

# Agile

Jay Trimble  
NASA Ames Research Center  
at Philips Agile 4 Ever  
October 2013

# NASA Centers



# My Background

- Missions
  - NASA Johnson Space Center, Houston
    - Shuttle Mission Control, Payloads
  - Jet Propulsion Lab
    - Robotic - Voyager Neptune
    - Shuttle - Space Radar Lab, Lead Ops Director
- Current
  - Mission Operations & Ground Data System Manager, Resource Prospector Lunar Rover



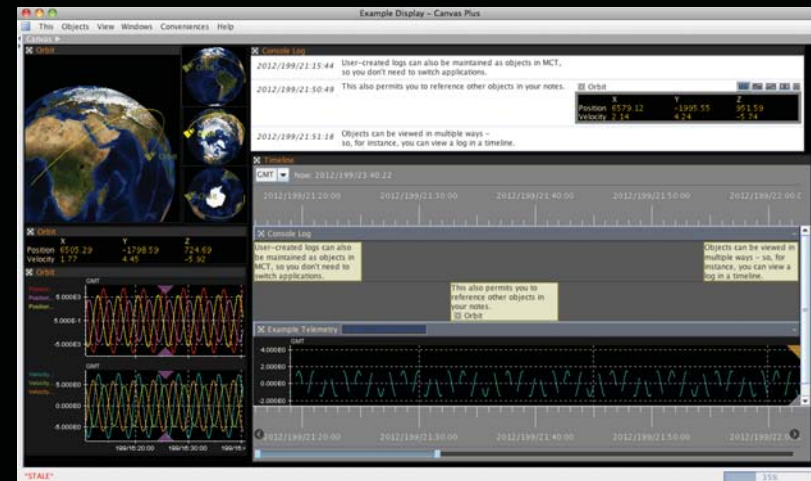
Space Radar Lab-I Ops Director



Internship in Mission Control  
(A long time ago)

# My Background

- Software Technology
- Human Centered Computing for Mars Rovers
- User centered technologies for mission control





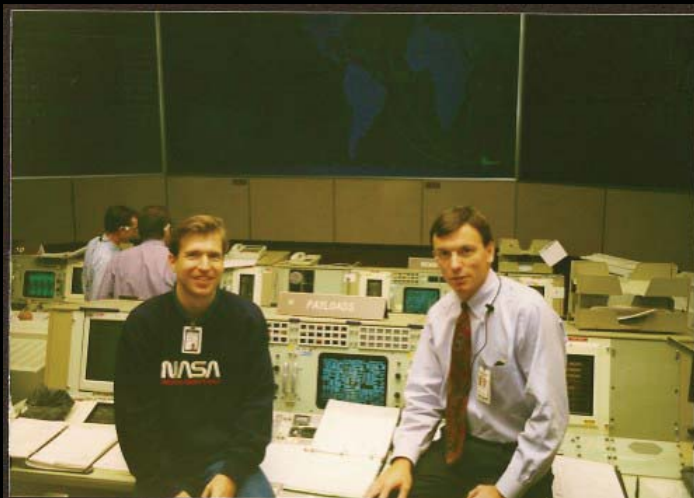
# Moments



Science Team after Mars  
Touchdown 2004



Shuttle Columbia



# One Story of Agile at NASA

- This is a bottom up story of how a group at NASA applied agile methods to software development for mission control
- This was approved, but not initiated by, management
- Beyond software
  - Taking agile beyond software we are applying agile and lean principles to the development of a Lunar Rover Mission Operations System

# The Project

- Our groups task was to build an architecture for mission control user applications, the primary focus being on developing interaction paradigms and technology for user composable software

# The Collaboration

- Design and Development Team at NASA Ames
- The Customer
  - Mission Control Users at NASA Johnson Space Center
- Using Participatory Design, we created an integrated team that included customer representation



# Issues and Mandates

- Some customers want a new product, others do not
- The product must have new capability, but must also not be disruptive within the organization
- Functional and visual connection to legacy product

# The Journey

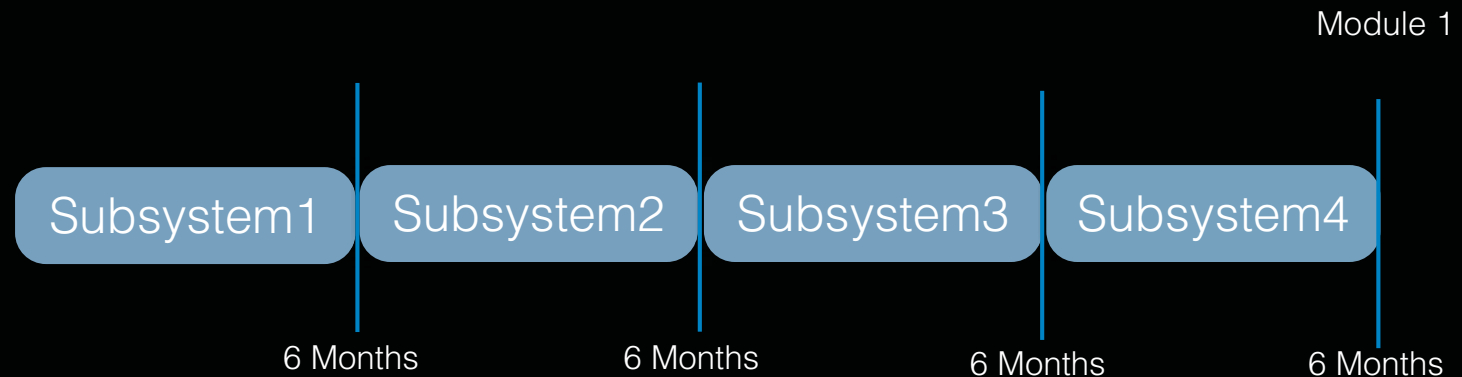
- We began with a six month software delivery cycle
- By iteratively fixing issues, we got the delivery cycle down to three weeks
- It took close to two years to complete the transition

# Time for Changes

- Fix the problems iteratively, without a broad proclamation of methodology, i.e. “we are going to be agile” or “we are going to be “lean”
- Just fix the problems

# Where we started

- Four six-month deliverables
- One User Experience Spec



# Issues we faced

- Long delivery cycle
- Difficult to manage feature prioritization and development, integration and testing
- Progress invisible to customer, lack of meaningful ongoing customer interaction to drive design
  - Mismatch in expectations between design/dev team and customer
- Difficult for the development team to know state of progress relative to goals
- Deliveries focus on subsystems rather than meaningful end user functionality
- Two-year final deliverable created a tendency to defer key issues

# The Team

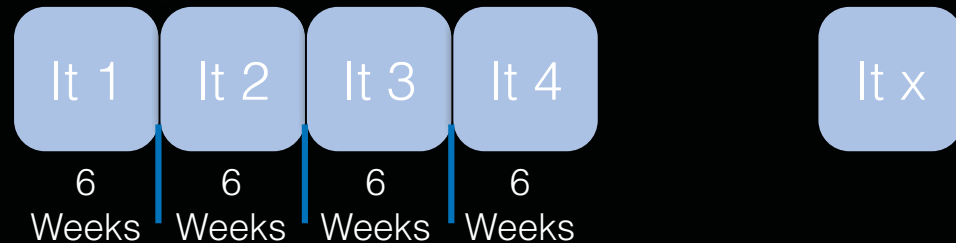
Traditional	Agile 1	Agile 2*
Developers 5-9	Developers 7	Developers 4
User Experience Design (2)	User Experience Design (2)	User Experience Design (1)
QA/Process Engineers (2)	QA/Process Engineers (2)	QA (.5)
Project Manager (1)	Project Manager (1)	Developers rotate PM role
Principle Investigator (Part Time)	Principle Investigator (Part Time)	Principle Investigator (Part Time)
Interns	Interns	Interns

\*Reduced Budget

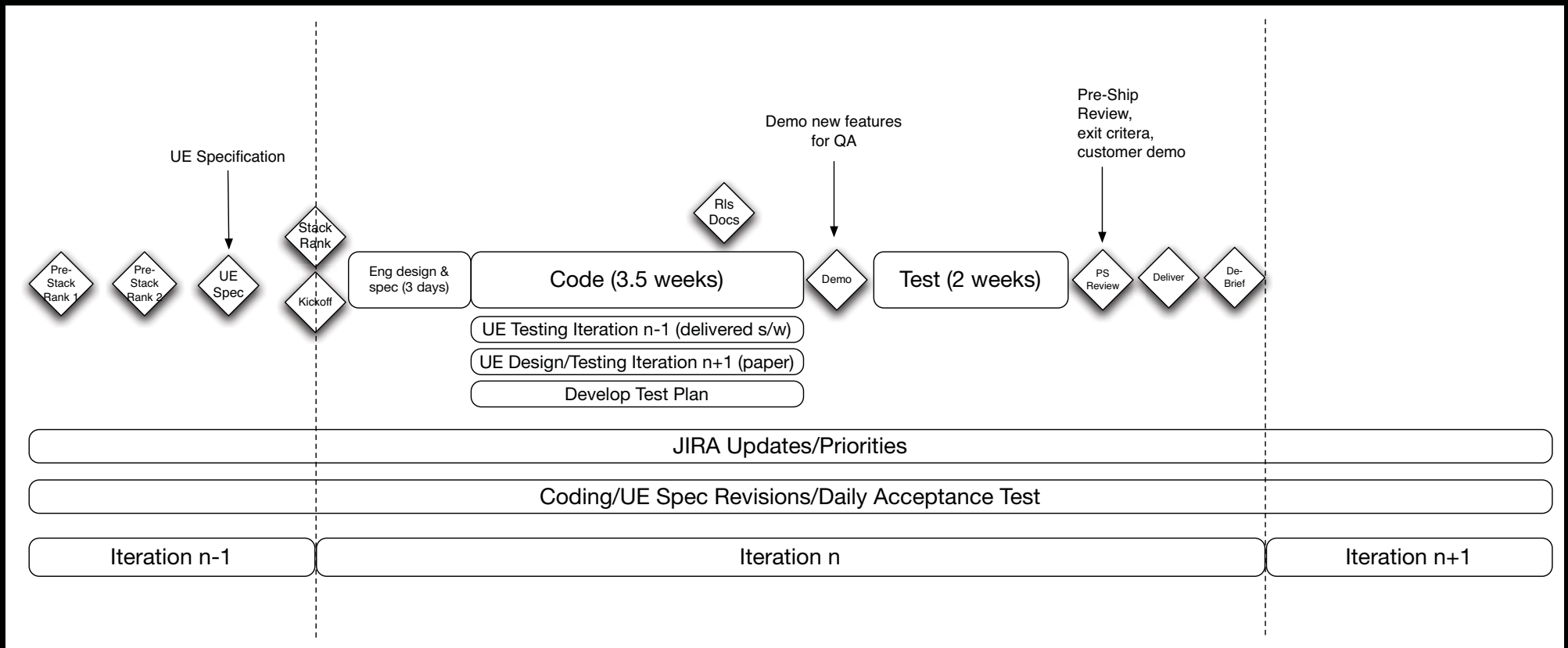


# Six Week Cycle

- We took the six month cycle and divided it into smaller pieces
- This was a start, but still left many issues



# Six Week Cycle

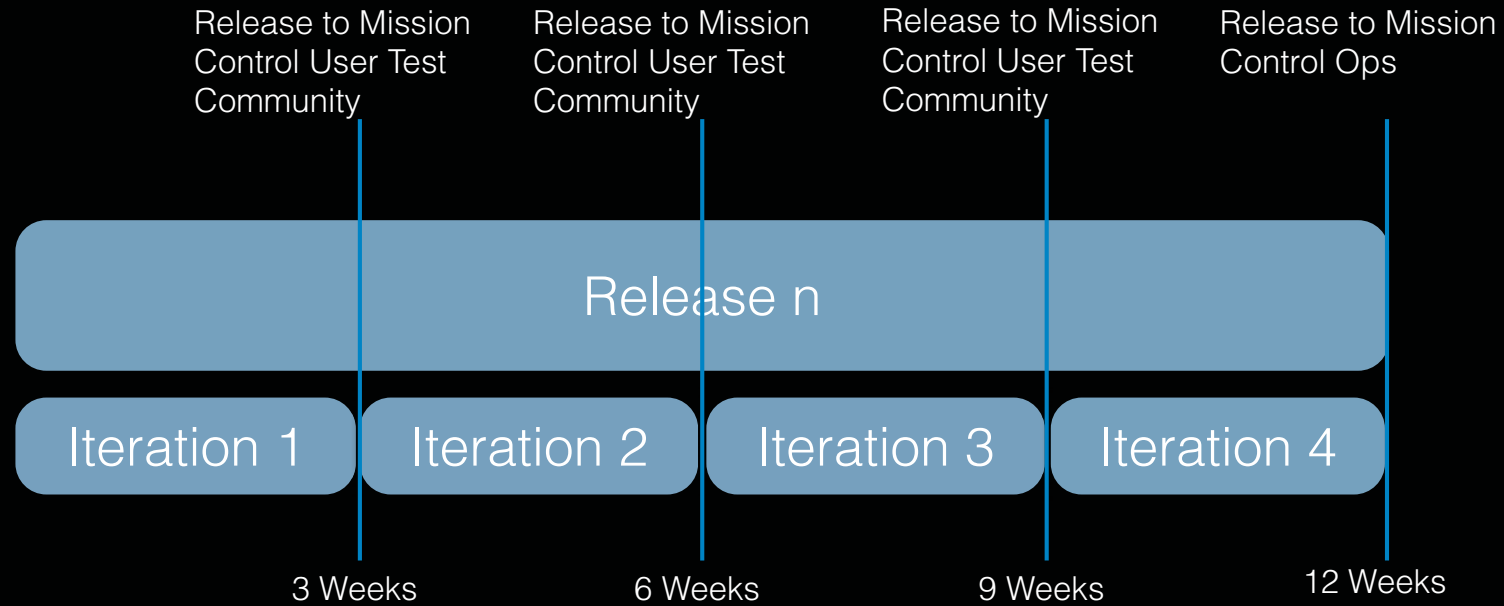


# Agile

- We shortened the cycle to three weeks
- Replaced discrete events, with integrated interactions
- Integrated strategic and tactical into our ranking process
- Each iteration had clear purpose, goals, ranked priorities
- Release, iterations, daily build
- Strategic road map

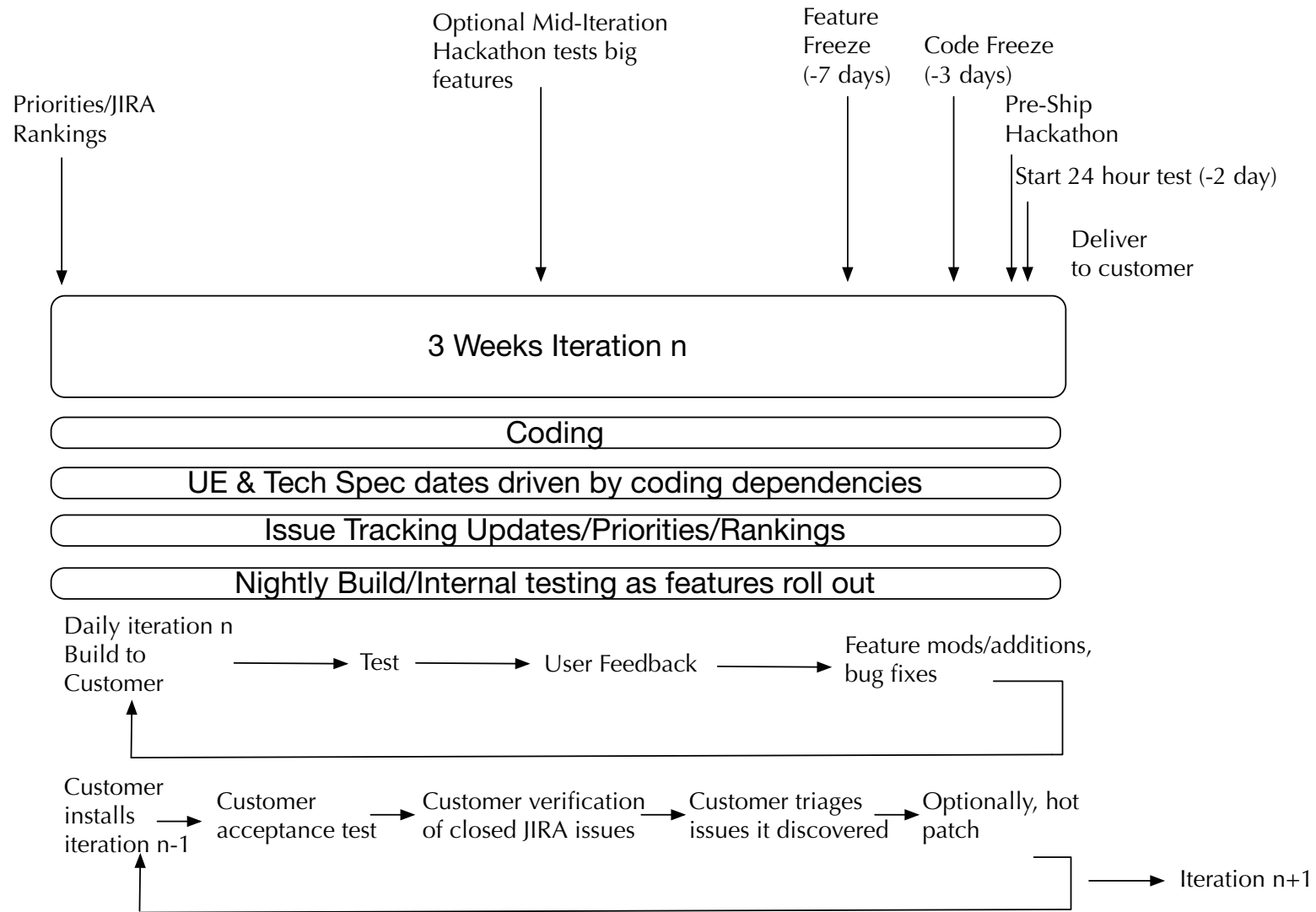
# Agile Cycle

- Deliver to customer every 3 weeks
- Nightly build
- Release every 3 months



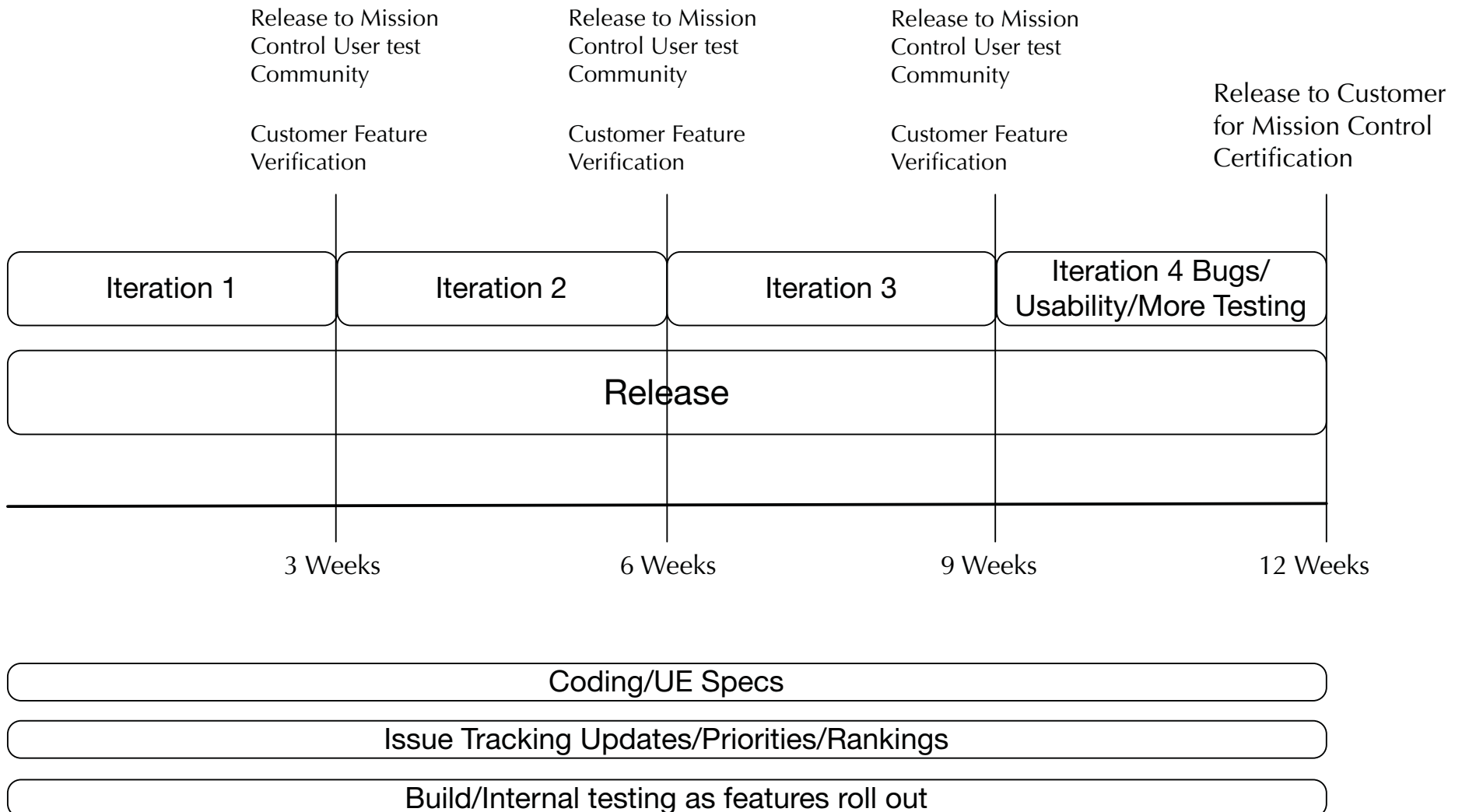
# The Three-Week Cycle

## Agile Development Iteration



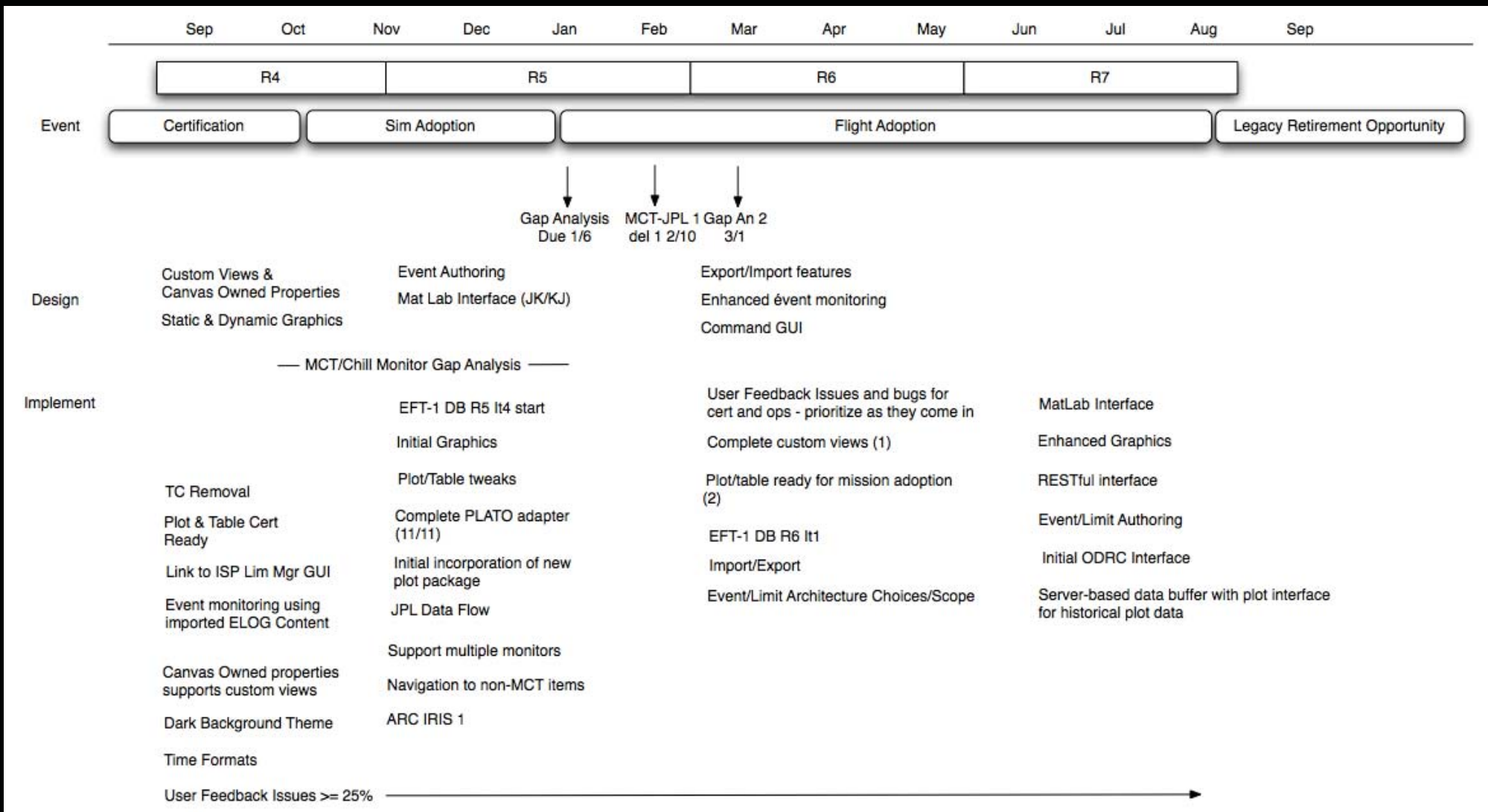
# The Release Cycle

## Agile Release Into Operations





# Strategic Road Map

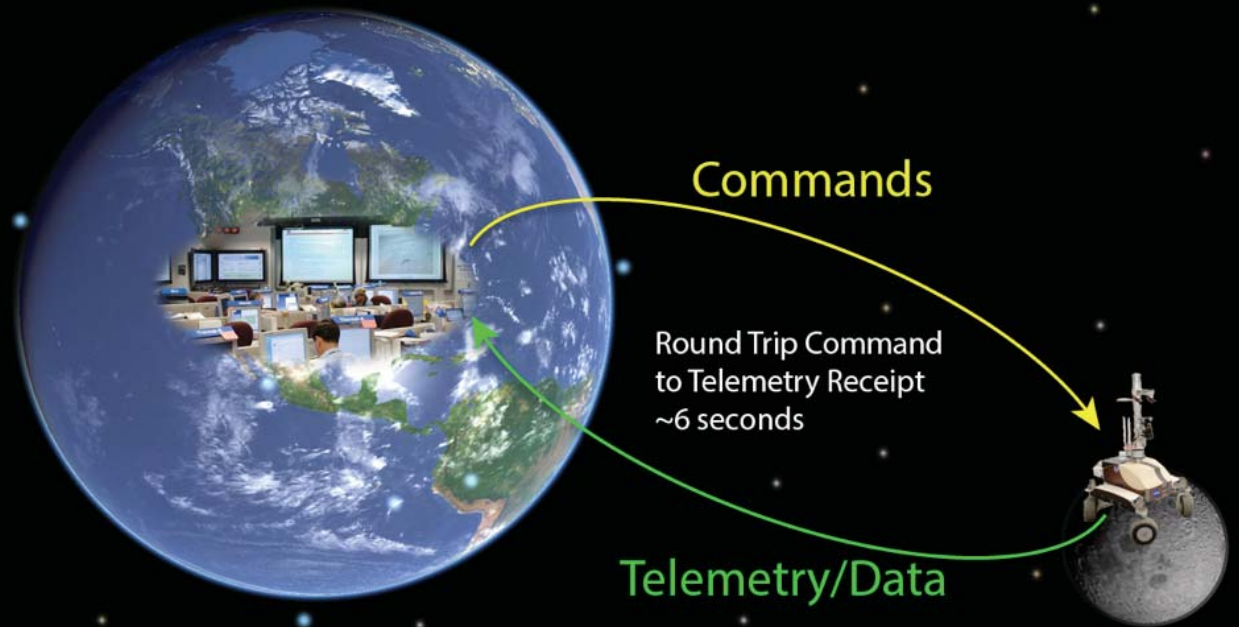


# Lessons Learned

- The measure of progress is working code
- Work on highest priorities first, avoid the temptation to do the easier things first
- Demonstrations, not presentations
- Customer interaction over extensive documentation
- Progress always visible, nightly build available
- Ship each iteration on time, only working features ship
  - Do not delay shipment for features - if a feature is not ready it goes into the next iteration
- Fit the process to your team context and culture, there is no one right way

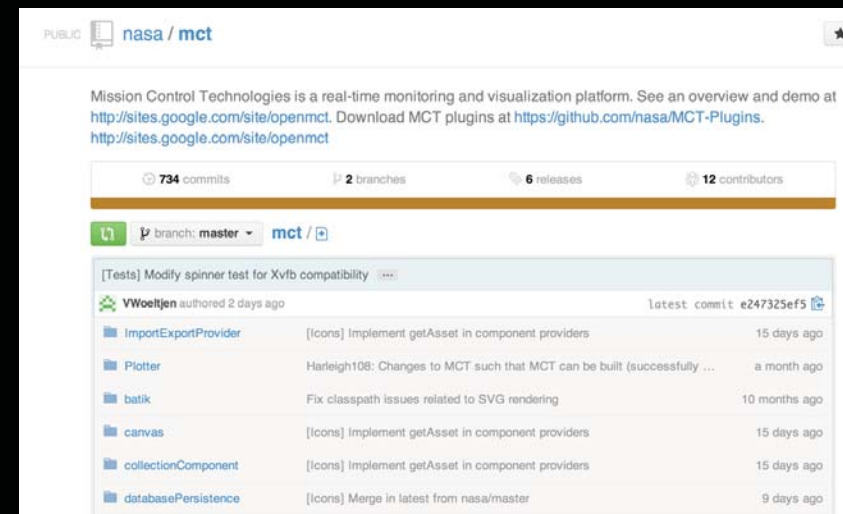
# Agile for Missions

- We are applying agile and lean methods to the design of a Lunar Rover Mission operations and ground data system



# Agile/Lean for Lunar Rover

- Principle
  - Measure of progress is working code
  - Agile - download nightly build
  - Space Mission - demonstrate operational capability through simulation



# Agile/Lean for Mission Ops

- Principle
  - Customer interaction over documentation
    - Agile - Participatory design (one method)
    - Space Mission - mission operations design session using PD methods, low fidelity simulation

TETAN: Mission Overview View

CADH	GNC	COMM
RTs		
ADM States	CMG	VHF
DSSAS	GPS	Video
Time Mgmt	RGA	Audio
SAM Conn	ATTITUDE CONTROL	S-band
RM	NAVIGATION	Ku-band
TTCG		
MDM BST		
C/W		
HPOL/STRAW		

