[50%]
516	2.12.6	xcesim	
517	2.12.7	L1 Global and L2 Global	
518	2.12.8	XTRP	
519	2.12.9	Parser	CDF Physicist[50%]
520	2.12.10	Database Access	CDF Physicist[50%]
521	2.12.11	Validation	CDF Physicist[200%]
522	**2.13**	**Documentation and Training**	
523	2.13.1	Coding Standards	
524	2.13.2	Infrastructure	
525	2.13.3	Documentation	CDF Physicist

4 Production Executable

left to be done 3/00. getting all code to talk to database. all detectors need to be aligned. Need alignment infrastructure. Error logging and error handling.Request from production farms to terminate with non-success code on failure. Farms operations items. Global. Local error handling when a module doesn't get what it needs. Only Chris Green and Rick are using error logger. May need a histogram manager.

13 Error Logging and Handling

Need to incorporate into reconstruction modules.
Need policy for passing error codes and for error severity.
One page guide.
Change cout statements to error logger statements.

14 Policy and Guide for Error Logging

This is both the policy for passing error codes, and the guide for which severity to use in logging, and a how-to for coding in the error logger/handler for users.

17 Example of Talking to Database

Done. Calorimetry and CMX.

18 Use by Tracking, Muons, Calorimetry, etc.

Wanted this for MDC2.
The online groups are going to drive us on this?
We must have this for engineering run.
Peter Tamburello for COT.
Need for silicon and beamline.
COT, silicon, beamline, muons, calorimetry needs to be finished,
API will be used by Geometry, but wont get data from oracle.
Geometry DB is a root file that does not exist.

21 Definition of Infrastucture and Constants

need database tables, want to use calibration database API.
Define set of constants: specify the alignment content and tables detector by detector.

25 Compliers and Debuggers

Ongoing. supposed to be examining the next gnu compiler in the spring.
Pressure from university groups that don't want to buy kai. Not high priority.
Using Totalview. sufficient but not wonderful.

28 Use of ISOCXX portability package

Substitute ISOCXX portability packag for defect macros.
Needed before we move to another compiler. Can grep for defect and replace with standard c++ and include header file from ISOCXX package.
Prerequisite for a new compiler.

31 Formatted Bank Dump

Done.

32 Build Job++

Need a student to do this, but Liz thinks it will take too much of her time to direct a student to do it. Could use some low level C++ programmer for this.
Previous version was written in Python. Could be done in a month if we can salvage some of the old Python stuff.

33 Zoom: heptuple, error log

Done. Products in maintenance phase.
Miscellaneous complaints about HepTuple from people not using the problem correctly. API is clumsy is a common complaint.
Some people would rather use Root. Half paw and half Root and paw people need HepTuple.

36 Design/implementation

Linked to the use of the new Geometry in simulation.

37 Database/Persistence

Not done. Root file.

38 Extrapolation Interface

Driven off of Geometry.
Interface for the extrapolation of tracks to various detectors.
We have extrapolator class but it needs rethinking. Need generic interface for users, but need detector specific interfaces for efficiency. Simple classes for specific easy extrapolations.

41 Prototype production table

Work has not begun 3/15/00.
List of modules, parameters, prerequisites, etc.
Level 4 Trigger Table.

42 Prototype Input validation control

How do we know that calibration constants are OK.
Valid set is ready and OK. Automated procedure by which production job

on the farm can determine that it has all the required inputs and they are all correct. Don't want to be able to run with stale constants. Neet t make sure this has been completed (contact Mark Lancaster).

51 Basic Design
Done.

52 Remake for EDM design
Done.

53 CDFTrack in new EDM
Done.

54 Re-implement CDFTrack persistency
Done write out TRCK and TRCL banks anymore.
R. Snider really wants this done.
Rick may do it, or Dave Waters.

56 Fast Event Vertex
Done.
Z-vertex from z-position of tracks along beamline.
XY-vertex from beamline.
Basically done, something that runs, could still use continued validation.
Dirk Neuberger at Karlsruhe was doing this. But not doing this now.
QPV is what it is called. May not be adequate.
Questions of efficiency and track multiplicity.
Want to make code to use tracks as well as silicon stuff for vertex.
VXPRIM is overkill, need faster method using tracks.
Roughly a month of work. Needed for start of run.
Karlsruhe responsibility. Patrick Schemitz is possible working on this.
Michael Feindt is chief. Rick thinks there not working on this.
Hans Wenzel was working on VXPRIM. Rinnert in group. New student.
Thomas Mueller may be the person to talk to, more active than Feindt.
Dave Gerdes will call.

57 VXPrim 3D Vertex
Not done as of 3/8/00.
Three dimensional vertex that uses XYZ of tracks.
responsibility of Karlsruhe group as of 3/8/00.
They wrote something but it is not integrated into the system.
Student wrote code. might only be a month of work.

58 Generic Vertex Fitting
According to Rick Snider.
Framework for either primary or secondary vertex fitting with various algorithms. For example, CTVMFT could be used within Generic Vertex Fitting. Needs to be ready at start of run.
CTVMFT has been wrappered. Rick has bigger idea of a framework.
B group wants it. Not necessary for CTVMFT.

59 constrained mass fitting (CTVMFT)
wrappered. how up-to-date is the interface? does it take CDFTrack.
Joe Boudreau, Petar Maksimovic. Need this for MDC2 if we have a level 3 trigger that needs this.

62 Standalone algorithm development
Done.

63 Reorganize geometry
Basically Done according to Rick Snider.
Not done according to Liz Sexton.

64 Alignment Hooks
Red Pen

65 Material Integrator
Not done because it requires initialisation of new geometry in simulation (geant).

68 Reorganize geometry
Done. Peter Tamburello.

69 R-phi Algorithm development
Mature for one algorithm. Aseet tracking.
2nd algorithm needed. A lot of work has been done.
May be integrated tracking approach: if so a lot of infrastructure exists.
Paolo has a prototype model.
Getting a final list of COT tracks based on multiple agorithms has not been done and is potentiall a lot of work.
2nd algorithm needed for Engineering run.

70 Stereo algorithm development
We don't know if it is done, needs more validation.
Stereo is working really quite well. Acceptable performnace.

71 Port Run I fitter
Bill Ashmanskas and Peter Tamburello are supposed to be working on this.
Rob Snihur at Rutherfoord is working on this.
Needed for a long time. Promised a long time ago.
Needed for Engineering run. Work has stalled. P. Tamburello needs to be gotten together with Snihur. Dave Gerdes will try to do this.

72 Optmization
Ken Bloom is working on this.
Speed Optimization. Going on continually at low level. OK for MDC2.
Sufficiently optimized for engineering run and run 2.

74 Integrate outside-in code
Done. Ken Bloom

75 Integrate silicon z-hit finding
Pattern Recognition is being done, but hits are currently not being used in the last pass of fit. Adding hits to fit is weeks work for ken.
Done by Arpil 1?

76 Full integration & testing
Not done. Dependent on integrate silcon z-hit finding.
Ken Bloom and Paolo Gatti and Patricia Azzi.

78 multi-algorithm, multi-fit, integration
Big job.
6 months of work.
nobody is commited to this.
Got to be there for beginning of run.

79 picking the best track for a set of hits
not started as of 3/9/00.
could be a lot of work. not a lot known.
Tracking Group wide issue.
Milestones for MDC2, engineering run ,start of run needed.

80 multiple fit infrastructure (CDFTrack needs list of fits)
Linked to completion of "Port run 1 fitter" which should be changed to final fitter. Start using it when final fitter is completed. Same end dates.
A lot of work. Track interface add fit method for CDFTrack, etc.
Could take 6 months. Strong religious opinions on this one.
Tracking Group wide issue, and Level 3 algorithms people.
Joe Boudreau agreed to wite a fit class a while ago, so he is a candidate.

85 E to S in C++
Dan Eisenberg is an undergrad from Illinois. Will be working 1/4 time on this with Prem. Will become more than 1/4 time during summer break but the average will be 1/4. Most work will start in May when he is based here at Lab.

90 D to E
25% FTE from Giovanni Paoletta's group.

100 Implementation
Done

101 MuonLink
July onwards, if there is any work left, Michael Schmidt has a postdoc with 25% FTE to work on this project.

154 TRYBOS
Still in use, but package is in maintenance-only mode. Done.

155 Event Data Model (CDFEDM2)
policies, software defining how event data is manipulated

157 coordinate software development
coordinate software development related to CDFEDM2
ongoing. Rob K., help from Liz S-K, on-going

158 follow-up on build and validation failures
ob K., help from Liz S-K and others, on-going

159 maintain web page
ongoing

160 consult with users on CDFEDM2
Rob K., help from Liz S-K, Rick S, Chris G, on-going

162 EDM for Novices
write first draft of EDM-for-novices note, post for comments by mgmt and EDM-WG, but do not announce yet
* Rob K., began 3/1/00, complete 3/24/00, current task

* Edm-for-novices note is Second Top Priority as of 3/23/00

163 Basic Tools for EDM and Documentation
create basic set of tools for CDFEDM2 and document
- in conjunction with EDM-for-novices notes
* Rob K., began 3/1/00, complete 3/24/00.

164 examples, datafiles, put into validation
incorporate examples, datafiles, put into validation
- in conjunction with EDM-for-novices notes
* Rob K., David Dagenhart, Ken Bloom, and others
* began 3/1/00, complete 4/1/00,

165 revise and announce
later: revise note per comments, fill in missing
pieces, test tools on all OSs, announce publicly
(This will become "maintain EDM-for-novices on all OSs")
* Rob K., begin 4/1/00, on-going from then

166 programmer reference guide
Rob K., begin post-MDC2. Roughly 7/1/00

167 maintain and test the guide
maintain programmer reference guide, test on all OSs
* Rob K., begin after creation, on-going from then

169 Fix arch_spec_rootcint.mk
Fix arch_spec_rootcint.mk to handle indirect depends
* Rob K., Liz S-K, input from Jim Amundson, began 3//00
* Liz S-K to test proposed fix by 3/24/00 after minimal drop
* I/O list task completed, release by 4/1/00
* Release of EDM optimizations is waiting on this
* Fixing rootcint.mk is Third Top Priority at present.

170 Adapt official arch_spec_root*.mk
Adapt official arch_spec_root*.mk to ease transition by
some away from RootObj variants of arch_spec files
* Rob K., input from Pasha Murat, began 3/1/00
* Release adaptation after indirect depends fixed

171 Optimize EDM2 for Sequential I/O
Optimize EDM2 implementation for sequential I/O

172 Profiling and Benchmarking on MDC1 Data
use profiling, benchmark programs on MDC1 data to find
more bottlenecks, "slow code", inefficient algorithms
* Rob K., Philippe Canal, Rick S., Chris G., and others
* began 3/1 when local MDC-1 data samples used
* on-going process, need MDC-2 goals to be met
* Strictest goal appears to be 20 MB/sec in Splitter

173 Rework postread()
rework postread() to use an object id-to-address map
* Rob K., begin 4/15, complete 4/22
* Should yield large reduction in CPU during post-read.

174 Speed up Object Selection
use more sophisticated internal event data structure to
speed up object selection, iteration, dropping objects
* Rob K., begin 4/22, complete 5/1
* Should yield moderate reduction in CPU used.

175 Use Memory Managers
use memory managers once available
* Rob K., interest shown by Rick S. also
* requires ZOOM or Jim K/Marc P to produce memory mgrs
* desirable to begin testing in late April

176 Making postread() optional
consider making postread() calls optional or on-demand
* Rob K., unknown begin and completion dates
* this may not be req

177 Optimize EDM2 for Data Logger
Optimize EDM2 for Data Logger = Write(TBuffer)
- Provides desired alternative to format XXX without
- sacrificing rate in Data Logger (writes BLOB to TBranch)
* Rob K., begin 3/23 when Philippe Canal releases RH5.X version
* version of ROOT with this capability (RH6.1 exists)
* I expect 1-2 weeks for completion of tests and integration.
* Write(TBuffer) is First Top Priority at present.

178 Optimize EDM2 for Splitter
- Optimize EDM2 for Splitter = Read(TBuffer), etc.
- Allows splitter to read most of event without allocating
- memory for individual objects in event. Coupling with the use
- multi-branch events, this may be needed for MDC-2.
* Rob K., begin when Philippe Canal/ROOT team releases version
* of ROOT with this capability. I expect 1-2 weeks of testing,
* integration after that. More difficult than Write(TBuffer).
* Desirable, but not required, for MDC-2
# 3-23-2000: Discussed with Miro. Tentative estimate for
availability is end of April, depending on delivery of feature by ROOT team. He
is eager to apply this to the Farms Concatenator program which does not need multiple branches (all information needed is external to event), but has a more severe I/O requirement since there is only one process treating all the data. Adding Read (TBuffer)/Write(TBuffer) functionality to the existing Sequential Root event model should yield a great improvement in the Concatenator's processing speed. Because it is simpler and applies to a more severe I/O requirement, this should be tried before exploration of multiple branches and the more complicated mixed Read(TBuffers)/Read(TBranch), which means read and interpret one TBranch while only reading into TBuffers the byte streams for other TBranches, as may be done in the Farms Splitter program.

180 Initial minimal implementation
Liz S-K, began about 3/15, completion 3/24

181 More general API
more general API and sophisticated implementation using
optimized internal event data structure
* Rob K. and Liz S-K, begin sometime after MDC-2,
* Completion in about one week, but depends on completion of RCP integration.

182 Support Object Compression in EDM2
Extend EDM2 to support object compression in various modes
- This is a broad topic, including the appproach to summary objects, DATSQZ-like approaches, and ROOT branch compression.
* Rob K., Rick S., and others interested. First need to define an overall plan as to what will be supported. Unknown begin and completion dates. Not needed for, but may be desired by, MDC2. Sometime in early April, we should have Henry Frisch's High-Level objects group, EDM working group, and other interested parties (Pierre, Avi, ...) to flesh out this topic.

183 List of Class Names in Data file in EDM2
Extend EDM2 to maintain list of class names in data file.
- Class name list is saved in a separate branch in data file.
* Rob K. May be only a client task of EDM, or may require
modifications to the EDM in order to be CPU efficient.
Depends on completion of first multi-branch data file class.
Should interface with the Production database in
space-efficient fashion. Unknown begin and completion dates.
Desirable, but not required, by Run2.

184 Templated Find and Append Methods
Extend EDM2 to support templated find and append methods
- Permits class specific information to be used in find.
* Rob K., unknown begin and completion dates.

185 indexedLink classes
Extend EDM2 to support "indexedLink" classes which provide
- persistent pointers to Streamable (rather than Storable)
- Object classes within Storable Containers
* Initial prototypes by Armin Reichold and Pierre Savard
* General case integrated into EDM by Rob K.
* Begin 3/20, no specific completion date yet.
* Desirable, but not required, before MDC-2.

186 Adapt EDM2 to new RCP System
adds ability to drop/find by creating module, parameter set
* Liz S-K and Rob K., should take only a few days to complete
the EDM-specific portion of this task.

187 Adapt EDM2 code to use ErrorLogger
Adapt EDM2 code to use ErrorLogger (rather than std::cerr)
* Does not require an expert for most of the work,
only for final tweaking of error reporting levels.
unknown begin date, should only take a few expert days.

188 Improve linking to new objects
Improve linking to new objects in production output stream
- General utilities now have no automatic way to recognize
new objects put out by production unless hand editting done.
* Desirable for Run2

189 Improve robustness
Improve robustness, close loopholes (esp. collections)
* Rob K., on-going work, a few specific instances need to be
* A few known cases now which may need to be looked into
* before examples may disappear in development

190 **Improve error handling**
Improve error handling, adapting ROOT errors to ErrorLogger
* Rob K., with some input from ZOOM w.r.t ErrorLogger code
being compatible with ROOT error-handling scheme.
* Perhaps we can request the Zoom work begin after ROOT-related
optimization work begins to slow down.
* Desirable, but not required, for MDC-2.

191 **ROOT Event Model**
organization of objects in event on disk

192 **Implement first sequential event model**
Rob K., COMPLETED

193 **ROOT data browsing/interaction**
Explore usefulness of ROOT data browsing/interaction and
implications on event model, CDFEDM2 code, and rootcint.
Is an n-tuple look-and-feel data file possible with CDFEDM2?
* Liz S-K and Philippe Canal to look into this soon.
* Not required for MDC-2.

194 **first multi-branch event model**
Propose, implement first multi-branch event model
* Rob K., begin soon, may take one to two weeks to implement
* This may be needed by MDC2 to improve rate in Splitter

195 **Optimize multi-branch event model**
Optimize multi-branch event model for production use
* Rob K., unknown begin date, estimate 2 weeks to implement
* Not required for MDC-2, should be done before Engineering Run

196 **branches and class-branch assigments**
Create mechanism for groups to define their own branches and
class-branch assignments, and record these as meta-data
* Rob K. and others, with input from physics groups and DH.
* Not required before Run II begins, but desirable

197 **Split-Root Files**
Determine how/where we can take advantage of Split-Root files
* Generally unknown how well we can match our DH system with
this approach to treating ROOT data. ROOT team knows about
this and is open to suggestions to improve our ability to
co-ordinate files containing differents pieces of the same
events given our DH system may deliver files "out-of-order".
* Not required before Run II begins, but desirable

198 **Event Data Format**
format of data in objects

200 **Determine which objects lack support**
Determine which objects lack support and are needed,
when, and by whom
* Rob K., David Dagenhart, On-line and Off-line managers
* needed as soon as possible

201 **definition and implementation of objects**
Co-ordinate definition and implementation of objects
* Rob K., David Dagenhart, On-line and Off-line managers
* needed by Engineering Run

202 **Consulting**
Consult on implementation related to CDFEDM2, C++, etc.
* Rob K., David Dagenhart, Rich Glosson, David Waters,
* and others. Effort is on-going.

203 **Header Banks/objects**
Header Banks/objects (LRIH, EVCL, etc.)
* Rob K., Liz S-K, and Rick S.
* Run I banks classes done, new LRIH done
* Major overhaul of what was stored in EVCL bank expected,
* the details have not been worked out, nor a person assigned.

204 **Raw Data, Calibration, Pedestal Banks**
(D, C, P banks)

205 **CDF 4152 maintenance and support**
David Dagenhart, on-going, with slow-down by 5/12

206 **Bank accessor functions**
* David Dagenhart, Rich Glosson, and Simona Rolli (for Trigger banks)
* began in pre-history, completed by 5/12

207 Bank simulation functions
- Simplify Simulation's task of filling raw data banks
- Includes add_block() and commit() methods
 * David Dagenhart has installed constructor(array)
 * Related to adapting Simulation software to use
 * StorableBanks (now use Generic Banks for some banks).
 * unknown who will complete this

208 Bank channel map functions
- Map logical id's to readout element or detector element
- May be should be separated from Banks classes and
  treated something like Geometry with database behind it.
* David Dagenhart documented in CDF 4152
* Calorimeters uses CalorKey, but other detectors?
* unknown who will implement this

209 Trigger Banks
Trigger Banks (T banks, trigger readout banks)
* Simona Rolli (L1/L2) and Kevin McFarland (L3)
* Some work is progressing in parallel with raw data banks
* For status, see WBS Section 2.12: Trigger Simulation and
  WBS Section 2.2.5: Level 3 Modules
* Some needed by MDC-2, others no later than Engineering Run

210 Simulation objects
Simulation objects (HEPG, OBSP, OBSV, M banks)
* Pasha Murat
* First version of nearly all Simulation ('M' banks) now
* consistent with Offline.
* Missing StorableBank classes for MCOT and MVTX.
* For status, see WBS Section 2.11: Detector Simulation
* Needed by when???

211 Reconstruction objects
Reconstruction objects (Track, Jet, Electron, Photon, Muon)
* Rick S., Pierre S., Muon Group
* Most already exist, improvement and support on-going
* For status, see WBS Section 2.2: Event Reconstruction
* Needed now for algorithm development

212 Higher-Level Physics objects
* Henry Frisch and Simona Rolli
* First prototypes appearing in Offline.
* For status, see WBS Section ???
* Needed into Run2

218 Calorimeter
Done. needs some fixes, a few tables.
1 1/2 months of work.

219 COT
3 tables done. Online tables defined.

220 SVX
table defined: need more. S. nahn. monitoring and offline needs definition.
Dead Channel lists an issue. 1 1/2 months.

221 Muons
7 tables defined. 4 out of 7 being used. David Dagenhart and Prem did this.

226 Reconstruction
Exists for Calorimeter, CMX, COT but not SVX.

227 L3
Done in principle, but there are bugs.

233 Beams
Task should take a month.

234 Alignment
Nobody currently in charge as of 3/22/00.
Workshop is being planned for april to organize this activity.

241 Constants
Examples are tracking t0, stage0.
Nobody in charge here either.

330 Prototype
Done

331 Batch Requirements
Done.

332 Choose Batch implementation
Done, LSF.
337 Basic
Basic functions include c/s, daemons/chrons.
405 Event Generators
Need names of responsible parties for all generators,
to work on the interfaces between CDF code and generators,
and also to do validation.
406 StdHep
Doesn't allow you to work with vecbos, dyrad, etc.
Doesn't have place to store generator tag.
HEPG format not optimal spacewise.

415 QQ
Interface needs to be redone.
417 Tauola
Tonya Molique from Perdue.
Interface needs to be redone.
425 Prototype with 1 subdetector
The sub-detector used for the prototype will likely be TOF
by Mathew Jones, according to Pasha and Marjorie and
Simulation WBS meeting.
428 Geometry database
needs to read geometry settings in at run time rather than compile them into
the code. Needs to define different alignments without compilation.
Tied Alignment issues? Want to store nominal geometry in one place,
and alignment in different place. correction for misalignments are closer
to calib constants, but still read nominal geometry from somewhere.
how do we change a single number if we store it in the database?
Where do we want the nominal geometry to be stored so that accessing it
is simple for a physicist.
429 Physics Param Tools
Call dan for explanation.
430 Track Embedding
Analysis oriented task, really shouldn't be here.
432 Digitization Rewrite
The strategy to do this quickly will be to reuse the old
code wherever possible.
436 talk to
Going to be redone with a new system of menus.
Related to the infrastucture
449 QFL
Frozen.
457 Update Detailed Geometry (GEANT)
Not certain whether John Mayer will do this.
458 Miniplug
This is a standalone program.
460 generally
no validation
462 Magnetic Field tuned
Nobody responsible. Mike Gold is most natural.
463 Validate Stub quality
Reconstruction task.
464 Test track-stub matching
Reconstruction task.
469 Magnetic Field Map
not done. not sure what it means.
470 Study Stub quality
Not sure if this is done because we cant link stubs to tracks.
482 Interface to Geometry DB
DB/alignment questions as always.
493 fix hit generation (change from fortran to C++)
Peter Tamburello has some of the hooks needed.
This is a big task. needs validation and integration.
Bill Ashmanskas interested.
494 Put correct material in active COT
not done. will use data.

hope bill Ashmanskas will work on this.
Ongoing task.

497 Geometry DB and Misalignments
Same Geometry/Alignment issues, and also calib database issues.

500 TOF
Needs work for new Geometry Framework.
Mathew Jones is doing a good job.

501 CLC
Needs work for new Geometry Framework.
Darin Acosta is doing a good job.

505 Geometry Classes
None.

512 XFTsim
Brian Winer. In CVS, development in a week.
Compiles and runs under development. Still needs to write output
in XFT banks, which
are in place. Two driver modules, one uses root, and the other is standard.
Root is not acceptable.

513 SVTsim
Level 2 SVT Simulation.
Lorenzo Moneta and Christian Berclaz from Geneva are writing an offline
simulation. Presented an accepted design. Punzi, Ashmansk and others lost
contact with them. SVT workshop in early April.
Geneve responsible for DB access for SVTsim.
Moneta, Berclaz, Punzi and Ashmansk are responsible.
Coding begun. Roughly 2 months of coding necessary. Want it by end of
May. Want to run it as a first filter in level 3.

514 Calsim
Simona Rolli, from Michigan L2 Monica Techio and Heather Ray (student),
Level 1 Ray Culbertson, Pedro Amaral from Chicago. Done as of 4/1/00.
In testing. Ongoing maintenance.

515 Cmusim
Muon is eric james and joe khua from Michigan.
Simona assisting. Design almost finished. Start coding soon.
About a month to code. Realistically by middle of may should be done.
The output bank is done, as with the other simulations.

516 xcesim
Argonne (K. Byrum) responsible. Simona hasn't heard anything.

517 L1 Global and L2 Global
Collects input from various L1 subsystems and produce scaler and TL1D bank.
Interacts with parser.
Greg Field from Yale responsible for L1
Stephen Miller and Ken Bloom responsible for L2: produces TL2D.

518 XTRP
Extrapolator from track to stub. XTRP boards simulation.
nobody is assigned to this.

519 Parser
Fully specifies the trigger in a language that the simulation can understand.
Reads DB to get list of triggers, and instructs simulation what the list of
triggers are. Requires Stephan miller and Kirsten Tollefson to have a design
for the trigger database. A level 3 parser exists but is not needed here
where we need a level 1 and 2 parser. Critical path item for the experiment
that the parser does not exist. Responsibility of upgrade; steve miller and
kirsten.

520 Database Access
Simulation needs to access the trigger database, calibration, run conditions.
Wants to have one person working on database issues only.
Same as parser.

521 Validation
Code that runs to verify that simulation of trigger looks like real trigger.
1. validation from andy.
2. engineering run, does sim agree with data.
3. march 2001 repeat 2.
connected with work going on in trigmon. validating caltrigger in trigmon.
Simona will supply tcl to run caltrigger, etc. for andy. ongoing task.