GLAST Large Area Telescope

Flight Unit Peer Review
16 September 2004

Flight Software Testing

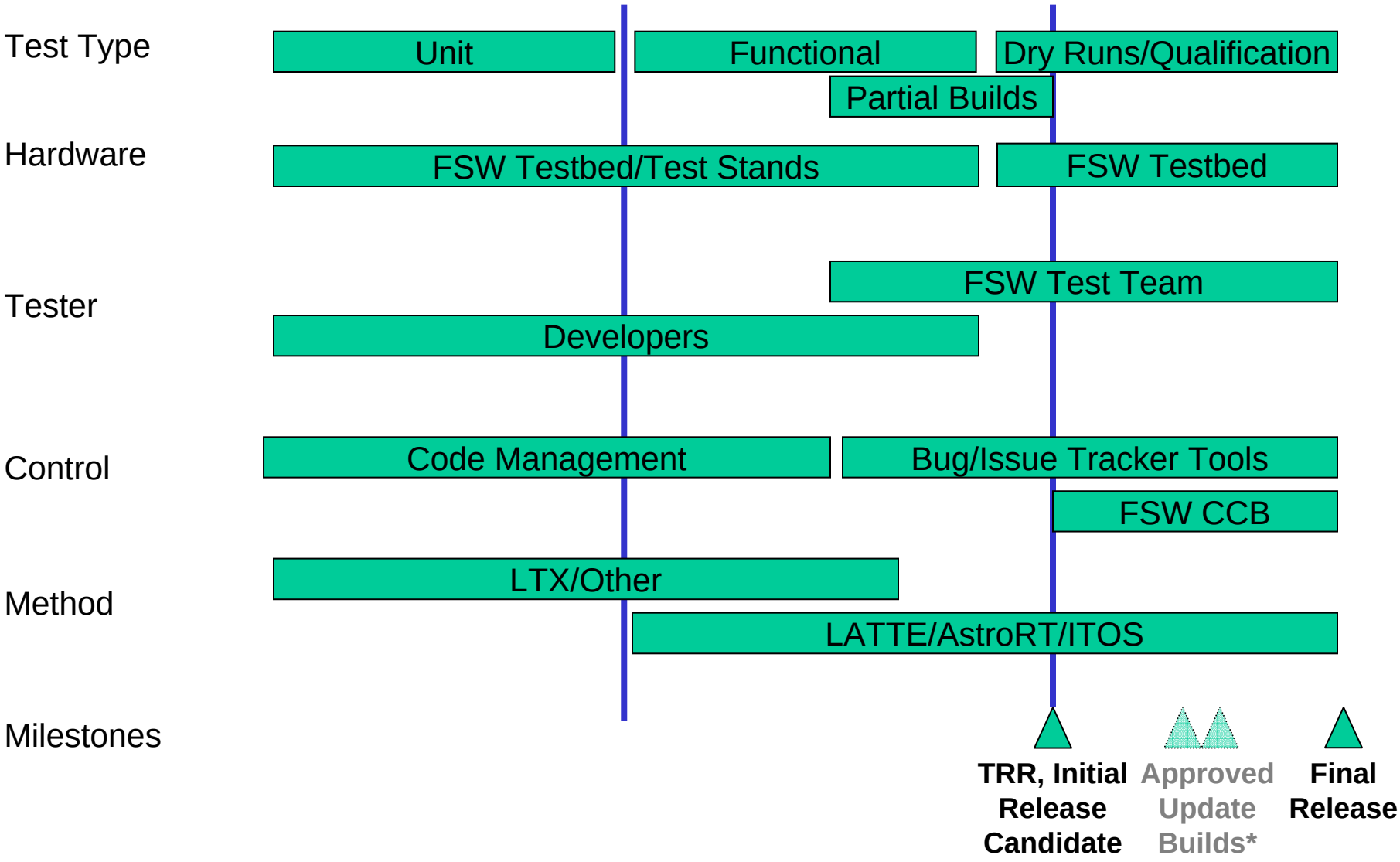
Eric Hansen
Stanford Linear Accelerator Center

hansene@slac.stanford.edu
(650) 926-4057

FSW Test - Agenda

- **FSW Test Process**
- **FSW Test Environment**
- **Implementation Example: ISIS as Pathfinder**
- **FSW Test Status**
- **FSW Test Schedule**

Flight Unit Test Process Overview



*If necessary

FSW Test Process

- **FSW Test Process**
  - **Unit Testing**
  - **Functional Testing**
  - **Formal Qualification Testing**
  - **Issue Tracking**
  - **Deliverables**
- **FSW Test Environment**
- **Implementation Example: ISIS as Pathfinder**
- **FSW Test Status**
- **FSW Test Schedule**

Unit Testing

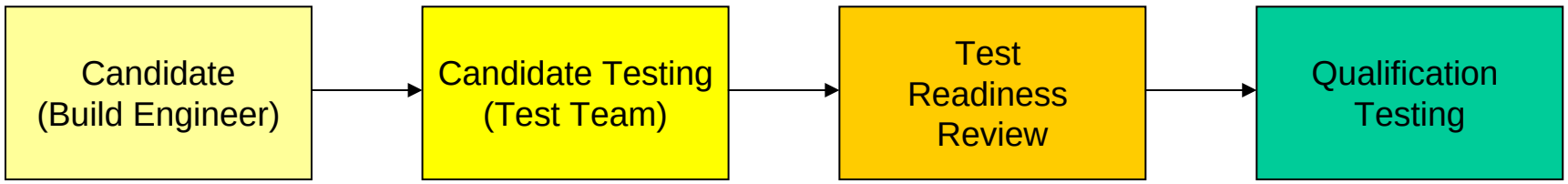
- **Goal:**
  - **Verify error (and warning) free compilation and code integrity.**
  - **Verify adherence to the specified programming standards of the GLAST-LAT SDP.**
  - **Verify proper function execution to include control flow, logic paths, error detection, and error recovery.**
  - **Verify compliance with processing accuracy requirements.**
  - **Verify compliance with memory and timing requirements.**
  - **Verify ability to accept and properly process the complete design range of input data and parameters.**
  - **Verify output functions and associated formats**
- **Performed by developer at package level**
- **Repeated whenever package updated**
- **Results documented informally in Package Test Directory (PTD)**
- **Scripts, results, and analysis products captured in LTX or other test environment**
- **Test code, descriptions, and vectors maintained in PTD**

Functional Testing

- Goal: Software-hardware and multi-package integration
- Performed on integrated packages by FSW developers
- Driven by test engine scripts
- Results documented informally in a common location.
- Results subject to informal sell-off to FSW Test Team Lead
- Continues until Flight Unit Candidate Release produced

Transition to Formal Qualification Testing

- **Software content freeze**
  - **~2 weeks prior to release date of a build containing all functionality**
  - **Changes require FSW CCB approval**
- **Release candidate submitted to FSW Test Team by FSW Build Engineer**
- **Flight Unit Candidate Testing confirms build is ready for FQT**
- **Test Readiness Review held**



Formal Qualification Testing

- Test Readiness Review (TRR)
  - Confirm readiness for Qualification Testing
    - Review test procedures and tests to be performed
    - Confirm test equipment readiness
    - Review open issues
  - Participants: GSFC, LAT project mgmt, LAT QA
- Qualification Testing
  - Conducted on testbed
  - Executed over several days
  - Monitored by LAT SQA
  - Test engineer executes procedures, writes test report
  - Issue reports generated as applicable
- Post Qualification Test Review (PQTR)
  - Confirm tests successfully completed (or specify what is required for completion)
    - Review test results package
    - Review issue reports
  - Held within 10 working days after last qualification test session

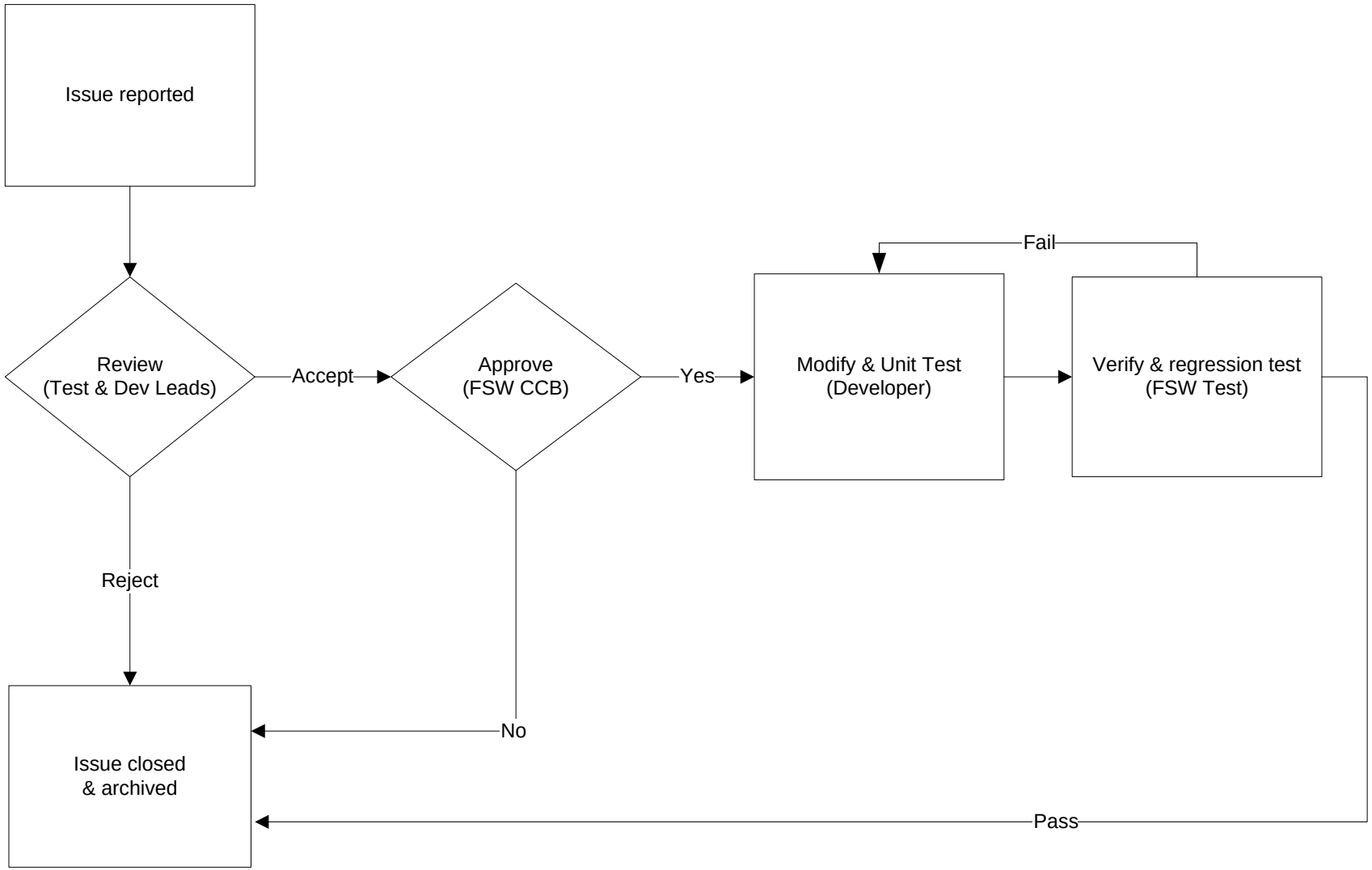
FSW Test Issue Tracking

- JIRA forum (bug-tracking system)
  - http://jira.slac.stanford.edu
  - Assists in evaluation of open issues and resolution
  - Includes these categories:
    - Software Problem/Change Report (SPCR)
    - Command & Telemetry Database issues
    - Test Script issues
  - Used to report problems, recommend changes
  - Status available online
    - Includes counts of open issues, assigned issues, etc.

FSW Test Issue Tracking (continued)

- **Process**
  - Issue report entered for FSW or GSE problem
  - Test & Development leads determine jointly if issue warrants investigation or modification to FSW requirements, design, or test procedure(s)
  - For accepted FSW issues, FSW CCB must approve updates and the scope of regression testing
  - Developer is assigned
  - FSW changes are unit tested
  - FSW Test Team verifies change and performs applicable regression testing
  - Test & Development leads
    - Review test results
    - Accept (or reject change)

FSW Test Issue Tracking (continued)



FSW Test Deliverables

- FSW Test Plan (Released by FSW CCB)
  - LAT-TD-00786-03
- FSW Test Procedures (To be reviewed by FSW CCB)
  - LAT number to be assigned
- Results packages (To be reviewed at PQTR)
  - Summary/analysis of qualification testing
  - Test reports

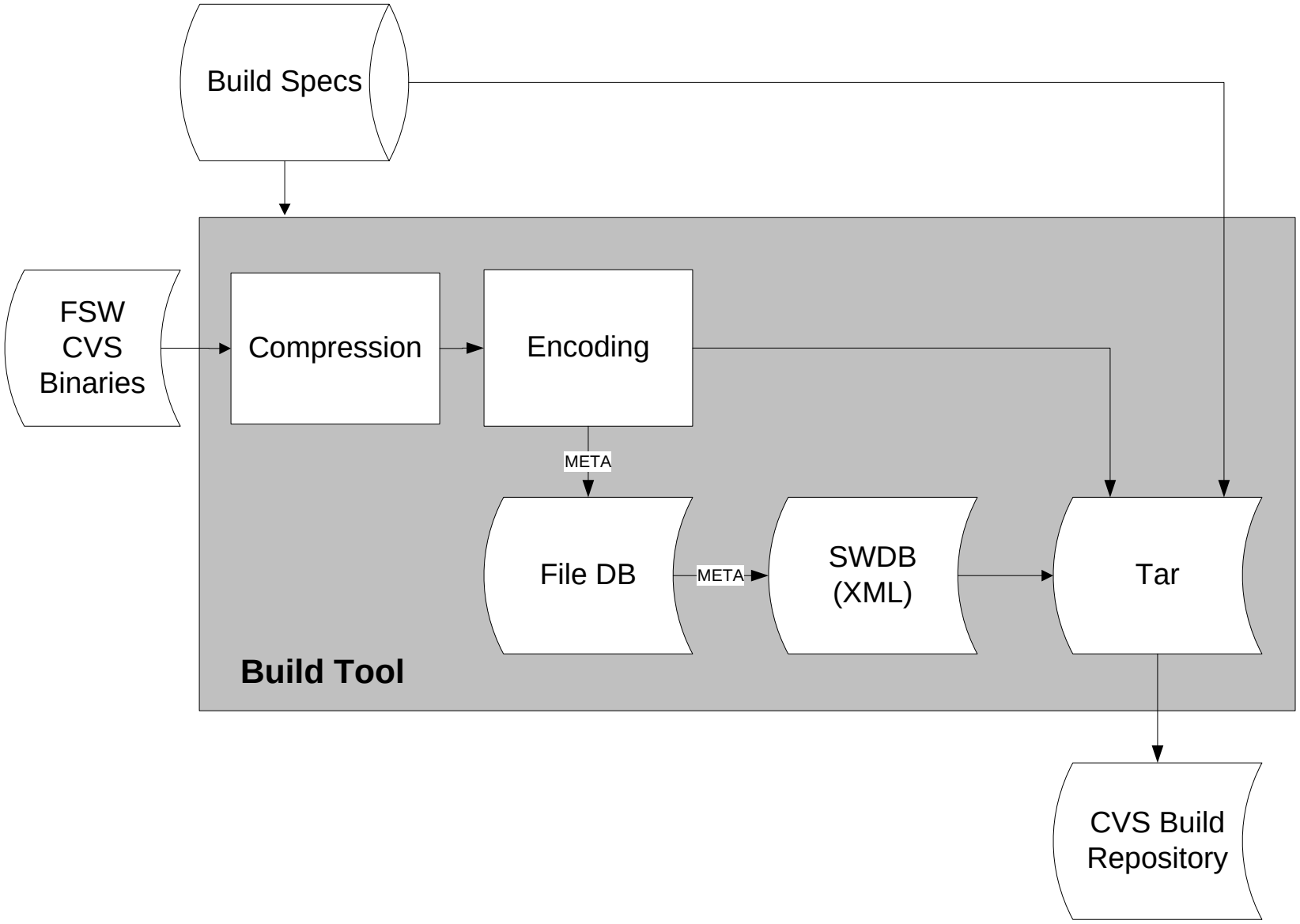
FSW Test Environment

- FSW Test Process
- **FSW Test Environment**
  - **FSW version control**
  - **ISIS version control**
  - **Hardware Infrastructure**
  - **FSW Test Engines**
- Implementation Example: ISIS as Pathfinder
- FSW Test Status
- FSW Test Schedule

FSW Version Control - CMX

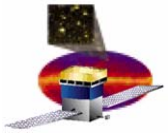
- All FSW is controlled under Code Management eXtra (CMX)
  - Wraps CVS repository
  - Allows for version tagging
- New CMX tool to create “uploadable” builds by specification file
  - Tool ingests a list of packages with versions
  - Outputs encoded, compressed files for upload with file number assigned automatically according to location in random access database table in MOC format
  - Also produces the Software Database (SWDB) XML file that PBC uses during SBC execution to locate and load software modules
  - Stores all products (including input list) as a tarball and commit to CVS for later extraction without the need to recreate the build environment
  - Update and extension of existing SRT (System Release Tool)
  - In progress, target completion is November 2004
- CVS repository set up for FSW test scripts

FSW Version Control – Build Tool



ISIS Flight Software Version Control

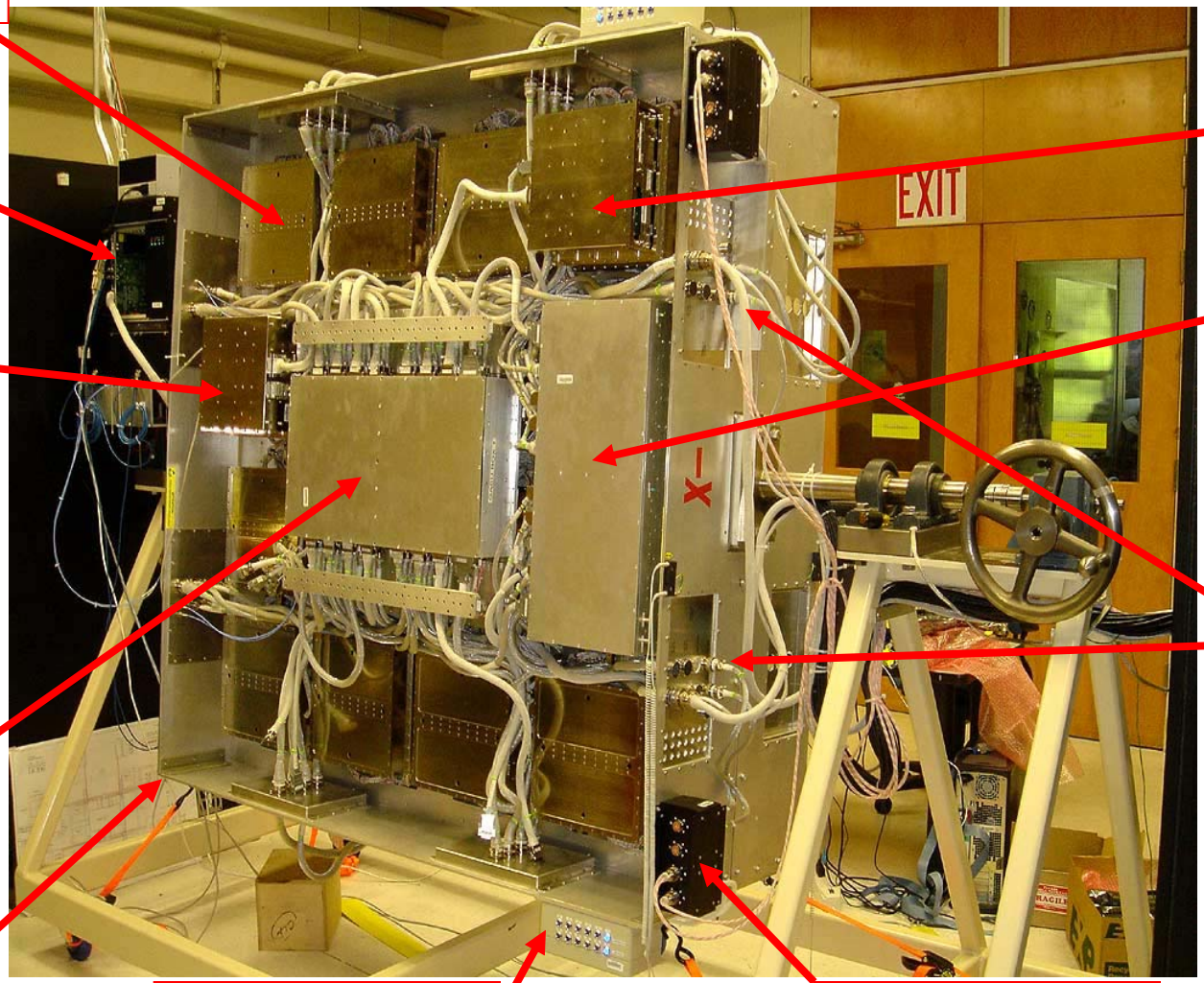
- We will take the following steps to ensure traceability of all software we burn into the ISIS EEPROMs.
  - Construct a build list containing all necessary software with version numbers and build dates.
  - Track file numbers by spreadsheet for future migration to random access database
  - Manually produce SWDB XML file
  - Tarball code, binaries, and C&T DB
  - Place tarball in CVS Build repository
- CVS Repository set up for ISIS test scripts



Hardware Infrastructure

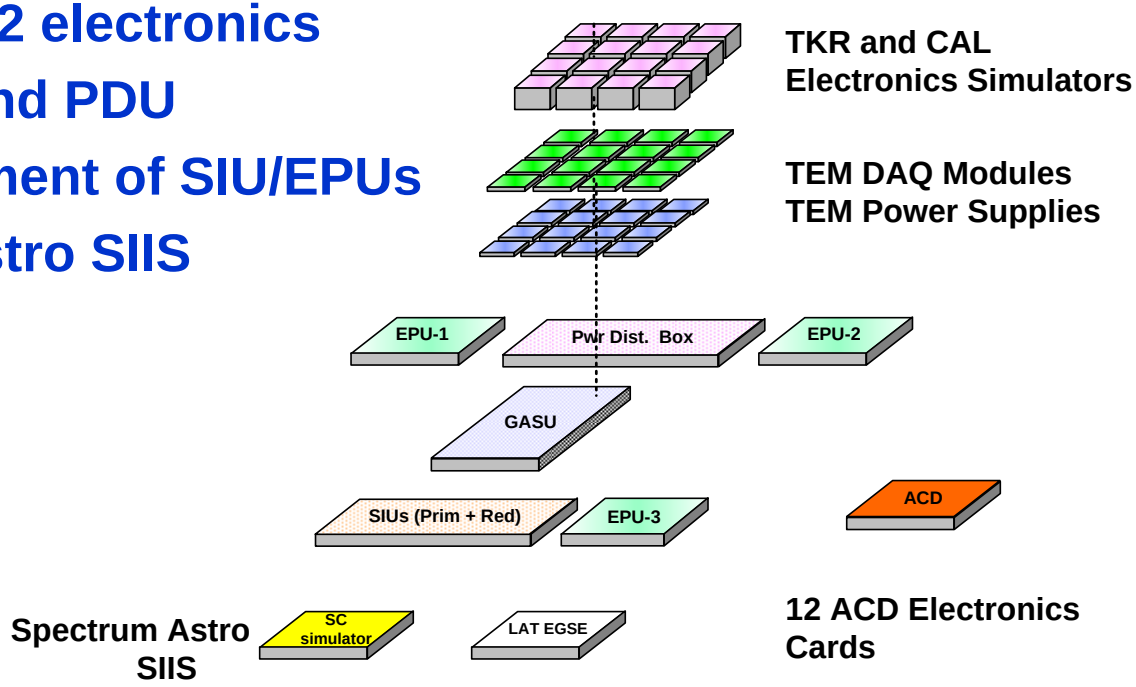
- Hardware versioning
  - All hardware units are assigned a GLAT# (bar coded)
  - Each FPGA also has unique VHDL# (stored electronically)
  - Hardware database is used to keep track of versions
- Problems/Bugs
  - Problems reported to the ELX group and tracked using JIRA
  - ELX group maintains hardware, evaluates and repairs bugs
- Hardware infrastructure needed to test the flight software
  - Testbed (including FES and DAQ)
  - SIIS (Spacecraft Instrument Interface Simulator)
  - Corner test stations
  - GPS

The LAT Testbed (-z)

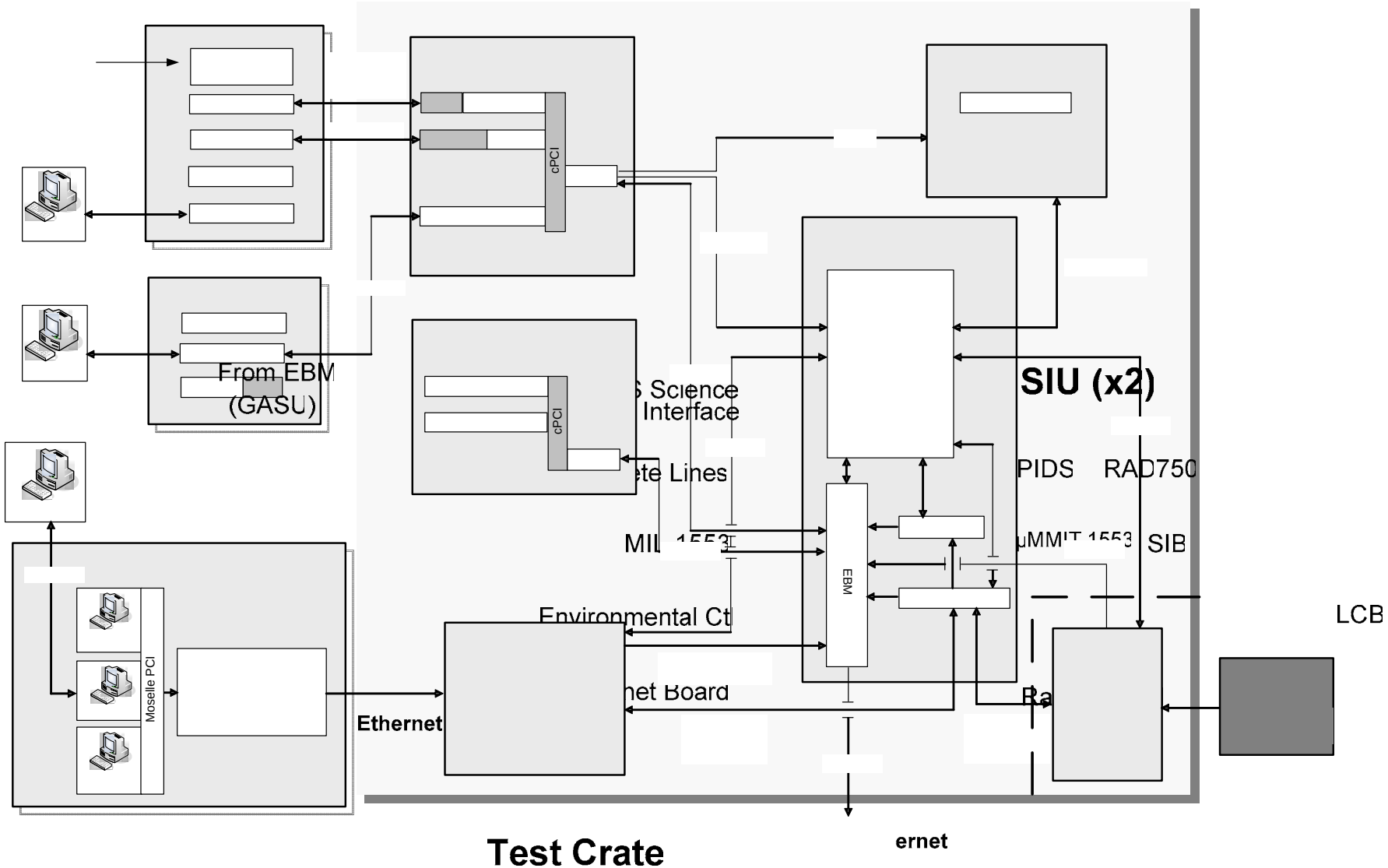


Testbed Layout

- **Provides a full DAQ system with EM2 hardware (interfaces and functionality identical to flight)**
  - **16 TEMs and 16 TEM power supplies**
  - **Full TKR and CAL front-end electronics for one tower**
  - **Front end simulators (FES)**
  - **Full ACD EM2 electronics**
  - **Full GASU and PDU**
  - **Full complement of SIU/EPUs**
  - **Spectrum Astro SIIS**

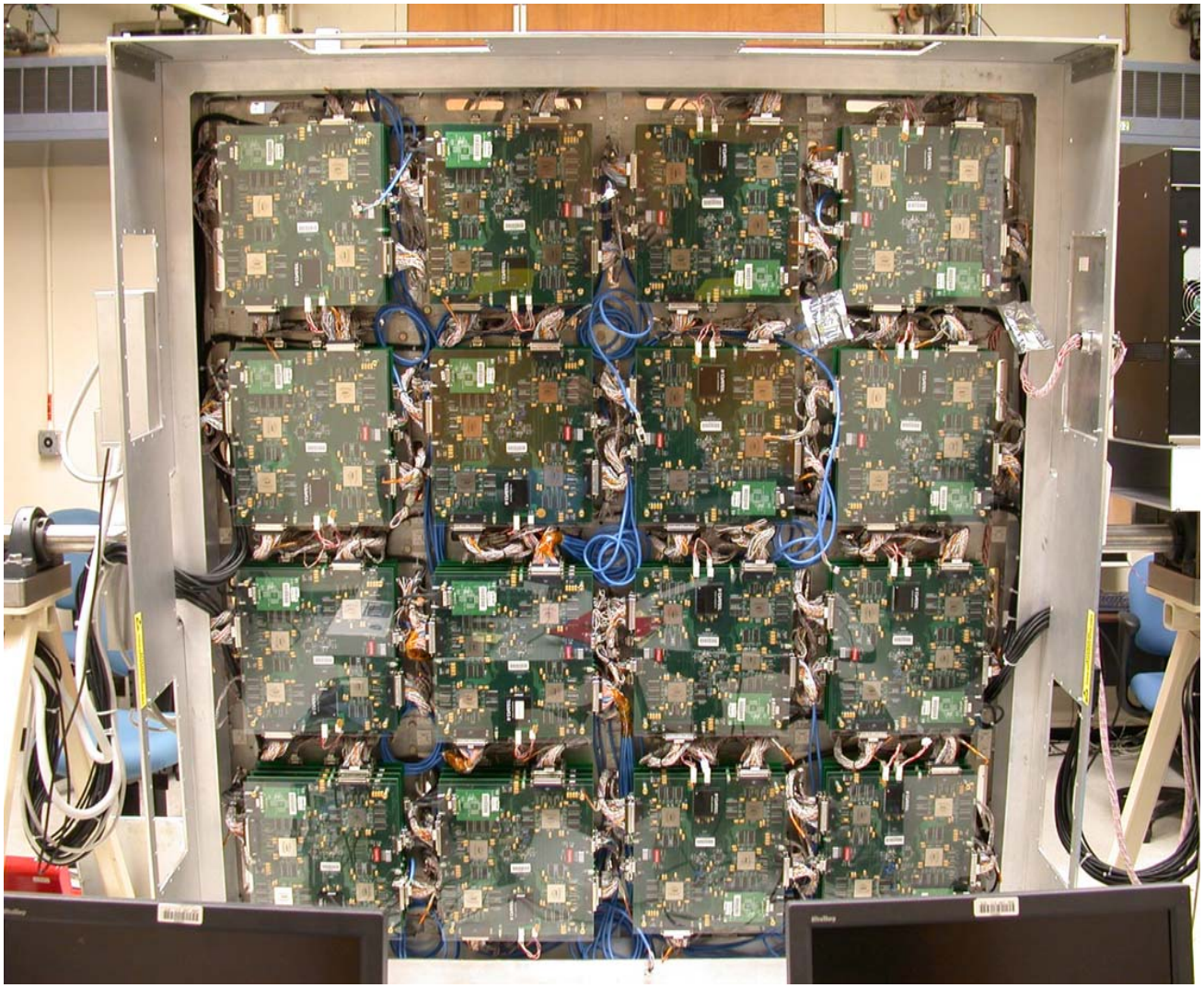


Testbed Schematic

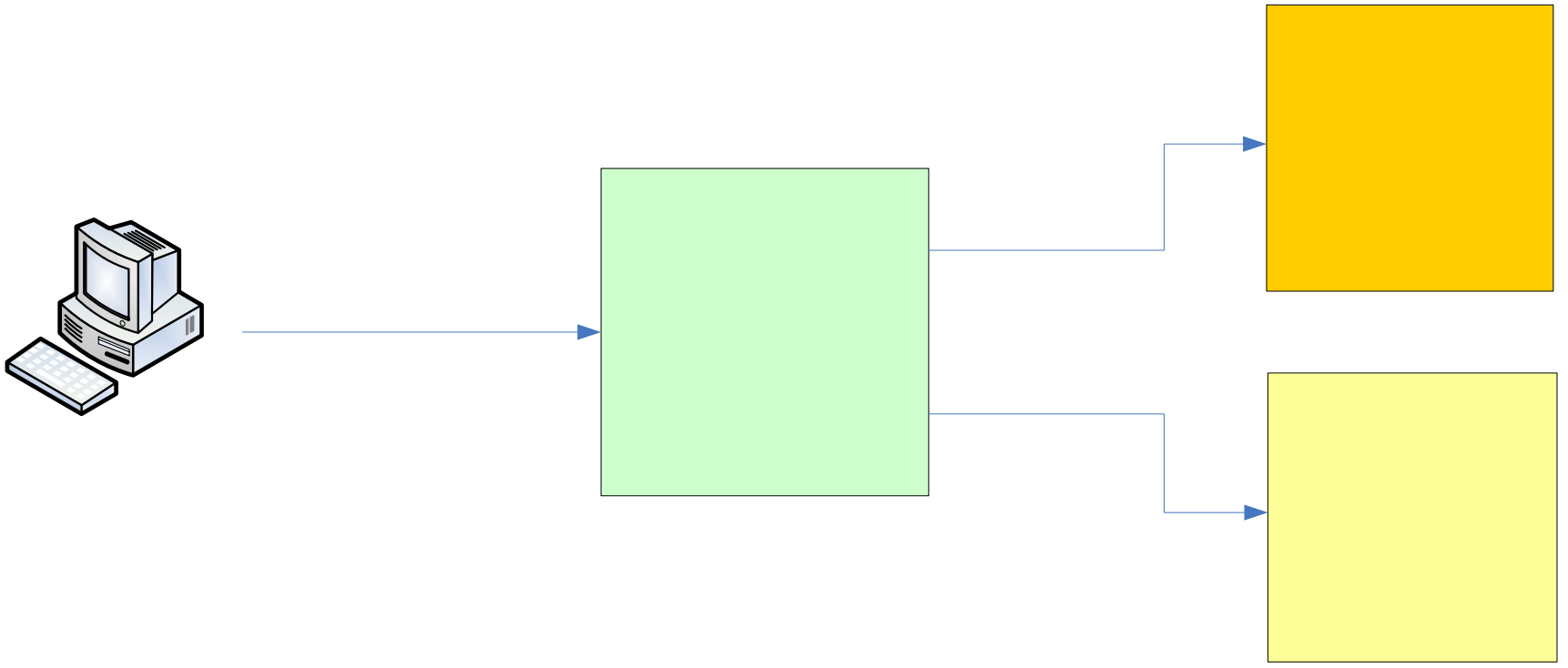


FES (Front-End Simulator)

- **16 Front End Simulator electronics boards (one for each tower) controlled by PCs.**



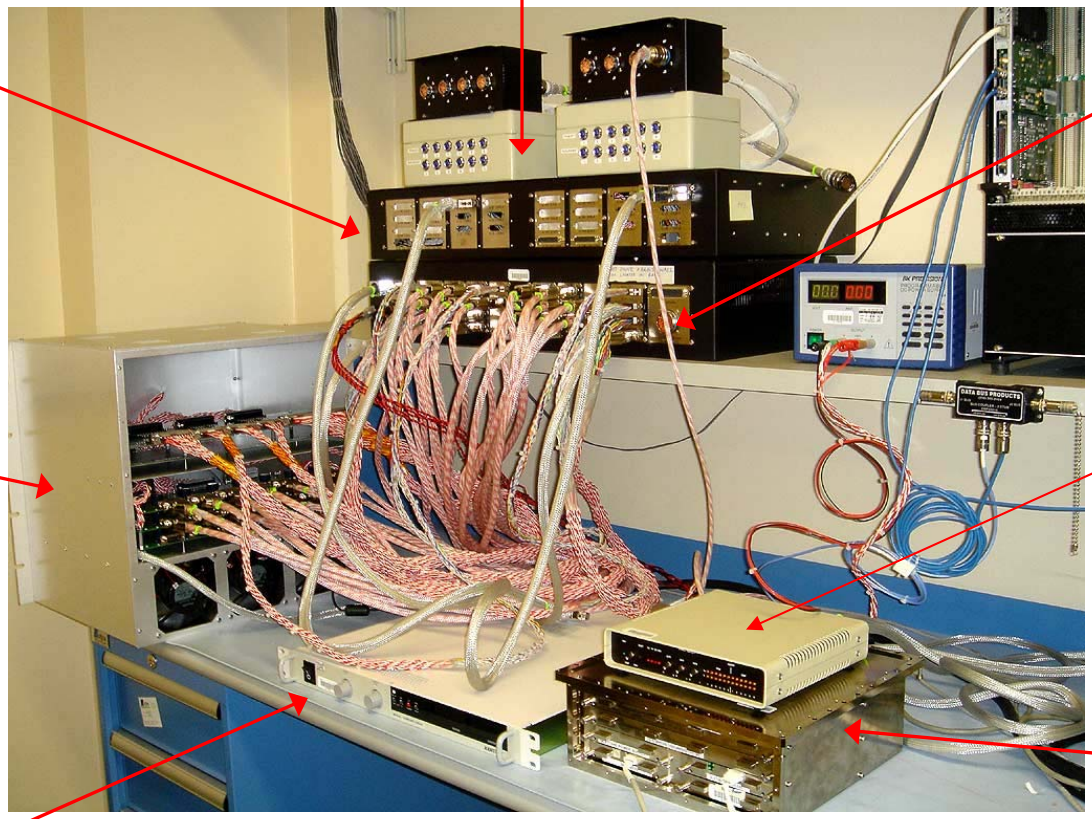
FES Data Flow



FES – Testing Goals and Functions

- Goals
  - Prove data can properly flow from the front-end cables to the CPUs
  - Provide a means for measuring the throughput of the T&DF system
  - Provide a means of simulating incident gamma-ray phenomena
- Functionality
  - Input side: PCs define the signal patterns to be pulsed to the T&DF system
  - Output side: The boards manage the time sequenced distribution of data to the T&DF subsystem trigger and data cables
  - User interface: Allows testers to simulate contributors anywhere from one up to a number representing the entire LAT
- Tests
  - Monte Carlo generated physics data patterns or other known test patterns are created and pushed through the FES
    - Check data flow integrity
    - Verify that the T&DF responds and handles errors correctly
    - Check data flow throughput by running simple test patterns through the system at >= nominal rates
    - Determine if the EPU filtering algorithm(s)/code work

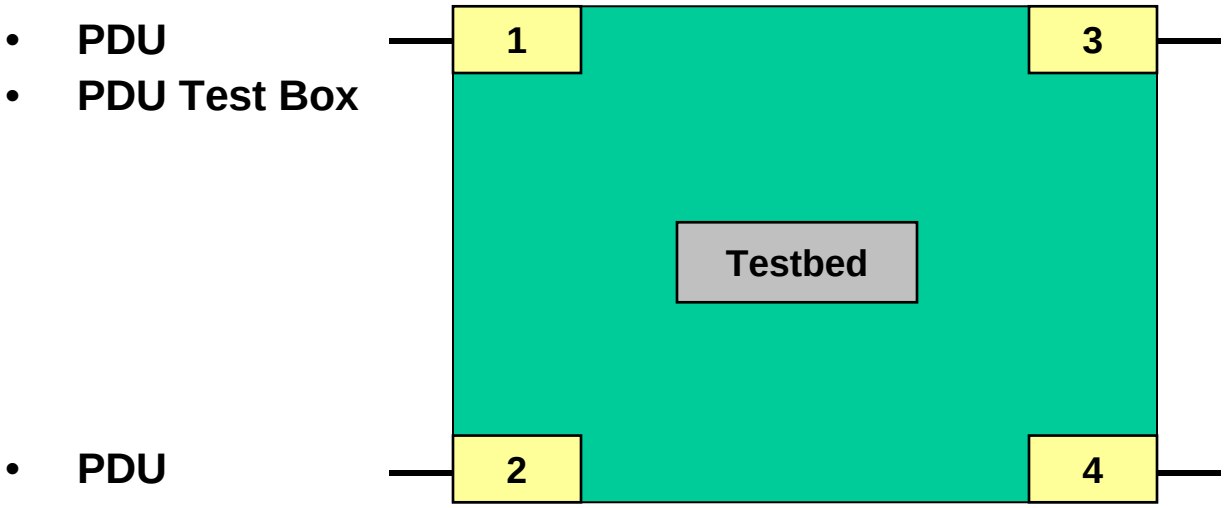
Corner Test Benches



Corner Test Benches

- Test benches located in the 4 corners of the dataflow lab
  - Each can contain some of the following:
    - GASU
    - PDU
    - PDU load box
    - SIU/EPU
    - TEMs
    - FREE cards
    - Mini-tower
      - 5 layers of TKR
      - 2 layers of CAL
    - Full tower (17th tower)
  - Components interchangeable for reconfiguration based on functionality and need
- Capable of handling every test except those that require physics data
- Allows parallel development with FSW
- Reduces competition for the Testbed

Test Station Layout + Contents



- PDU
- PDU Test Box

- Partial tower
- TEM
- 2 EPUs
- 1 SIU

- PDU

- Full tower
- TEM

- All corners
  - VME crate (emulates SIU or EPU)
    - LCB and 1553 PMC cards
  - GASU
  - cPCI crate
    - SIU or EPU
  - Crate tester
    - Discretes
    - Heater Control emulation
    - SC Reset signal

- Apportioning of test stands
  - 2 for FSW/FSW Test
  - 2 for HW Commissioning
- Majority of test scripts can be developed on corner test stations

SIIS

- SIIS = Spacecraft Instrument Interface Simulator

- Used to validate electrical interface between spacecraft and LAT
  - LGIO/Science interface (SSR)
  - 1553 – commanding the LAT
  - Spacecraft discrete monitors and controls

- Used in several capacities
  - Initially, used for ISIS development
  - Eventually, will migrate to testbed for system testing
  - In April 2005, I&T expects delivery for clean-room usage

FSW Test Engines

- LAT Test Executive (LTX)
  - Developed by Flight Software Team
  - Uses Python test scripts combined with XML test descriptions
  - Integrated with MySQL for test reporting
  - Useful for lower-level testing
    - Unit and Functional
- LAT Test Executive (LATTE)
  - Developed by LAT I&T Team
  - Also uses Python test scripts
  - Has a hook for MySQL reporting (eLogbook) but may need reformatting
  - Useful for majority of testing (excluding 1553 and Science Data interfaces)
- AstroRT
  - Developed by Spectrum Astro/General Dynamics
  - Uses Perl test scripts to interface to their engine
  - Depends on test engineer to properly formulate a test log/report
  - Useful for high level testing from spacecraft interaction perspective
- ITOS
  - Possible use to emulate MOC/ISOC environment
    - Evaluation of applicability complete next week
  - Much better support than AstroRT
  - Useful at same level as AstroRT

FSW Test Engines (continued)

- ITOS Database format
  - We must ensure that the files we provide the MOC/ISOC conform to the ITOS format
  - Only current test tool is AstroRT translation
    - Not fully implemented
    - Somewhat buggy w.r.t. bitfields and size/type
  - May need additional ITOS interpreter to ensure proper formatting
    - Relatively small effort to produce this tool if necessary
    - Depends on how much fidelity the new delivery of AstroRT provides to ITOS standards.

FSW Test - Agenda

- FSW Test Process
- FSW Test Environment
- **Implementation Example: ISIS as Pathfinder**
- FSW Test Status
- FSW Test Schedule

Implementation Example: ISIS

- The ISIS project is the pathfinder for the LAT QT
- Steps in the process
  - Requirements identified and approved by FSW CCB and LAT management (LAT-SS-03975-01)
  - Test Plan complete and approved by FSW CCB and LAT management (LAT-MD-04086-01)
  - Several test procedures written
  - Scripts conform to a 1:1 relationship with requirements
  - Test Readiness Review (TRR) scheduled for October
    - Involves GSFC, LAT Project Management, LAT QA
    - Allows participant review of all Qualification Test Procedures
    - Ensures full test coverage of all ISIS requirements
  - Formal Qualification Testing (FQT) to follow TRR
  - Post Qualification Test Review (PQTR) to follow FQT
    - Allows PQTR participants to review test results and accept/reject product
    - Involves GSFC, LAT Project Management, LAT QA

ISIS Test Procedure Layout

- Description of required hardware and connections
- Description of required software setup (GS-SW)
  - Displays needed
  - Any initialization needed for script to run properly
- Steps necessary to initialize and run test script
  - Name/location of test script from CVS repository
- Description of how to interpret results and where to find them
  - Name/location of log file(s) and any analysis file(s)
- Pass/Fail criteria

FSW Test - Agenda

- FSW Test Process
- FSW Test Environment
- Implementation Example: ISIS as Pathfinder
- **FSW Test Status**
  - **Status**
  - **Dependencies**
  - **Progress tracking**
- FSW Test Schedule

FSW Test Status

- FSW Requirements defined
- FSW Test Plan complete and approved
- Engineers working on test procedures and scripts
  - Approximately 14 of 198 written
- Repository set up for FSW test scripts
  - /nfs/slac/g/glast/flight/fsw_test/FSW
- Database storage of test results/reports in progress
  - For now, they reside in a common location
  - Expected mid-October
- Web reporting of test procedures, test results/reports
  - In progress
  - Expected mid-October
- JIRA forum (bug-tracking) system to assist in evaluation of open issues and resolution
  - Set up and in use
- Hiring a physicist to perform relevant science-oriented testing
  - Will bring FSW Test Team to 4 people

FSW Test Development Dependencies

- Hardware Dependencies
  - SIIS
  - Hardware availability
    - Competition from FSW developers
    - Competition from HW commissioning
  - GPS device for PPS signal and other timing tests
    - On order from Symmetricom
- Software dependencies
  - Proper data sets (Monte Carlo) for testing of filtering algorithms, GRB detection, trigger, and celestial coordinate handling
  - Completion of FSW coding in time to perform appropriate testing

FSW Test Progress Tracking

- 1:1 requirement to script relationship
  - Eases progress tracking
- Integration into FSW Demos
  - Delayed by test team ramp up, ISIS production
  - Look for this in upcoming monthly demos (Oct.)

FSW Test - Agenda

- FSW Test Process
- FSW Test Environment
- Implementation Example: ISIS as Pathfinder
- FSW Test Status
- **FSW Test Schedule**

FSW Test Schedule

- Milestones
  - TRR
    - Conducted when FSW Test Team comfortable with Test Procedures, Scripts, and FSW Build
  - FQT
    - Conducted following TRR and approval of Test Procedures
  - PQTR
    - Follows FQT to review results of the qualification testing

FSW Test Types

- **Used to organize tests by functionality and divide testing effort**

Test #	Test	Description
1	1553 Interface Test	These tests shall first verify the electrical characteristics of the LAT 1553 interface, then demonstrate (in conjunction with a spacecraft simulator) that the LAT can exist on the bus without responding to messages directed to other remote terminals. Next, processing of telemetry requests shall be demonstrated, followed by exercise of the command interface to the LAT, and verification that all commands and command data are correctly interpreted. Commands need not be executed to successfully carry out these tests . Timing measurements for the standard message traffic shall also be performed.
2a	Command Functional	These tests shall verify that all LAT commands are exercised through the 1553 interface, with command execution and completion status reported in telemetry when command verification features are switched on. The tests shall also verify that the SIU FSW validates commands and command parameters prior to command execution. Finally, the tests shall verify that the SIU executes individual commands received directly from the SC or executes preprogrammed command sequences (block commands) when triggered by a command from the SC, and can properly execute single commands and block command items in an interleaved or serial fashion, as necessary.
2b	File Management and Memory Upload and Dump Verification	These tests shall verify that the FSW can, on command, report file system status, create and delete file directories, dump directories, delete files, copy files (on a single SIU or EPU, and from an SIU to an EPU), dump files, and load files during (SIU) primary boot. These tests shall also verify the capability to successfully upload data into a specified block of SIU or EPU memory and verify the correct memory contents. These tests shall also demonstrate the successful dump of the contents of a specific block of SIU or EPU memory. Finally, the tests verify that the event filter can be reprogrammed by choosing or replacing files in the file system.

FSW Test Types (continued)

3	Narrowband Telemetry Verification	These tests shall verify the successful collection, sampling rate, formatting, downlinking and transformation into user units of housekeeping and low-rate science data, alert telemetry, and diagnostic telemetry. If telemetry content is specific to a particular mode (i.e., boot mode or standard operation mode), these tests shall verify the correct content as a function of mode.
4	Wideband Telemetry Verification	These tests shall verify the correct formatting of events and associated supporting data, followed by transmission through the external LAT interface in a form that may be correctly decoded.
5	Vehicle Signals Interface Verification	These tests shall verify the timing and processing of all signals from the Spacecraft, as defined in the LAT-SC ICD [29].
6a	SIU/EPU CPU-to-CPU Communications Functional Test	These tests shall verify the LATp communications protocol for sending and receiving information between the CPUs, including timing, acknowledgments and other transport level functions.
6b	EPU Internal Configuration Test	These tests shall verify that the SIU can configure each EPU as desired via the CPU-to-CPU communications protocol and receive the necessary information back to verify the configuration. These tests shall verify the correct transmission, decoding and reply sequence for each configuration message between the SIU and the EPU(s).
7	T&DF, TKR, CAL, and ACD Internal Configuration Test	These tests shall verify the ability of the SIU to configure the T&DF, TKR, CAL, and ACD subsystems as desired and read back the necessary configuration information to completely determine each subsystem's configuration.

FSW Test Types (continued)

8a	Event Filtering – Reception of Events and Verification of Filtering Algorithm	These tests shall confirm that fully assembled, properly formatted events are received directly into EPU memory, where they can be processed by EPU FSW. These tests shall also verify that the on-board filtering algorithms running on the EPU(s) achieve the purity/efficiency ratios predicted by Monte Carlo. These tests shall include running a specific set of scenarios derived from Monte Carlo simulations with known, expected results. The filtering algorithms must provide acceptable results for each scenario in order to pass this test.
8b	Electronic Calibration	These tests shall verify the end-to-end processing of a known pattern of data from the LAT electronics to the EPUs. A pre-determined charge signal shall be injected into the electronics at the detector interface and the resulting event data shall be compared to expected values. These tests shall include at least 3 scenarios for coverage of various readout paths (to verify redundancy) and readout modes (to verify event data structures).
9a	FSW and LAT initialization	These tests shall verify that the SIU and EPU(s) can correctly boot themselves, report system errors during primary boot in housekeeping telemetry, establish a reference state based upon information in nonvolatile storage, signal boot status over a discrete line to the SC (SIU only), and then power and initialize the remaining LAT subsystems in an orderly manner, resulting in a known reference state at the end of the process. This state shall be verified through internal configuration read back and LAT telemetry. These tests shall also verify that the SIU and EPU(s) can write a boot error log and return the contents of the log in telemetry, on command.
9b	FSW and LAT re-initialization and recovery	These tests shall verify the mechanisms for recovery should the LAT or one of its processors become non-operational or lose communications, either internally or with the SC. These tests shall include a number of scenarios: orderly shut-down and re-initialization of LAT when communications with SC are lost, reboot of the SIU on receipt of the SIU_RESET discrete signal from the SC, re-initialization of the SIU by the watchdog function, re-initialization of one or more EPUs by the SIU or when the EPU watchdog function is triggered.
10a	LAT Pointing Calculations	These tests shall verify that the SIU FSW can promptly transform pointing vectors reconstructed from event data between the LAT relative coordinate system and the celestial coordinate system. The transformation shall be based on GPS, attitude and orbit information provided by the S/C and the event time tag read in with each event. Using simulated events and simulated SC time/attitude data, these tests will demonstrate the accurate reconstruction of the Monte Carlo-provided primary particle vector in celestial coordinates.

FSW Test Types (continued)

10b	LAT Time Signal Processing	These tests verifies that the LAT FSW correctly processes the 1 Hz GPS time message from the SC to provide a mapping of external time to the LAT internal 20 MHz clock, and that the resulting mapping allows event time measurements to be accurate to within 10 µsec of SC time.
11	GRB Handling/Repointing Requests	These tests shall verify that the SIU FSW correctly handles and responds to GRB alerts and repointing requests from both the ground and the GBM (via the SC). Verification shall include a variety of scenarios including repointing enable, different operational modes and conflicting requests. These tests shall verify that the SIU FSW responds to these alerts and requests properly, rejecting the appropriate ones and forwarding repointing requests to the SC for the other cases.
12a	Additional Science Processing – GRB detection and location	These tests shall verify the SIU FSW's ability to monitor for and detect GRBs in the LAT science data, with default GRB detection criteria adjustable by ground personnel. The test shall involve a variety of simulated observing scenarios based on Monte Carlo data with signal concentrations in various portions of the FOV. These tests shall verify the detection sensitivity and location accuracy requirements for the LAT. These tests shall also verify that, when a GRB is detected, the SIU FSW automatically applies a set of looser event filter parameters, allowing more events to be collected for a limited period of time. Finally, these tests shall verify the SC notification function and timing requirement.
12b	Additional Science Processing – Repointing Logic	These tests verifies the LAT's internal logic for requesting repointing for LAT-detected GRBs. Verification shall include a variety of scenarios including repointing enable, different operational modes and conflicting requests. These tests shall verify that the SIU FSW responds to these events properly, forwarding repointing requests to the SC, sending a command indicating the time of GRB detection, and (for a LAT-detected burst alert) issuing a LAT closeout message to the GBM.

FSW Test Types (continued)

013a	Operational Modes – Safe Modes	This test shall verify that the SIU FSW can perform the correct transition of the LAT into Safe Mode upon command, can properly execute the Safe Mode algorithms, and can perform the subsequent recovery via Engineering Mode. It shall also verify the correct response to SAA entry commands and subsequent recovery.
013b	Operational Modes – Science Modes	This test shall verify correct transitions between the various science observation modes and proper operation of the LAT in those modes. The test shall involve all observation modes: Sky Survey Mode (non-rocking as well as various rocking profiles), Pointed Observation Mode, and Repointed Observation Mode.
14	LAT Diagnostic and Calibration Modes	This test shall verify that the SIU FSW properly transitions to Diagnostic and Calibration Modes upon command, then properly executes the requested diagnostic or calibration algorithms.
15	LAT Thermal Control System (TCS)	This test shall verify that the SIU FSW can properly execute the algorithms of the TCS implementation, and that the TCS implementation successfully maintains the LAT within thermal limits.
16	LAT Event Performance Monitoring and Operations	This test shall verify that the FSW operates properly in every Science Mode, monitoring event data for integrity, tracking changes in event and detector statistics, and generating a report of any detected anomalies.

Test Readiness Schedule

Test Name	Test Type	Responsible Eng.	Procedure/Script Development	HW Needed
1553 Interface Test	1	Shantha	Sep-04	2
FSW and LAT initialization	9a	Igor	Sep-04	6
Command Functional	2a	Shantha	Oct-04	2
SIU/EPU CPU-to-CPU Communications Functional Test	6a	Igor	Oct-04	5
FSW and LAT re-initialization and recovery	9b	Igor	Oct-04	6
File Management and Memory Upload and Dump Verification	2b	Shantha	Nov-04	2
Vehicle Signals Interface Verification	5	Shantha	Nov-04	3
LAT Thermal Control System (TCS)	15	Eric	Nov-04	3
LAT Pointing Calculations	10a	TBH-Physicist	Nov-04	1
Electronic Calibration	8b	Igor	Nov-04	1
Narrowband Telemetry Verification	3	Shantha	Dec-04	2
EPU Internal Configuration Test	6b	Igor	Dec-04	5

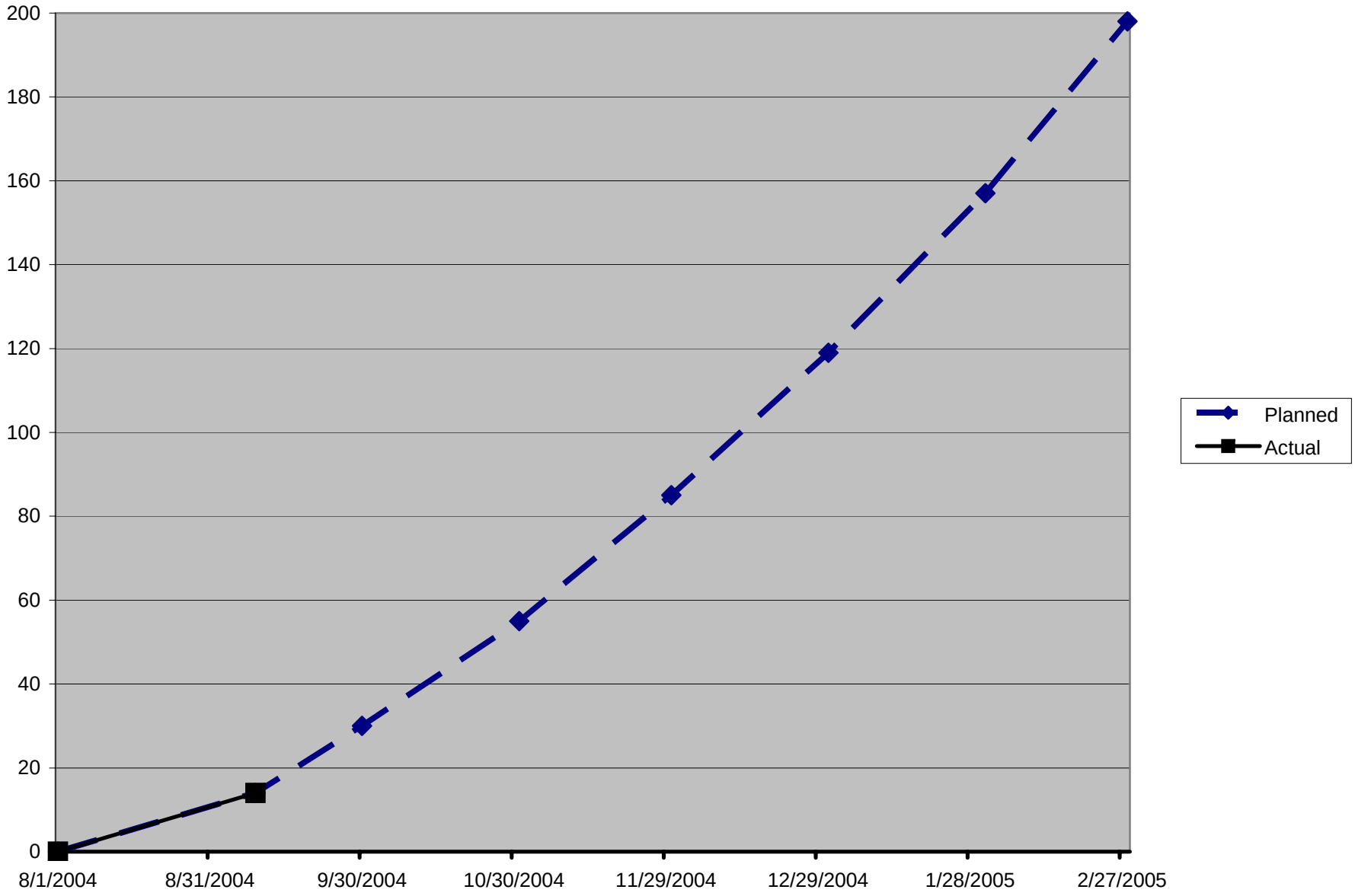
HW Code	HW Included
1	Testbed
2	SIU
3	SIU,GASU,PDU
4	SIU,GASU,PDU,TEM
5	SIU,EPU,GASU,PDU
6	SIU,EPU,GASU,PDU,TEM

Test Readiness Schedule (continued)

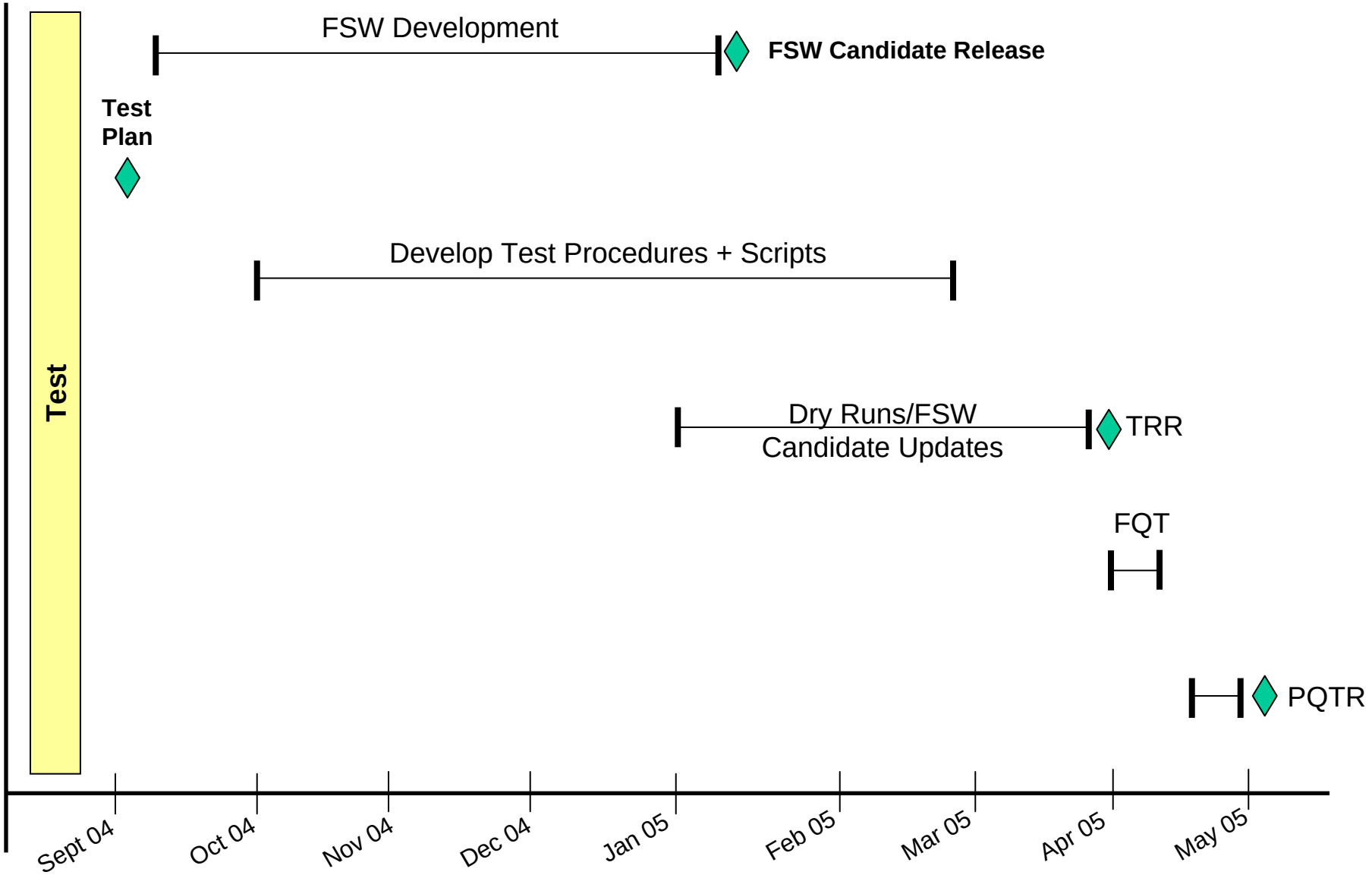
Additional Science Processing – Repointing Logic	12b	TBH-Physicist	Dec-04	1
Operational Modes – Safe Modes	13a	Eric	Dec-04	3
Event Filtering – Reception of Events and Verification of Filtering Algorithm	8a	TBH-Physicist	Dec-04	1
Wideband Telemetry Verification	4	Shantha	Jan-05	1
GRB Handling/Repointing Requests	11	TBH-Physicist	Jan-05	1
LAT Time Signal Processing	10b	Igor	Jan-05	3
Additional Science Processing – GRB detection and location	12a	TBH-Physicist	Jan-05	1
Operational Modes – Science Modes	13b	Eric	Jan-05	1
T&DF, TKR, CAL, and ACD Internal Configuration Test	7	Igor	Feb-05	1
LAT Diagnostic and Calibration Modes	14	Eric	Feb-05	1
LAT Event Performance Monitoring and Operations	16	TBH-Physicist	Feb-05	1

HW Code	HW Included
1	Testbed
2	SIU
3	SIU,GASU,PDU
4	SIU,GASU,PDU,TEM
5	SIU,EPU,GASU,PDU
6	SIU,EPU,GASU,PDU,TEM

FSW Test Procedure + Script Creation



FSW Test Schedule Through FQT



Summary

- We know how much work needs to be performed
- We know what we need in terms of hardware and personnel
- We have processes in place to execute our plan