Systems Testing

Michael Starch
NASA Jet Propulsion Laboratory
October 18, 2023

Jet Propulsion Laboratory
California Institute of Technology

Overview

- Introduction to Systems Testing
  - What is system testing?
  - Why do we do system testing?
  - System testing tools
- Types of System Testing
  - Manual systems testing
  - Automatic system testing
  - Continuous integration
- System Testing with F´

Introduction to System Testing

Systems Testing

- Testing software within the full running system

- Includes flight-like hardware and ground support

- Also called *integration testing* and similar to *regression testing*

- "Test as you fly, fly as you test"
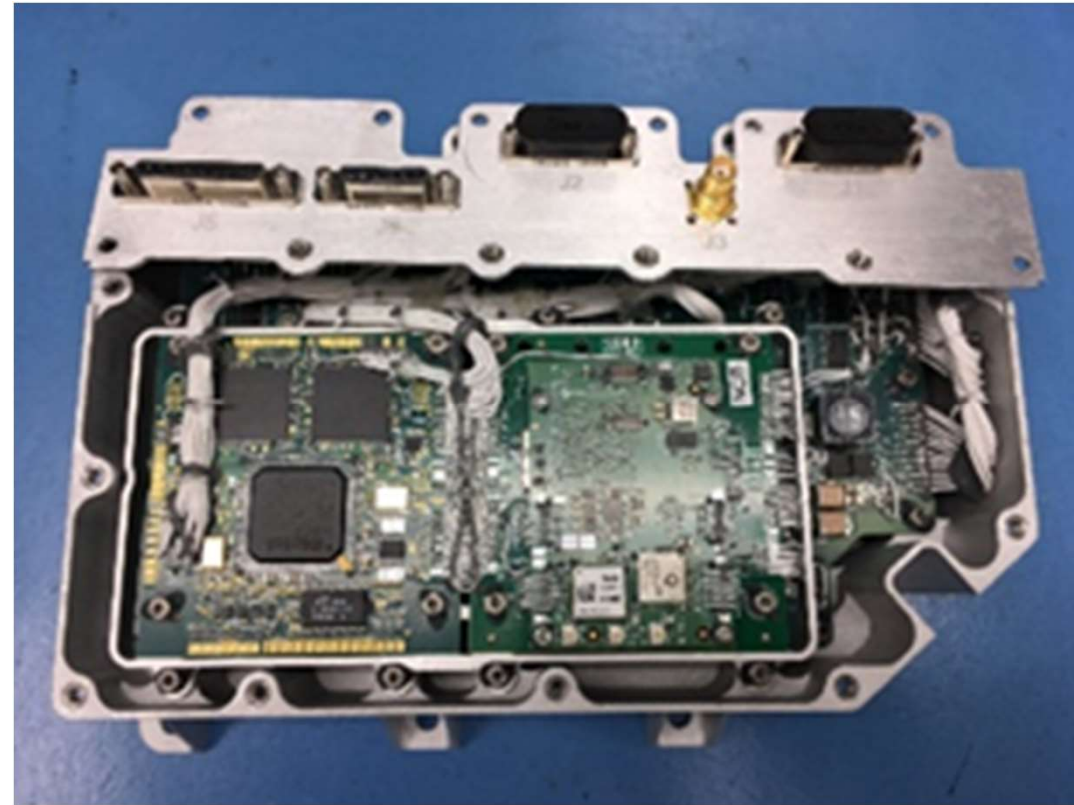
JPL/Caltech

Why Systems Testing?

- A system is more than the sum of its parts

- Some requirements can only be verified at system level

- Hardware operations need testing on hardware
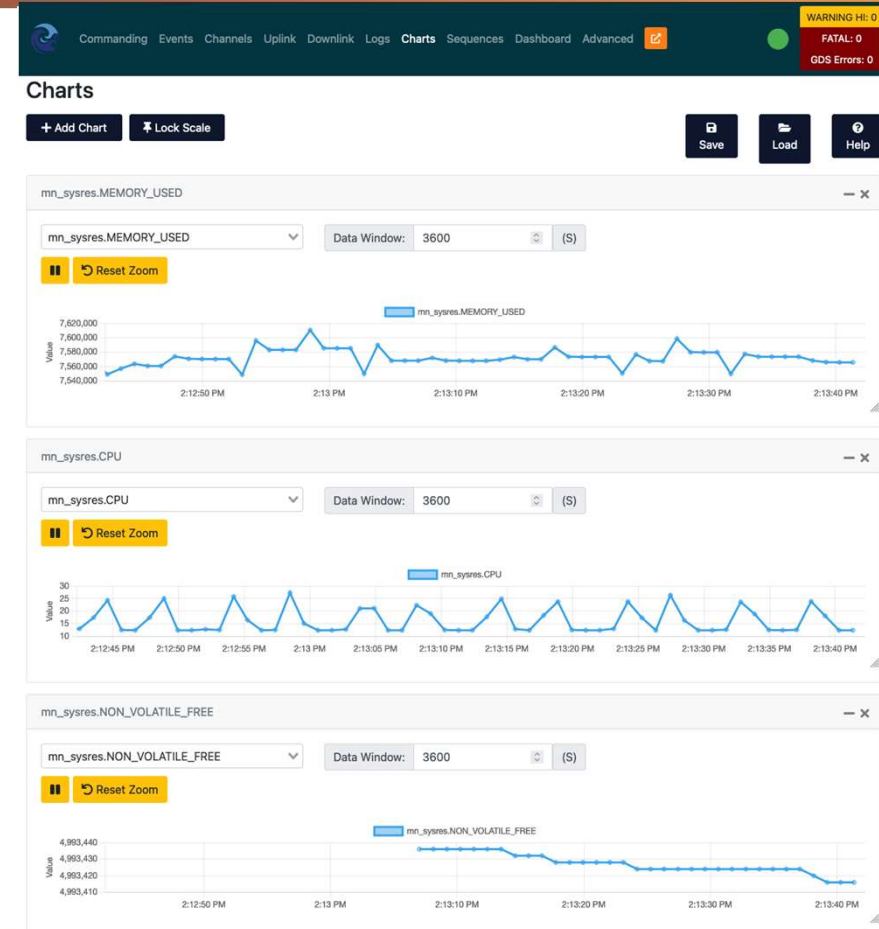
- "Eat our own dog food"

Public Domain

System Testing Tools

- Ground Data Systems
  - Operate spacecraft from earth
  - May be used to conduct testing
- Testing Frameworks
  - Allow for automated testing
  - Reporting, verification, reproducibility
- Hardware Simulations
  - Allows system testing without physical hardware
- Command Sequences
  - How the flight spacecraft is operated
  - May be used to construct tests

Types of System Testing

Manual Systems Testing

- Operating the spacecraft by-hand to verify requirements

- Running sequences and commands and verifying results

- Ground system is used to send commands

- More complete tests often require full test team
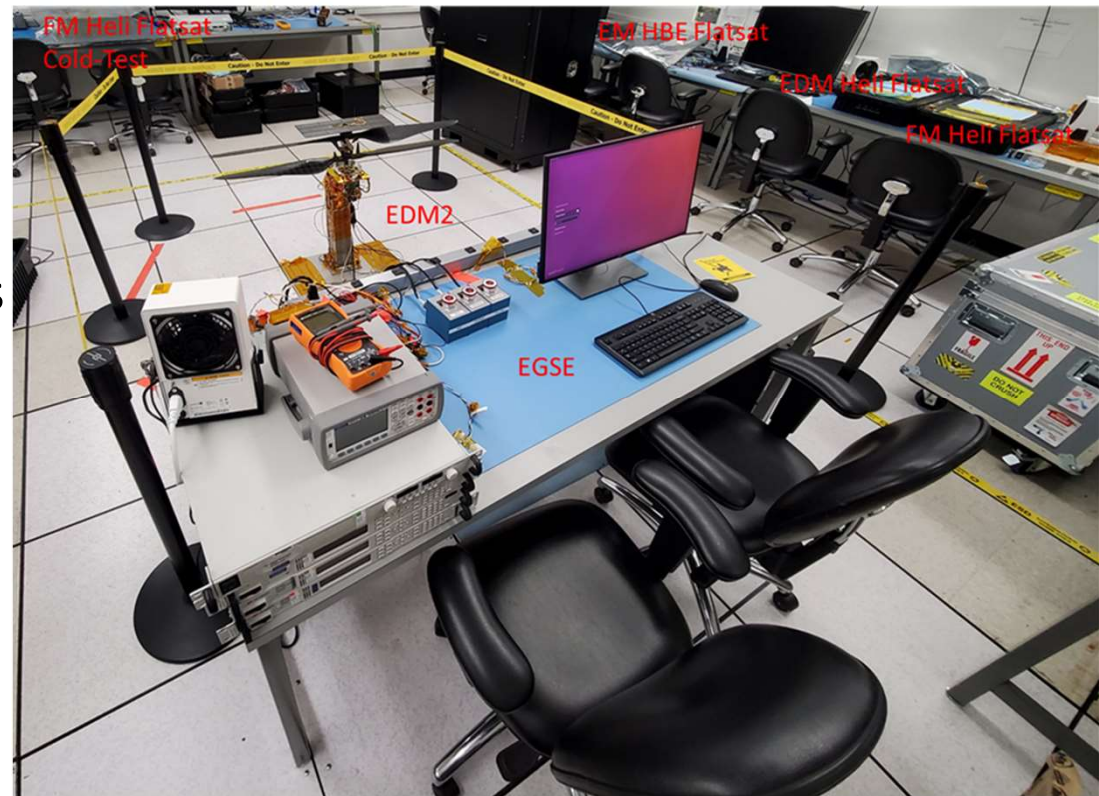
JPL/Caltech

Automated System Testing

- System testing performed by software

- Uses automated testing support from ground station

- Automatically logs data, results, and verification products

- May not require dedicated test team

Continuous Integration

- Extension of automatic system testing

- Triggered by software change request or other criteria

- Requires no human interaction

- Implements full regression testing

System Testing with F´

F´ GDS

- Provides GDS capabilities out-of-the-box for F´ projects
- Designed for system testing early in project development
- Enables developers to:
  - Send commands
  - Receive events and telemetry
  - Uplink and downlink files
  - Design and upload sequences
  - Compose custom dashboards

Jet Propulsion Laboratory
California Institute of Technology

F´ Integration Test Framework

- Framework designed for integration testing with F´ GDS
- Allows users to write automated tests in Python
- Tests dispatch commands and assert on events and telemetry
- Creates excel test logs
- Can be run within continuous integration system

```python
def test_active_logger_filter(self):
    self.set_default_filters()
    try:
        cmd_events = self.api.get_event_pred(severity=EventSeverity.COMMAND)
        actHI_events = self.api.get_event_pred(severity=EventSeverity.ACTIVITY_HI)
        pred = predicates.greater_than(0)
        zero = predicates.equal_to(0)
        # Drain time for dispatch events
        time.sleep(10)

        self.assert_command("cmdDisp.CMD_NO_OP")
        self.assert_command("cmdDisp.CMD_NO_OP")

        time.sleep(0.5)

        self.api.assert_event_count(pred, cmd_events)
        self.api.assert_event_count(pred, actHI_events)

        self.set_event_filter(self.FilterSeverity.COMMAND, False)
        # Drain time for dispatch events
        time.sleep(10)
        self.api.clear_histories()
        self.api.send_command("cmdDisp.CMD_NO_OP")
        self.api.send_command("cmdDisp.CMD_NO_OP")

        time.sleep(0.5)

        self.api.assert_event_count(zero, cmd_events)
        self.api.assert_event_count(pred, actHI_events)
    finally:
        self.set_default_filters()
```

jpl.nasa.gov