

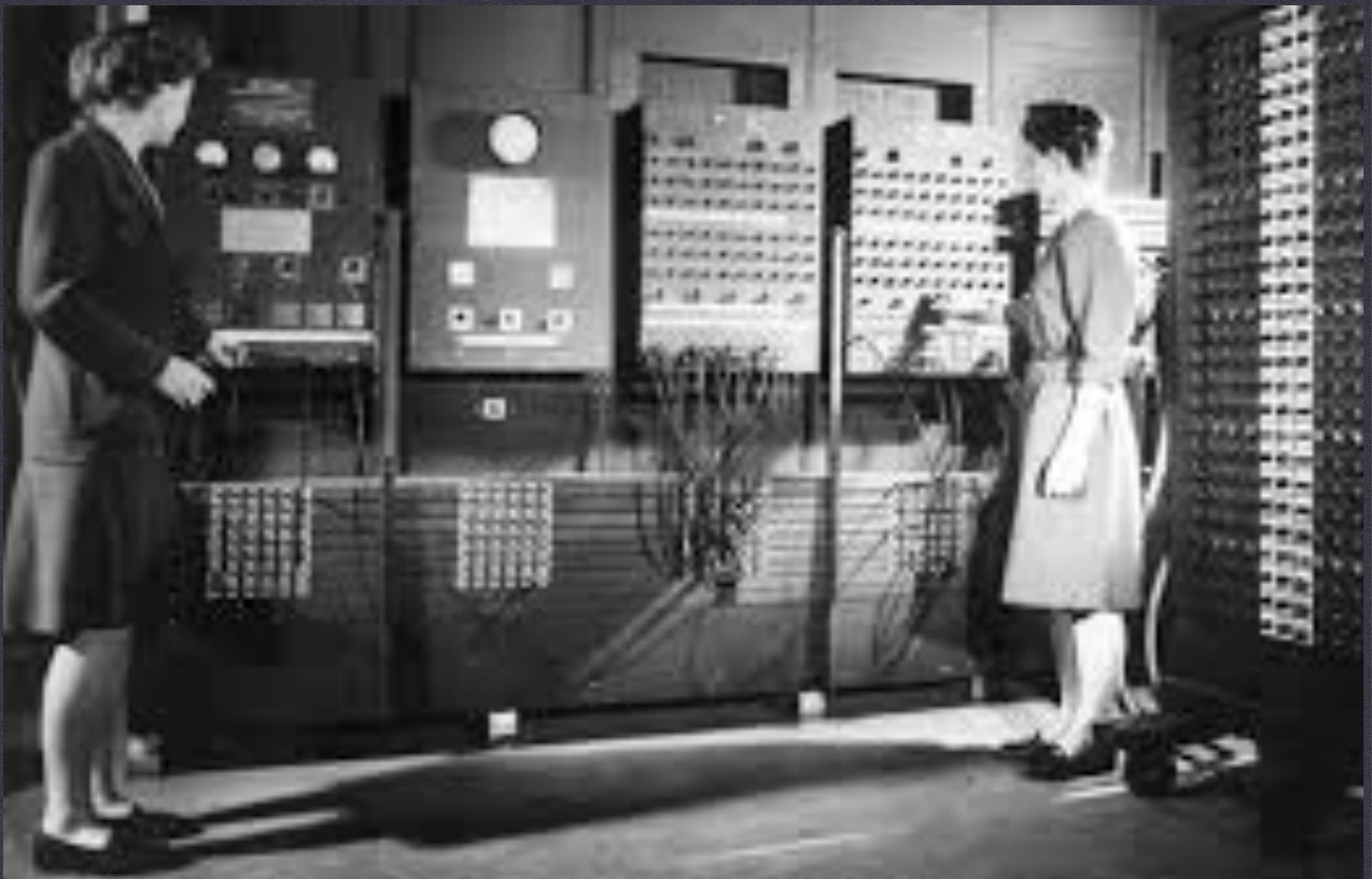
LIVING FLIGHT SOFTWARE

A 40 YEAR
INTERPLANETARY
VOYAGE

David E. Smyth
*Principal Flight Systems
Software Engineer*

 MILLENNIUM
SPACE SYSTEMS





THERE WERE PROGRAMMERS BEFORE ME
I MAY BE OLD, BUT I AM NOT THAT OLD

THE FIRST MICROPROCESSORS

1972: PPS-4

128KIPS

4-BIT WORD

12-BIT (4K) ADDRESS SPACE

12-BIT STACK

1 GP REGISTER

1 MATH REGISTER

1 INTERRUPT

ASSEMBLY + MACHINE CODE





PROGRAMMERS IN 1972 WERE PROFESSIONALS

HAND PICKED FROM THE ENGINEERING RANKS

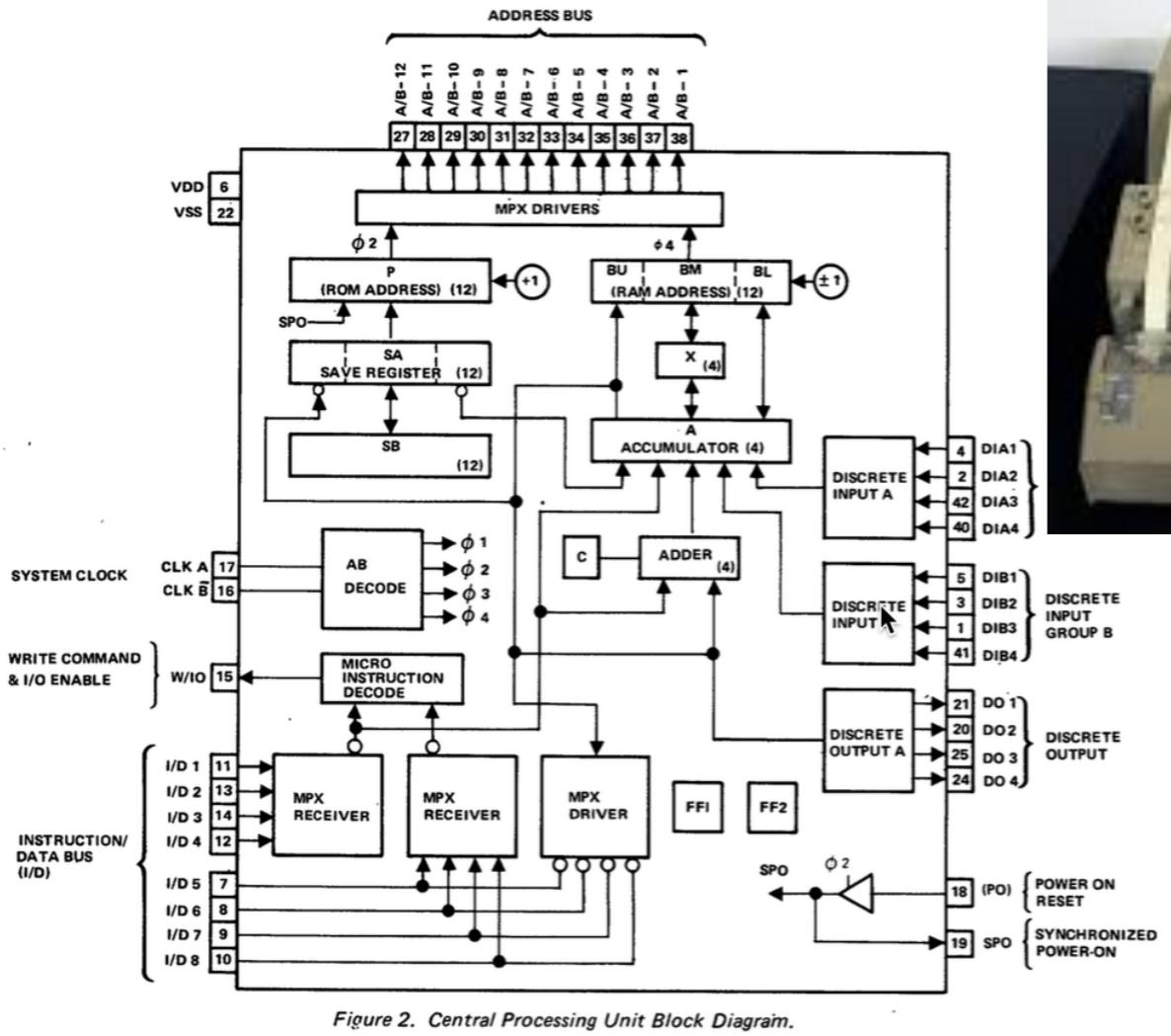
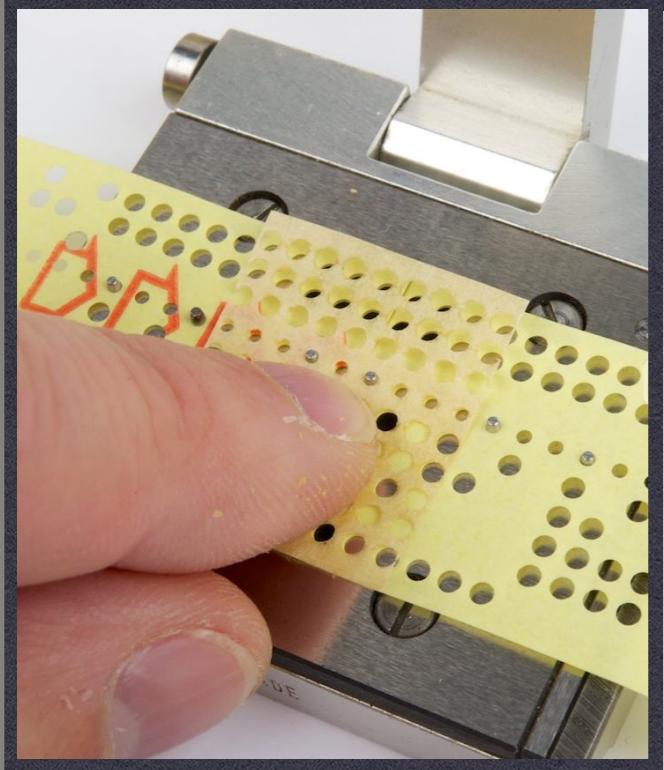
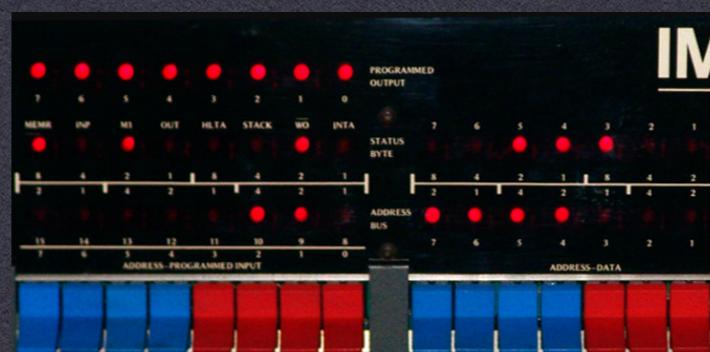


Figure 2. Central Processing Unit Block Diagram.



PROJECT

PROGRAMMING TOOLS

DIRECT MANIPULATION, SYMBOLIC, HIGH SPEED NETWORK,

CODE REVIEWS, FULL COLOR USER INTERFACE, WITH OBVIOUS RELIABILITY

DATE

1972

CLIENT

SR-71 DIGITAL FLIGHT CONTROL



**STACK OVERFLOW? WIKIPEDIA?
BUT PATTERNS DID EMERGE**

The Pattern

- * Identify the data
- * Decide types (4-bits for each BCD digit, fixed point)
- * Identify algorithms (get from the mathematician)
- * What data is operated on by what algorithm
 - * Data locality forced by limited addressing - data needs to be within 128 bytes of instruction
- * Select instruction that takes data from source, operates on it, and then stores in destination.
- * Repeat



PROJECT

OPERATING SYSTEMS AND PROGRAMMING LANGUAGES CP/M, RDOS, FLOPPIES, BASIC, AND MORE ASSEMBLER

DATE

LATE 1970'S

CLIENT

NASA, NATO, SONY

The Real-Time Pattern

- * Set priority based on time to deadline
 - * Closer the deadline, higher the priority
- * Corollary: high priority must have short run time

LIVE FREE OR DIE

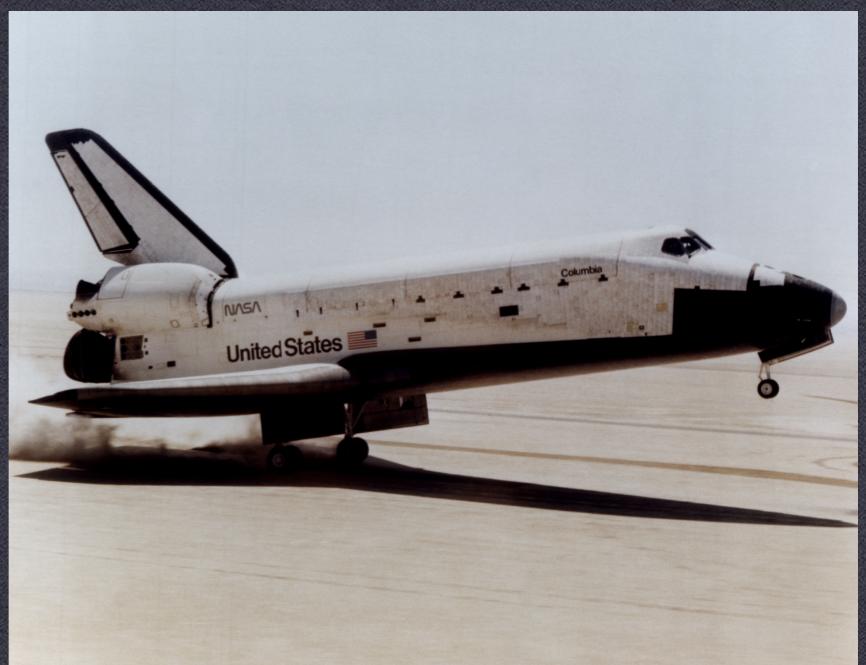
UNIX*

TRADEMARK OF BELL LABS*

17061

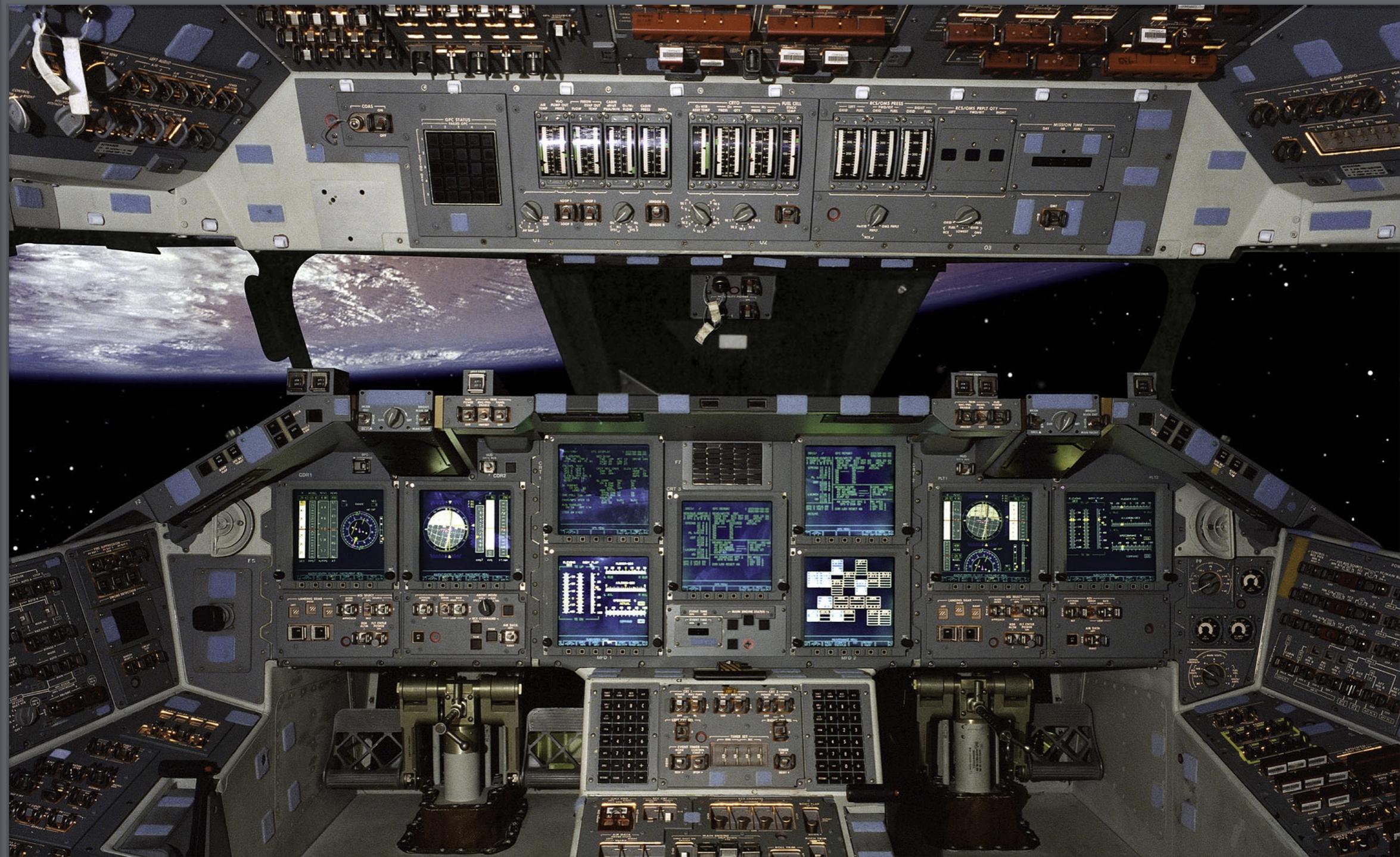
WHO KNOWS, THIS JUST MIGHT WORK

1979 SEVENTH EDITION, ALSO 2.7 BSD



SPACE SHUTTLE
1981

Wait for the next hand...



T-9mins



Lessons

- * Even when you are not allowed to refactor, if refactoring allows you to meet deadlines and pass tests, then seek forgiveness ...
- * Tests must be comprehensive, automated, and fully repeatable. Failing to repeat is a fail.



PROJECT **MARS PATHFINDER**
OBJECTS, AGENTS, AI, TELEMETRY, AND ORGANIZATION

DATE

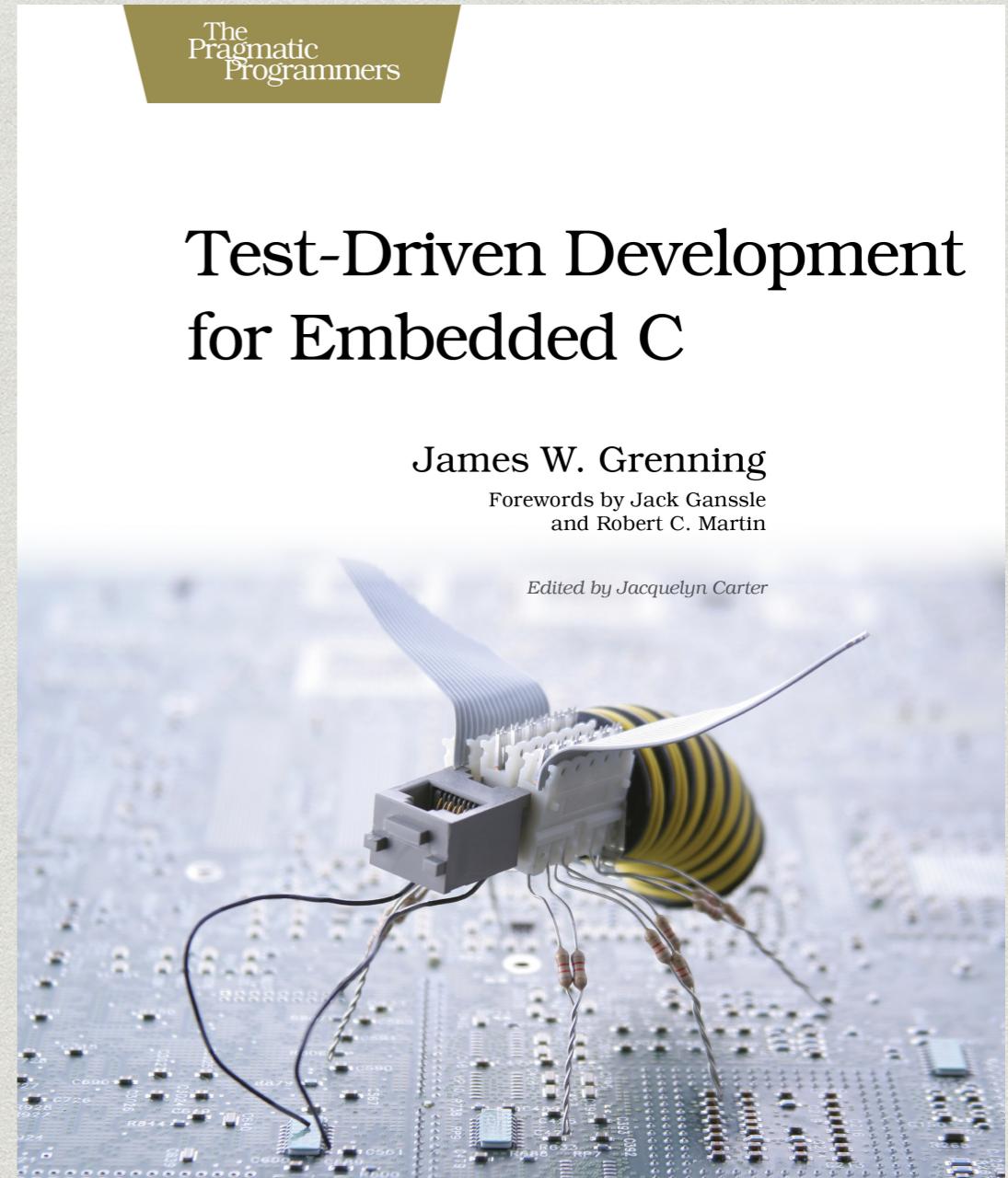
1997

CLIENT

MARS

Objects

- * Research at Siemens 1990-1992 comparing multiple projects found C a better object-oriented language than C++ or Obj-C
- * Object programming in C uses simple, common rules:
see



Agents and the Law

- * Simple model for safe, high performance multi-threading
- * Law of Demeter
 - * Each method operates only on the arguments and the receiving object's state.
 - * Similar to REST
- * DSL, implemented with Tcl, used to autogenerate messaging infrastructure

AI on Pathfinder

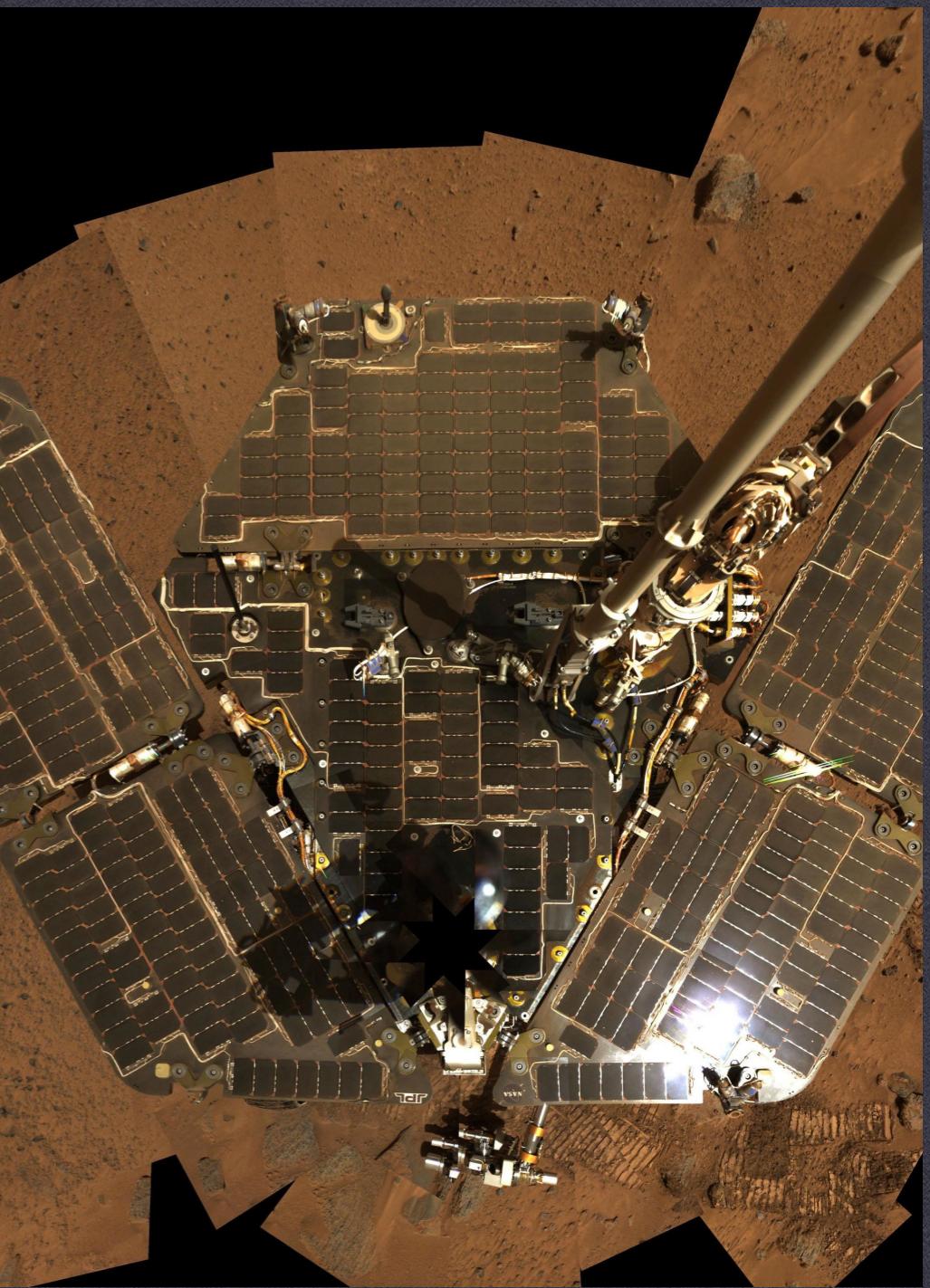
- * Had to be done on the sly...
- * Based on Subsumption by Rod Brooks, and coordination concepts by Maja Mataric.
- * Start at the bottom, and work up
- * As schedule allows, abstraction (apparent intelligence) increases. Base behaviors remain, but are subsumed and coordinated by “higher level” behaviors.
- * 300+ commands in command dictionary, but only about 20 used for mission.

Telemetry

- * Maximize the value of the telemetry
- * Don't send low value telemetry
 - * On-change, on-delta, ...
 - * Logger with Debug, Info, Warn, Fatal ...
- * Ack/nack to delete or re-xmit

Organization

- * Everybody knows the prioritized success criteria
 - * Use to make tactical decisions: i.e., When is this meeting over?
- * Everybody works to prioritized requirements
 - * Concentrate on what is needed first
- * Focus on flight artifacts
 - * Documentation minimized — refer to document from another mission, don't write new ones
 - * Each engineer does systems engineering — who are your interfaces, make sure you understand their needs, work those issues. No intermediaries.
- * Tony Spear was project manager, mentor.
 - * Tony also mentored CEO of Millennium Space Systems.



PROJECT

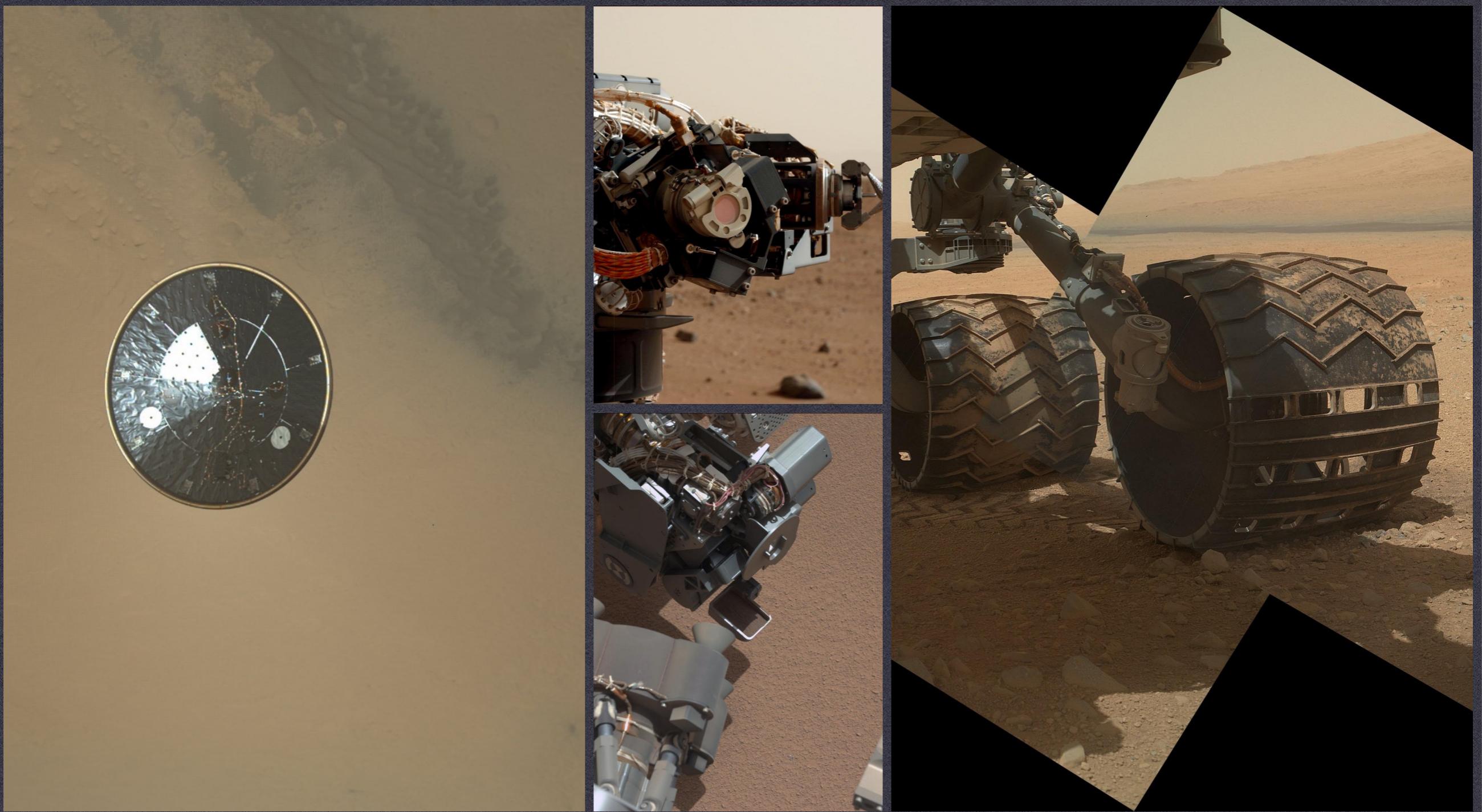
SPRIT AND OPPORTUNITY MPF V2

DATE

2004

CLIENT

MARS



PROJECT

CURIOSITY

HOW HARD CAN WE MAKE IT ON OURSELVES?

DATE

2012

CLIENT

MARS