

---

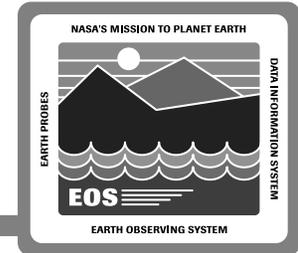
# FOS Road to AM-1 Launch

## Andy Miller

---

18 October 1995

# FOS CDR Roadmap



## FOS Overview

### FOS CDR Overview

- FOS CDR goals
- Driving requirements

### Engineering Activities

- Activities since PDR
- FOS team approach

### System Architecture

- Overview
- Features

## FOS System Architecture

### IST

- Capabilities
- Plans

### Hardware Design

- Computers
- Peripherals

### Network Design

- EOC LAN
- IST Connectivity

### FOS Infrastructure

- Mgt Services
- Comm Services

### Segment Scenarios

- End-to-End Flow
- Subsystem Interfaces
- Building block linkage

## FOS System Design

### Subsystem Design

- Detailed design
- FOS functions/tools
- Subsystem design features

### RMA

- RMA allocation
- FMEA/CIL

## FOT Operations

### Operations Overview

- EOC facilities
- FOT positions

### Operational Scenarios

- End-to-end flow
- Operations perspective
- FOT tool usage

## Road to Launch

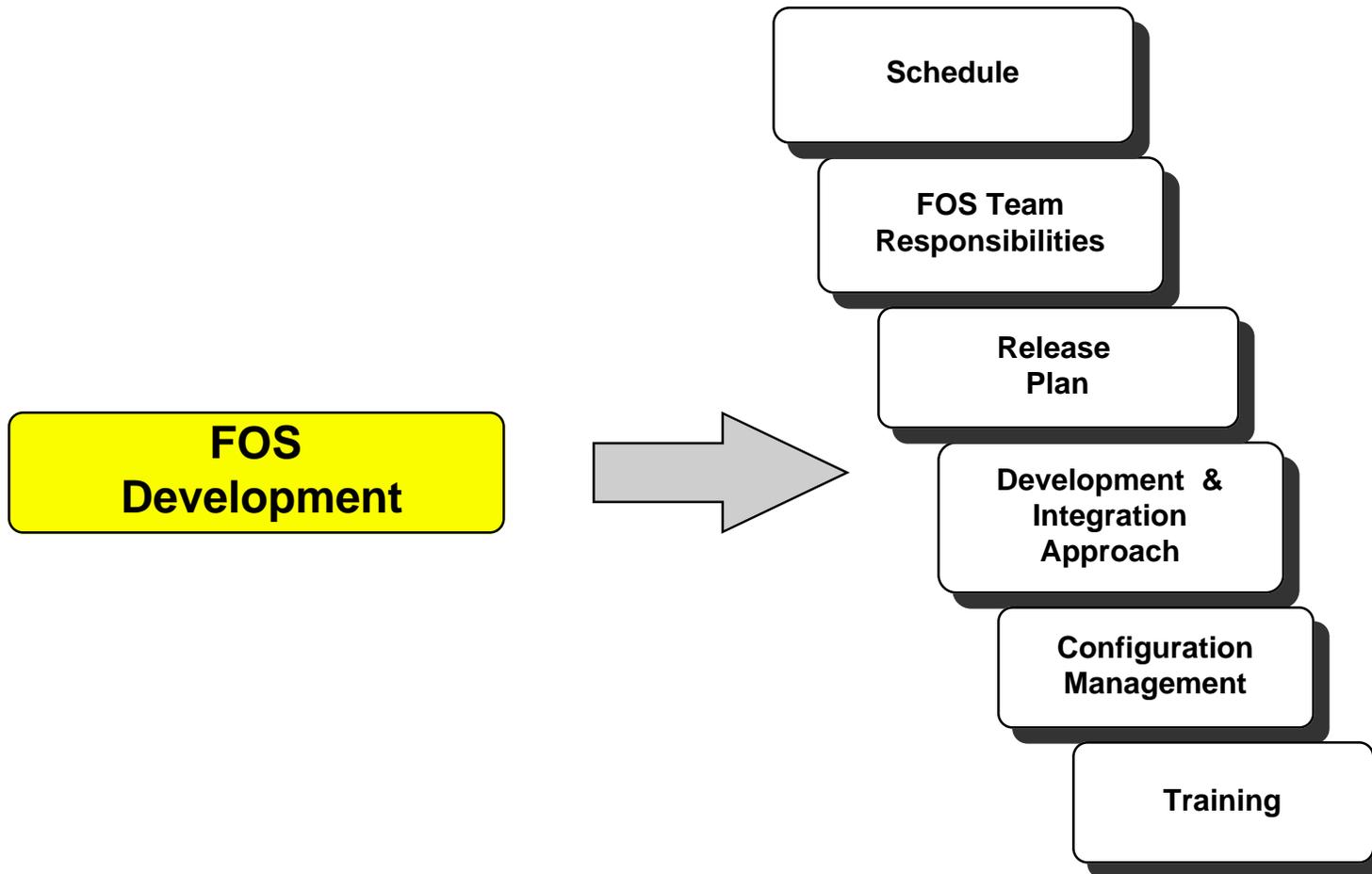
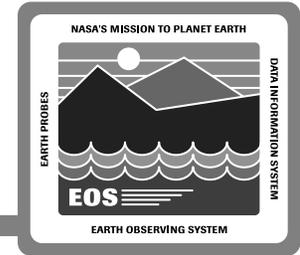
### Development

- Release Plan
- Development approach

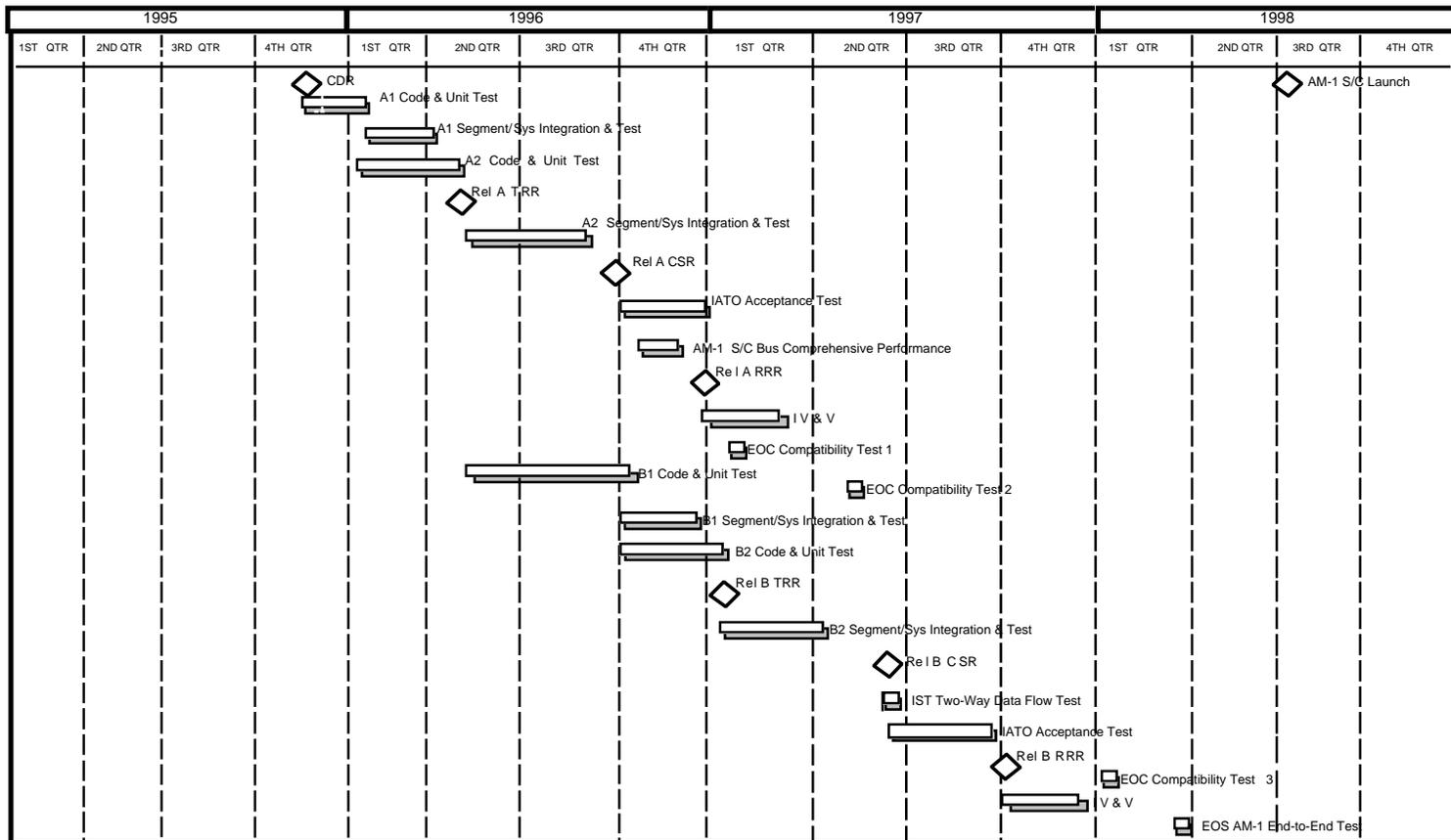
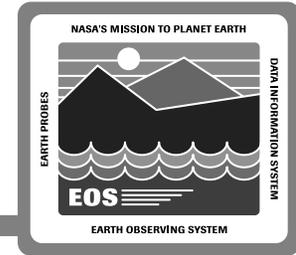
### Testing

- Test approach
- Test organization

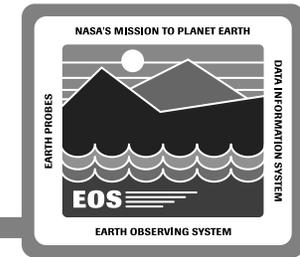
# FOS Development



# FOS Development Schedule

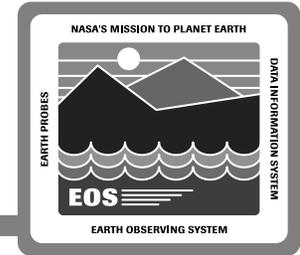


# FOS Team Responsibilities



Organization	Development				Operations	
	Code & Unit Testing	Integration	System/ Segment Testing	Acceptance Testing	IV&V Testing Pre-Ops	OPS Testing
<b>FOS Segment Development</b>	Perform	Perform	Support	Support	Support	Support
<b>FOS System/ Segment Test</b>	Monitor	Witness	Perform	Support	Support	Support
<b>IATO</b>			Witness	Perform	Support	Support
<b>FOT</b>	<b>Support</b>	Support	Support	Support	Support	Perform
<b>IV&amp;V Contractor</b>			Monitor	Witness	Perform	
<b>Quality Office</b>	Monitor	Monitor	Witness	Witness	Support	Support
<b>Configuration &amp; Data Mgmt</b>	Support	Support	Support	Support	Support	Support

# FOS Release Planning



## Build/Thread Approach

### Define Segment-level scenarios

Segment-level scenarios are comprised of a sequence of threads that transcend individual subsystems.

Develop, integrate, and test FOS in sequence as defined build/thread

## Release Definition

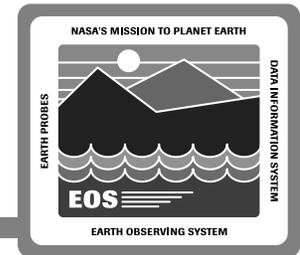
### Define general objective for each Release

Define infrastructure components

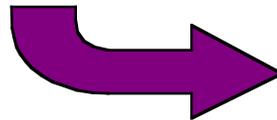
Define development dependencies  
- predecessor/successor relationships

Partition threads according to Releases and internal builds

# FOS Release Planning Drivers



**Release A**



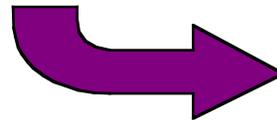
**Establish infrastructure early that is required for development**

**Establish basic connectivity internally and with external interfaces**

**Demonstrate basic telemetry and command processing**

**Support early interface testing and Spacecraft Tests (e.g., EOC Compatibility Test, 1/97)**

**Release B**

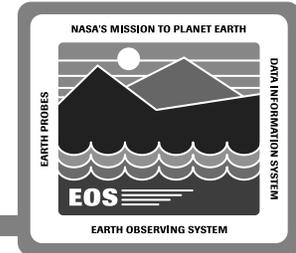


**Expand system based on building blocks established in Release A**

**Support Spacecraft Training, Simulations**

**Support AM-1 launch**

# FOS Release A Capabilities



## Scheduling

### Planning & Scheduling

BAP Definer Tool  
Detailed Activity Schedule

### Command Mgt

Uplink Load Generation  
ATC Load Generation  
Ground Sched Generation  
Load Manager  
Microprocessor Loads  
RTS Load Builder

## Real-Time

### Resource Mgt

String Manager  
Command Authority  
Ground Control Privilege

### Telemetry

Decom Engine  
Mirrored Telemetry  
Parameter Server

### Command

Cmd Validation, Generation  
and Uplink

### Real-Time Contact Mgt

NCC GCMRs, NCC Test Msg

## Analysis

### FUI

Analysis Request Tool  
Analysis Report Generation

### Analysis

Request Manager  
Dataset Generation  
Expert Advisor

### Data Mgt

Data Archive

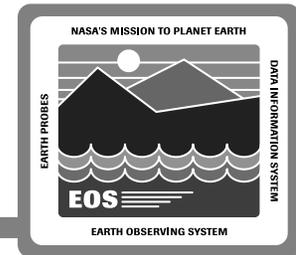
## Support Services

ECL	Screen Management	Display Builder	Time Selector	DB Generation
Events	Window Manipulation	Procedure Builder	Quick Analysis	(Activities, Cmd, Tlm)
Help	Status Window	Room Builder	Ground Script Control	DB Edit

## Infrastructure

Message Passing	Name Services	Network Services	Interface Connectivity
Security Services	Directory Services	Management Services	Time Services

# FOS Release B Capabilities



## Scheduling

### Planning & Scheduling

Timeline Tool  
 Planning Aids Mgt  
 TDRS Scheduling  
 Activity Constraint Check  
 What-If

### Command Mgt

Cmd Constraint Check  
 Load Catalog  
 Load Reports  
 Generate Patch Loads  
 Memory Mgt/Compares

## Real-Time

### Resource Mgt

String Cfg Change Requests  
 Resource Monitor  
 Failure Recovery

### Telemetry

Derived Parameters, Selective  
 Decom, Tailored Telemetry  
 S/C State Check

Memory Dump

### Command

Cmd Verification  
 Load Processing

### Real-Time Contact Mgt

NCC ODMs, EDOS CODAs

## Analysis

### FUI

Standing Orders

### Analysis

Clock Correlation  
 Solid State Recorder Mgt  
 S/C Activity Log  
 System Statistics  
 User-Defined Algorithms

### Data Mgt

Triggers  
 Queue Mgr

## Support Services

User Customization  
 E-Mail  
 Document Reader

Report Browser/Editor Tool  
 Quick Msg  
 Display Builder (schematics)

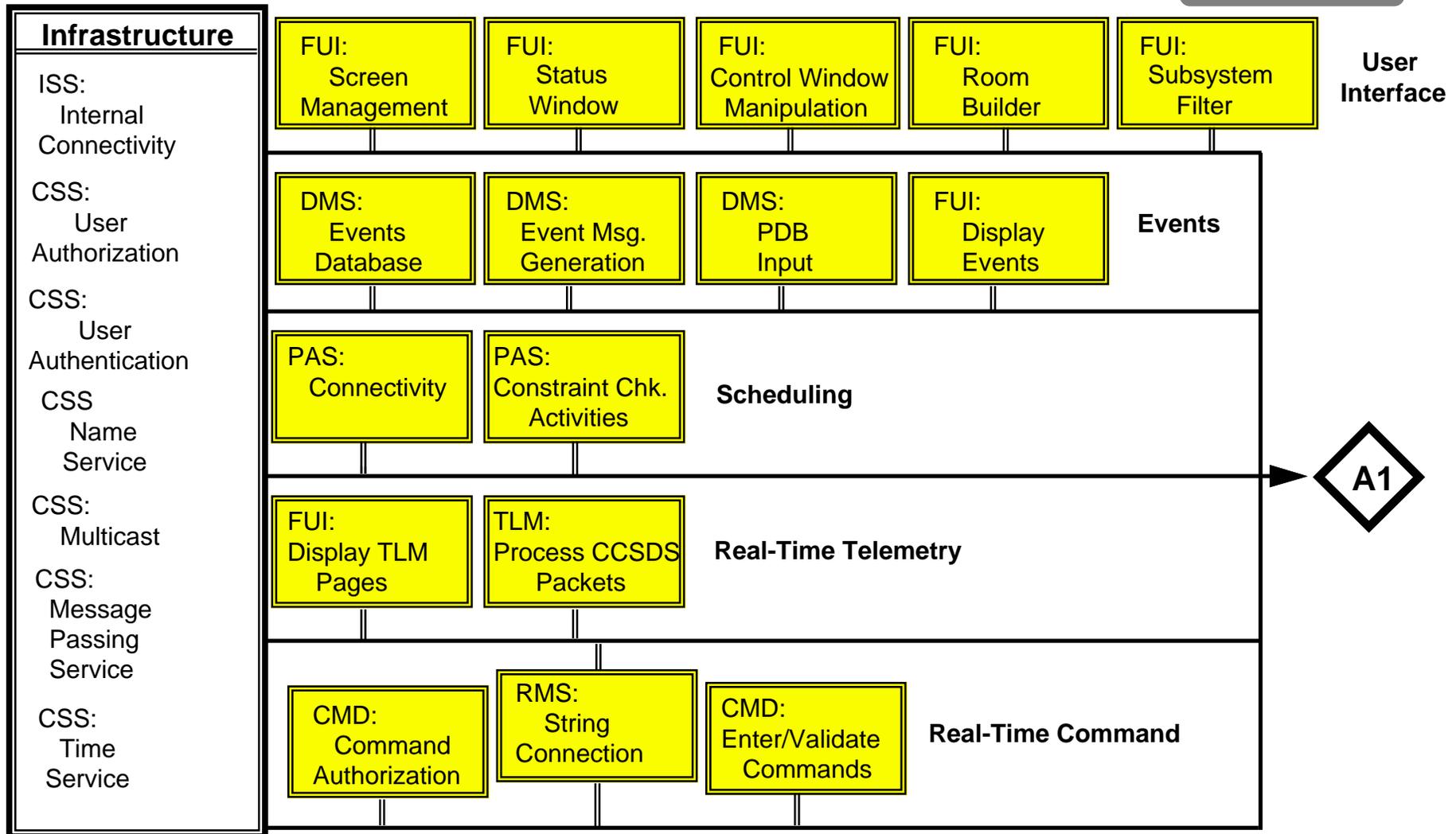
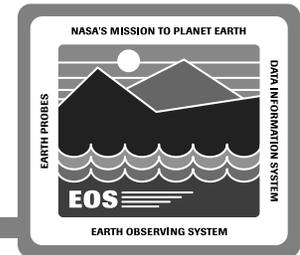
DB Backup/Restore  
 Archive Events  
 Event History Tool

DB Reporting  
 DB Maintenance  
 Data Mover

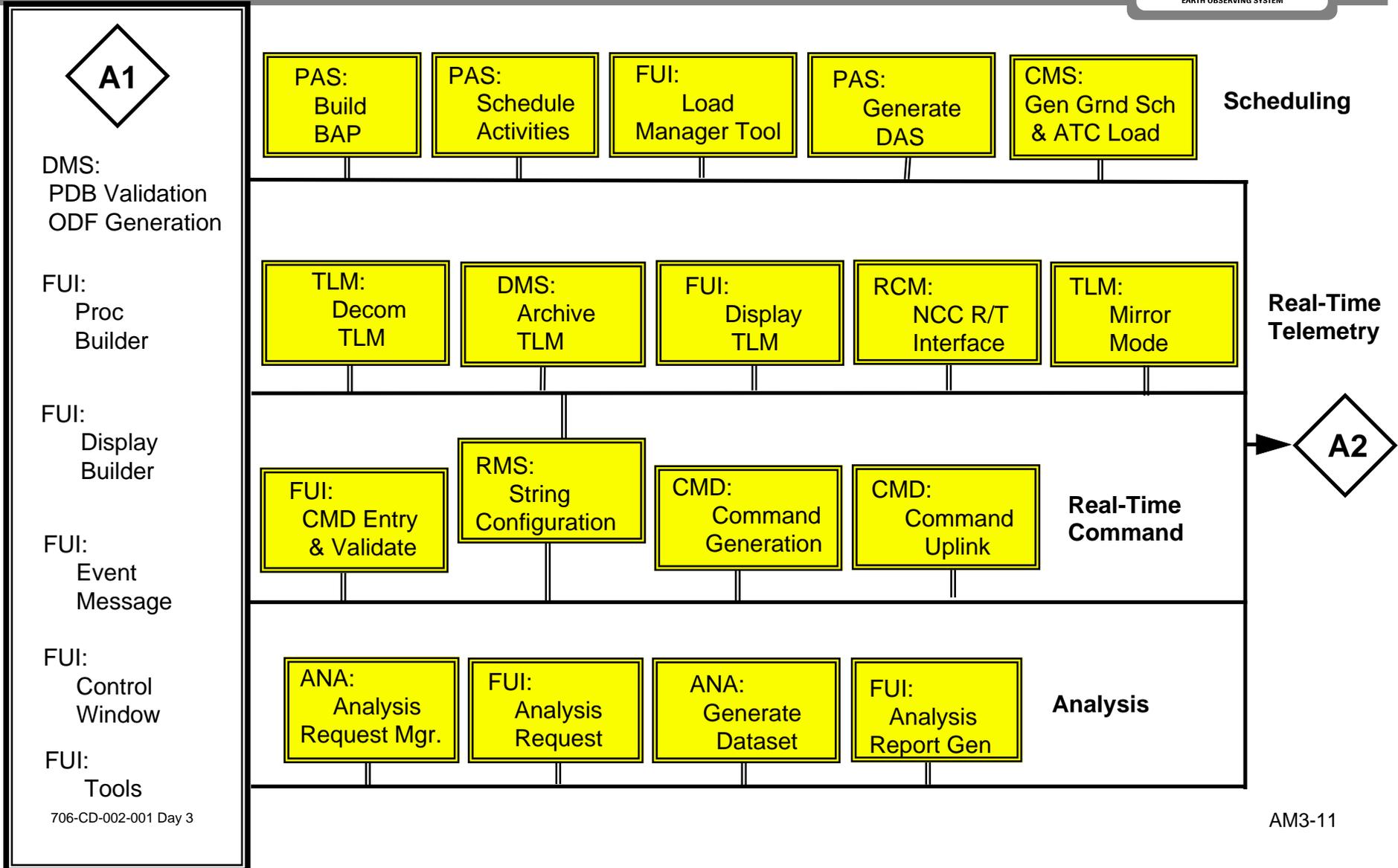
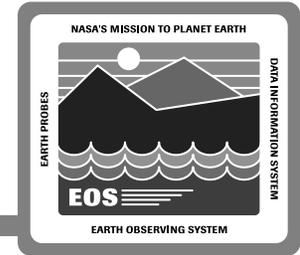
## Infrastructure

Release A Capabilities  
 Additional Management and Communication Services

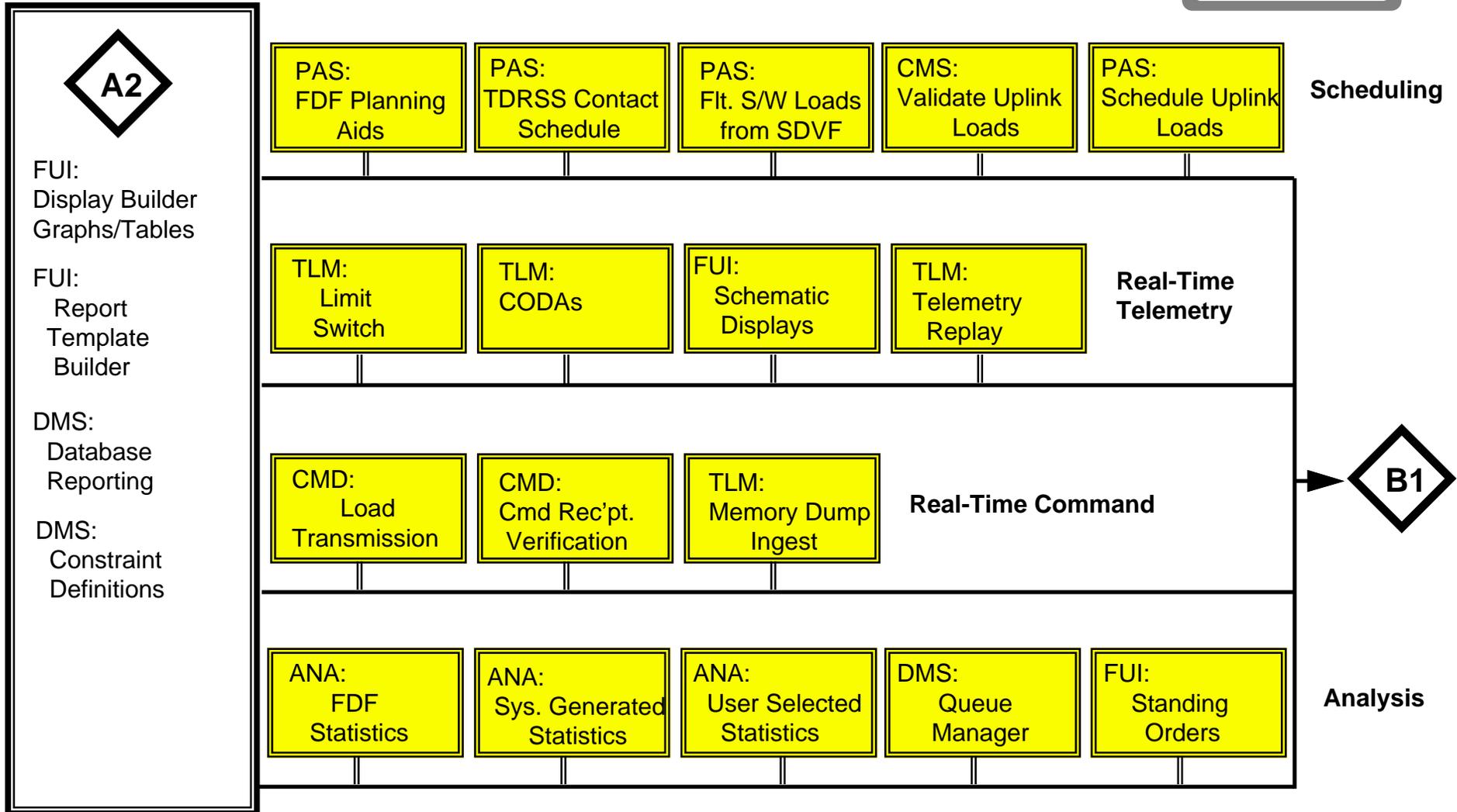
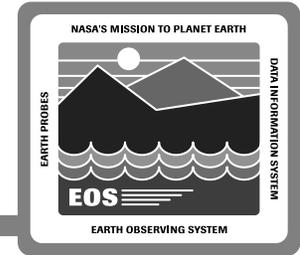
# FOS A1 Integration Thread Highlights



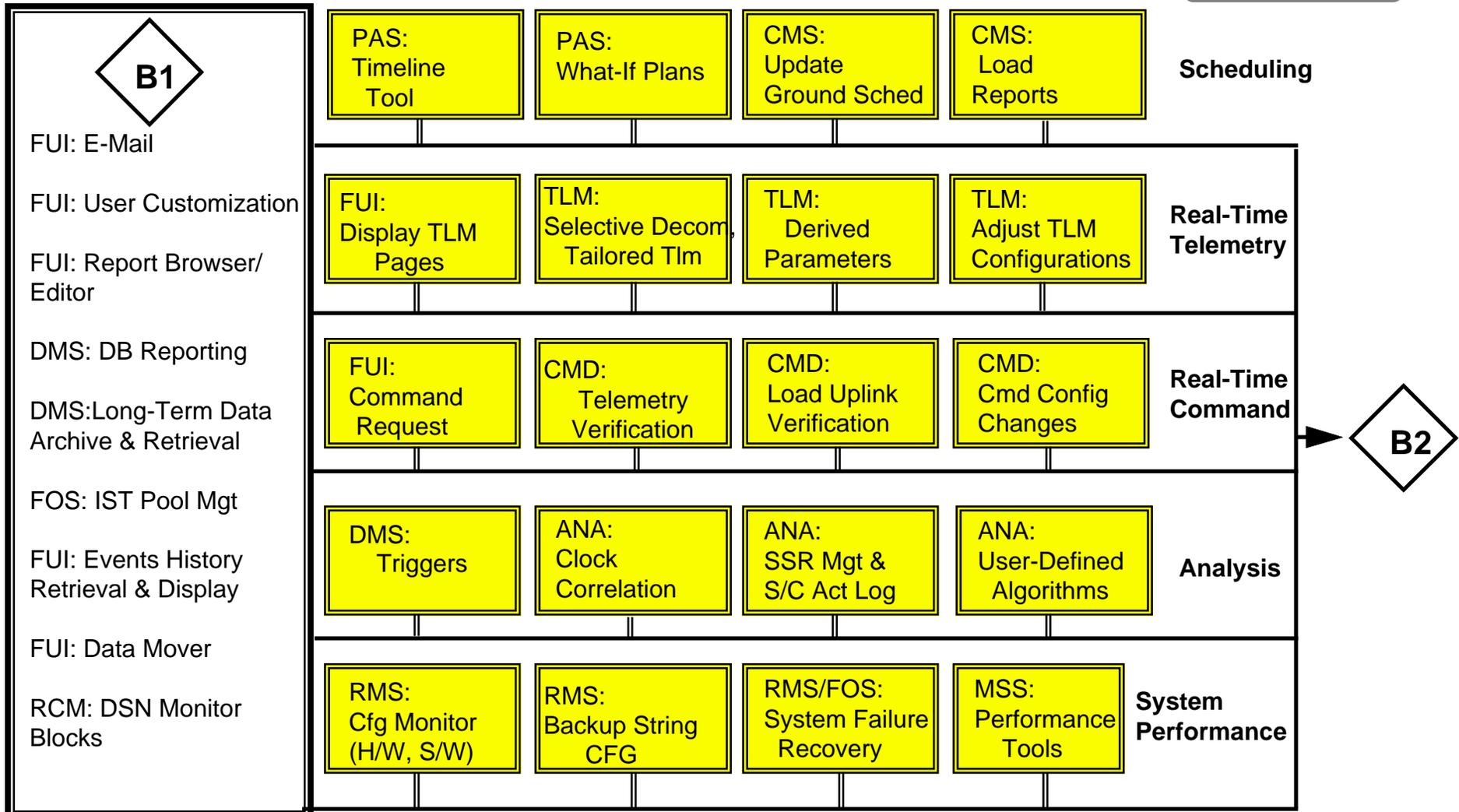
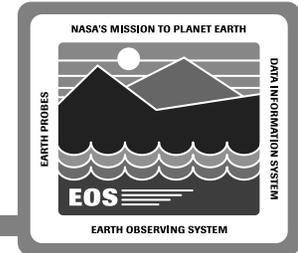
# FOS A2 Integration Thread Highlights



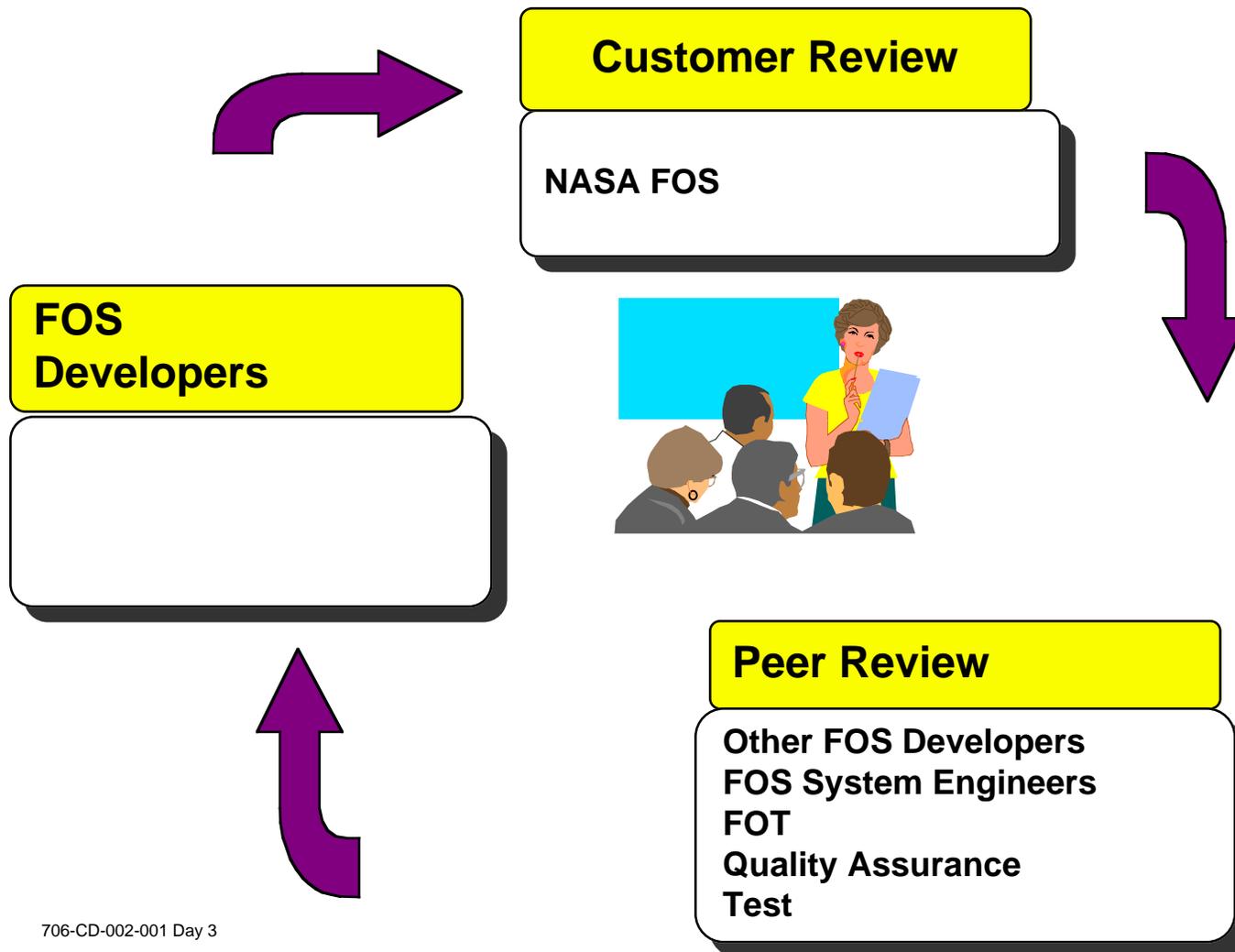
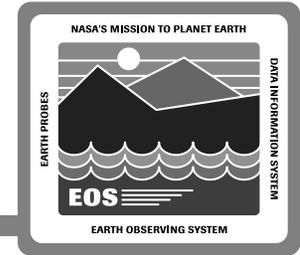
# FOS B1 Integration Thread Highlights



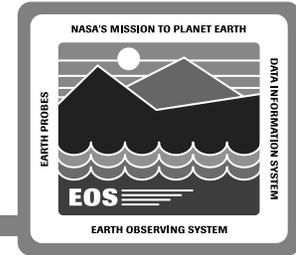
# FOS B2 Integration Thread Highlights



# FOS Code Walkthroughs



# FOS Development Metrics



## FOS Development Metrics Have Been Planned for Release A Timeframe

- Release B metrics planned at build level
- Test Defect metrics also will be tracked

## Code and Unit Test Metrics Based on Components

- Component is defined as a functional part of the FOS system
- Examples: Decom, Schedule Model, Resource Model

## FOS Development Metrics for Release A

### Code Components:

11/95	12/95	1/96	2/96	3/96	4/96	5/96
14	19	6	4	19	10	0

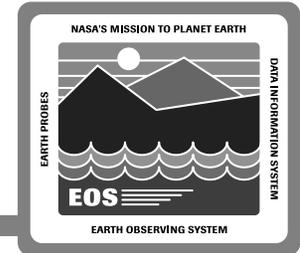
### Unit Test Components:

11/95	12/95	1/96	2/96	3/96	4/96	5/96
2	7	20	8	7	19	14

### Test Cases:

11/95	12/95	1/96	2/96	3/96	4/96	5/96	6/96	7/96	8/96
0	0	0	3	15	3	0	10	40	17

# FOS Integration Approach



## Integration Team

**Integration Lead**  
**FOS System Engineers**  
**FOS Development Team**  
**FOS Test Team (monitor)**



## Integration Approach

**Subsystem components are integrated after unit testing is complete.**

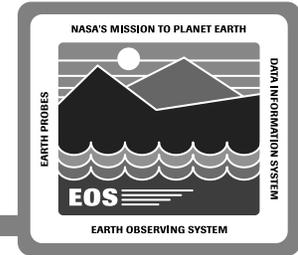
**Functional threads are integrated and tested.**

**Dry-runs of FOS tests are conducted for each test defined for the internal build.**

**Regression testing is performed after the first build.**

**Performance testing is conducted throughout the integration phases.**

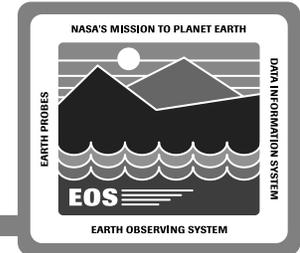
# Configuration Management



## FOS Configuration Environment Summary

<b>Configuration Environment</b>	<b>User</b>	<b>Purpose</b>
<b>Development</b>	<b>Software Developers</b>	<b>Perform software development including code and unit test.</b>
<b>Test</b>	<b>Integration Team Test Team</b>	<b>Perform integration of components that have completed unit test.</b>  <b>Perform formal testing of integrated software components.</b>
<b>Operations</b>	<b>FOT/IOT</b>	<b>Execution of FOS operational software which has successfully completed testing.</b>

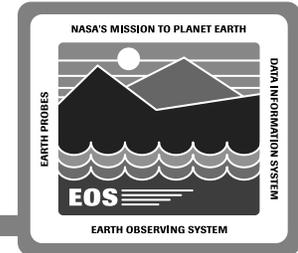
# Configuration Management



## FOS Configuration Environments

- **Each configuration environment is uniquely maintained**
  - **CM library maintained separately for each environment**
  - **Software migrates from one environment to the next as it proceeds through the development life cycle**
- **Different user groups can perform their tasks independent of other groups**
  - **Sufficient development hardware to physically separate different environment users**
  - **Logical names used to point users to correct set of directories and files**
- **Supports concurrent software development life cycle activities**
  - **Concurrent code/unit test and integration and test**
  - **Concurrent operations and test/sustaining engineering**

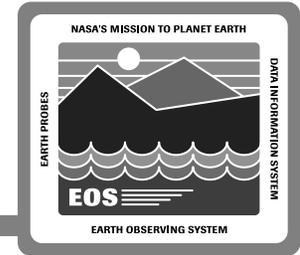
# Configuration Management



## FOS Build Summary

Release Baseline	Release Type	Release Activity
A1	Internal	FOS will integrate and dry-run A1 tests
A2	Incremental	Integrate and dry-run A2 tests Formally test A1 and A2 test cases Release A delivered to GSFC
B1	Internal	Regression dry-run Release A functionality Integrate and dry-run B1 test cases
B2	Formal	Integrate and dry-run B2 tests Regression test Release A functionality Formally test B1 and B2 test cases Release B delivered to GSFC

# CM Tool: ClearCase



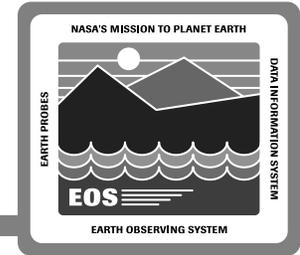
## Distributed Software Library Tool

- **Used for control of large software development projects**
  - **Complex, distributed, heterogeneous**
- **Customized by locally developed scripts executed at key transition points in the CM process**
  - **Segment/System test**
  - **Acceptance test**
  - **Post-Release Readiness Review (RRR)**

## Testing

- **Software entering each test process goes under CM control at that level**
- **Each level of testing will maintain and control a separate CM branch for software undergoing test**
- **Test tools, data, scripts are under CM via ClearCase**

# Non-Conformance Reports (NCR)



**Written By Test Teams, Operations Team, Developers**

- **NCR forms available via Internet**

**NCRs**

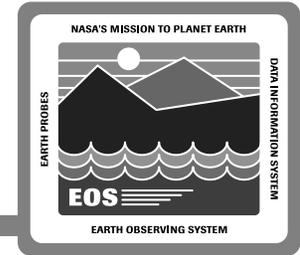
- **Internal NCRs written for Internal Releases**
  - **Tracked internally by FOS Team**
  - **Closed by Test Team**
- **NCRs written after Release A and Release B baselines have completed FOS system/segment test**

**NCRs are Submitted to the FOS NCR Board**

**NCR Board**

- **NASA FOS Program Manager**
- **ECS FOS Program Manager**
- **ECS FOS Technical and Test**
- **Flight Projects and FOT**

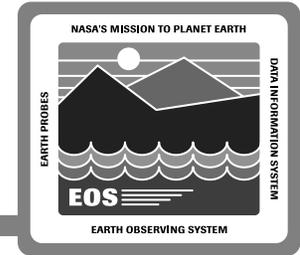
# Non-Conformance Reports (NCR)



## NCR Disposition

- NCRs categorized as either high priority or low priority
- High priority NCRs are fixed in the current software baseline
- Low priority NCRs are fixed in the next software baseline
- Example: FOS Release A
  - High priority NCRs are defined as those functions that need to be corrected to support AM-1 testing in the pre-Release B timeframe
  - Low priority NCRs will be corrected and included in the Release B baseline and verified during the Release B segment/system test

# Non-Conformance Reporting Tool: DDTS



## Defect Tracking System

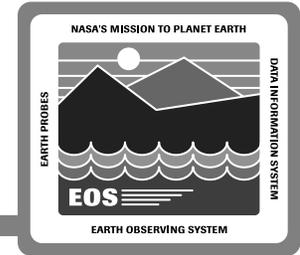
- Capable of working in an environment
- Multiple subsystems (projects) are easily maintained
- Standard and custom reports
- Text and graphics
- Interacts with ClearCase to link CM and NCRA processes

## Testing

- Each test activity is declared separate DDTS projects to track problems
- Unresolved problems are transitioned to the next DDTS project or test activity
- NCR system serves as a source for test metrics

# Non-Conformance Reporting Tool: DDTs

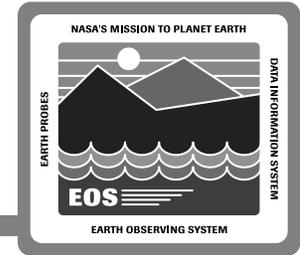
---



## User Access to NCR Reports

- Available via Internet
- Internet address: <http://newsroom/sit/ddts/ddts.html>
- User can query the DDTs database
  - Priority
  - User submittal name
  - Type (subsystem e.g., Telemetry)

# Training



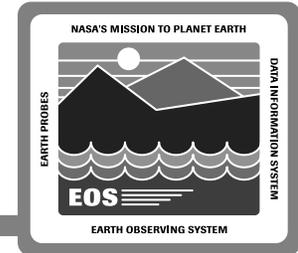
## Documentation

- **User's Manual provides information for the Flight Ops Team and the Instrument Ops Teams describing how to use the system**
- **Scheduled delivery date is 2 weeks prior to the Consent to Ship Review**
  - **Release A draft -> October 1996**
  - **Release B -> April 1997**

## Training at GSFC

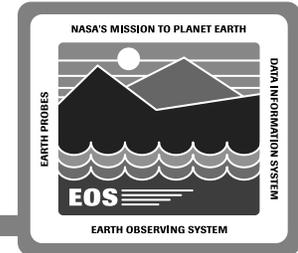
- **Date: tentatively scheduled for fall 1997**
- **Training provided to FOS users describing how to use the FOS**
- **Training material will be packaged for the FOT and the IOT**
  - **IOT is a direct subset of the FOT training**

# Key FOS Dependencies



Organization	Dependency	Date
AM-1 Project	AM-1 DB (preliminary) AM-1 schedulable resources AM-1 DB (update) Subsystem constraint definitions (samples) Subsystem activity definitions (samples)	11/95 11/95 5/96 5/96 5/96
IOTs	CRC algorithms for microprocessor loads Instrument constraint definitions (samples) Instrument activity definitions (samples)	2/96 5/96 5/96

# Key FOS Dependencies



Organization	Dependency	Date
User Planning System (UPS)	Delivery to FOS	2/96
EDOS	ICD (preliminary)	10/95
	ICD (final)	12/95
FDF	ICD (preliminary)	10/95
	FDF test data	1/96
	ICD (final)	5/96
SSIM	ICD (prelim)	10/95
	ICD (final)	12/95
EBnet	EBnet ICD (final)	1/31/96
	EBnet connectivity	8/96