Mars 2020 Simple Planner

Mission Systems Engineering

James Hazelrig, James Biehl, Steve Chien, Dan Gaines,
Elyse Fosse, Guy Pyrzak, Nick Waldram

Jet Propulsion Laboratory
California Institute of Technology

11th March 2025

JPL Clearances
URS 332040
CL # 25-1108

Talks on the M2020 Simple Planner

Topic	Speaker	Date
Overview of Simple Planner	Moffi	5th December 2024
Onboard Planner: Flight Software	Gaines	4th February 2025
Onboard Planner: Trusted AI on Mars	Reich, Chien	18th February 2025
Simple Planner: Ground Tools for Operations	Connell	25th February 2025
Simple Planner: Systems Engineering Operations with Autonomy	Hazelrig	11th March 2025
Rollout of the Simple Planner	Waldram	19th March 2025

You are here

Location: All talks are in Pickering Auditorium, Building 321, JPL Campus.
Time: All talks are 12 noon - 1 PM PST
Miss it? Recordings of all talks will be archived on JPLTube
Slides will be posted at https://ai.jpl.nasa.gov/public/projects/m2020-scheduler/

Introduction

Simple Planner is flight and ground system that enables the Mars 2020 Perseverance Rover to adjust to: **unexpected state**, such as Martian temperature fluctuations or battery performance, and **activity execution feedback**, such as activities failing, ending earlier or later than expected.

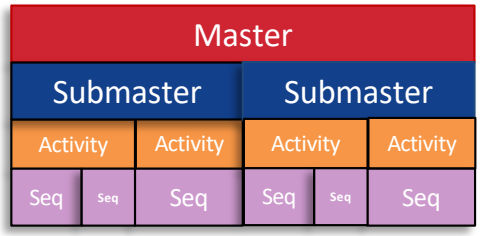
Simple Planner development began in 2016, with semi-continuous ground system design, build, and iteration through present, with milestones of operations use beginning with “technically we landed with SP” February 2021, formal “OBP in control” October 2023, and final MS desired capabilities online as of May 2024.

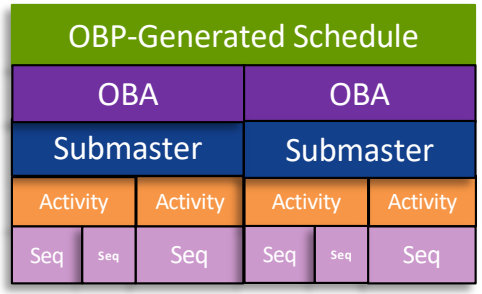
Systems Engineering Operations with Autonomy is an iterative human-centered design process that begins with an assumption about what a day in the life of the operator will look like and then goes through thousands of rounds of refinement, redirection, redesign, and resignment until an operable tool is accepted by end users.

This talk describes the end-to-end thought process and steps taken not only enable Simple Planner, but to maximize its value-add to the mission.

Simple Planner vs Traditional MSM Surface Commanding

Orchestration Function	Master/Sub Orchestration	Simple Planner Orchestration
Dispatch execution instructions	**Master Sequence**	**On Board Plan File (OBPF)**
Group sequences for dispatch	**Submaster**	**On Board Activity**
Constraints	Honored via **ground** checks	Honored **onboard** via OBPF
Cleanups	**Always** executes*	**Conditionally** executes (rare)
Execution timing & ordering Heating Rover shutdown Rover wakeup	**Fixed** planned on the ground includes margin	**Flexible** criteria provided via OBPF real-time onboard decisions
Use available onboard resources	**N/A**	**On Board Planner**





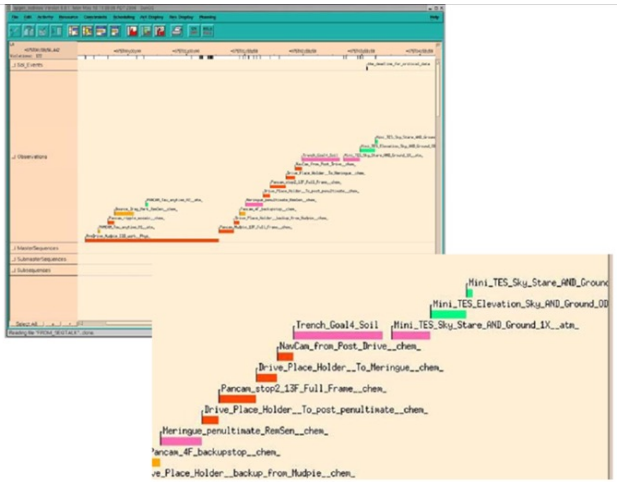
*cleanup structure updated on M20 to allow parallelism between submasters

Any views and opinions expressed herein do not necessarily state or reflect those of NASA, JPL, or the California Institute of Technology.

First, a bit of History

Previous Mars surface missions that helped get M20 Mission System centered on constraint-based planning (CBP) and autonomous scheduling:

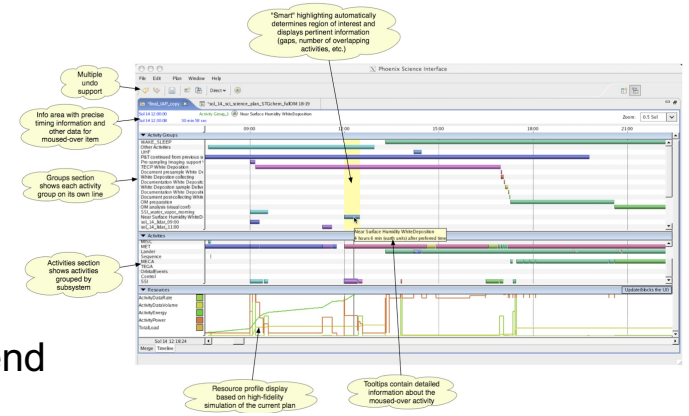
- **Mars Exploration Rovers (MER) (2004)**
  - Majority of the mission used MAPGen as its GUI for planning, which was a version of APGen using Ames' Europa2 planner under the hood that was fed by Constraint Editor
  - AEGIS autonomous targeting + CASPER scheduler onboard proposed (2007); AEGIS approved for implementation, CASPER not approved. AEGIS operational 2010+

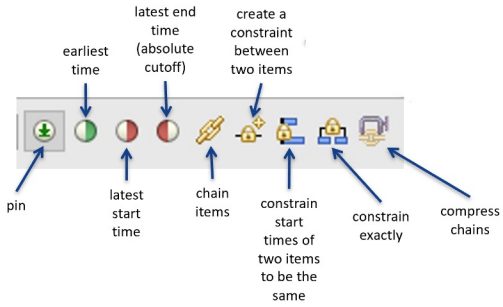


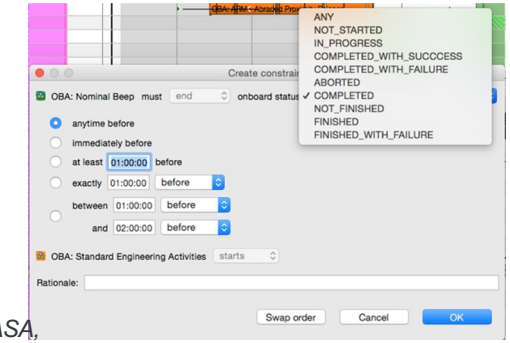
The Integrated GUI

- **Phoenix (2008)**
  - Used a Phoenix Science Interface (PSI) as its GUI, a proto-MSLICE tool and utilized the APGen procedural scheduler
  - No onboard autonomy mentioned in interviews, not end-to-end

- **Mars Science Laboratory (MSL) (2012)**
  - Uses MSL InterfaCE (MSLICE) as its GUI, the standard for complex flagship level Mars surface operations
  - MSLICE has CBP UI elements which can and are used occasionally to this day to convey intent and influence some sequencing
    - Plan Advisor built into MSLICE as a response to feedback from MER
  - Onboard scheduler evaluated in formulation, not baselined for implementation
  - AEGIS autonomous targeting post landing deployment, operational in 2016+.
  - Partially end-to-end for robotic operations







Any views and opinions expressed herein do not necessarily state or reflect those of NASA, JPL, or the California Institute of Technology.

Now a bit on design philosophy

Any views and opinions expressed herein do not necessarily state or reflect those of NASA, JPL, or the California Institute of Technology.

Design Challenges

Phase D through Pre-landing Phase E, October 2019 - February 2021

1. **Build out of the ground system always had simple planner in mind, with a philosophy of *"solve the hard problems first"*** *-Trosper*

   This implies everything *not* simple planner related is easy and can be deferred to later; spoiler not easy

2. **Enable M20 planning on a shorter timeline than previous missions had at their landing**

   Mission plan in 2018 laid out that we needed to LAND with a 5-hour timeline in order to accomplish all goals of the science campaign; don't worry, this story gets worse!

3. **By the way, ALL facets of a "regular" surface mission need to be ready by landing day!**

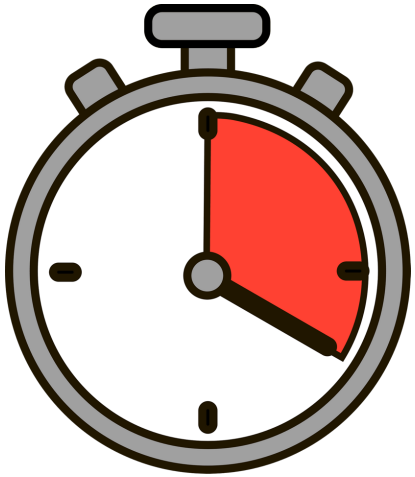
   Rule 1, don't break the spacecraft
   Rule 2, don't safe the spacecraft
   Rule 3, do all the things, preferably fast because even the 5-hour timeline has negative margin
   You get ONE surface nominal ORT in 3 months out from landing

"You can't make it better until you make it work."
-Akin's Law #40

Challenge #1, Simple from the Start

- Copilot was the onground planner counterpart to onboard planner and was used throughout thread tests, ORTs, and in use on sol 0
- Copilot would also house OBP planfile binary generation for flight
- Constraint-based planning was built to support specifically having an onboard planner in flight, not just the ground
- A high fidelity FSW port (SSIM) was needed for validating command loads
  - SSIM vs SeqGen trade included OBP implementation
  - Strategic modeling capability was built out but later descoped in Phase E
- Having enough of an idea about what Simple Planner operations will look like to:
  - set expectations on the wider science and engineering operations teams
  - Training often had a duality, “you’re going to do it this way for now, but in the future this *other way* will be how you do it!”
  - govern how they build their activity modeling and sequence expansions
- Because the ground model is now directly feeding OBP in flight, heavy requirements on the underlying Activity Dictionary to support all needed features, i.e. parallelism and heater enforcement
- Cleanup strategy in MSM mode specifically crafted to behave more like OBP would, submasters able to run in parallel

Challenge #2, Timeline



- MSL & MER at landing used two operations teams per planning cycle across a 16hour (2x8) timeline
- The goal for M2020 was motivated to land with a single shift:
  - Human factors lessons from prior experiences
  - Staffing and training did not support a 16-hour timeline
- An expectation might be to land with how MSL surface mission was operating as of 2020, single team on a 7hr timeline, but this was impossible
  - MSL surface tools and processes were largely discarded until late phase D; M20 MS was largely starting from scratch
    - i.e. 4 months before landing we could not bundle command loads in an scmf in under 2 hours, per iteration
- The work required for basic functionality in MSM mode had not been fully scoped for M20 GDS until about 6 months pre-landing, at which point more resources had to be mustered to get across the finish line

Challenge #3, Mars Rover Ops 101

- Subsystem assessment capability in downlink ✅
- Downlink data processing pipeline function ✅
- ROUNDS Rules ✅
- Activity Dictionary ✅
- Power modeling without validated Copilot model, means MMPAT ✅
- Data modeling ✅
- OIA's signed, everyone knows what they have to deliver and when ✅
- **NEW** architecture that supports FSW dumps to bootstrap SSIM ✅ ← *and it didn't blow up!*
- Robotic navigation and arm planning software ✅
- Infrastructure, procedures, and processes stable enough to support tactical and CI ✅
- **NEW** Simulation of integrated command load on a tactical timeline ✅ ← *and it didn't blow up!*
- Flight Rule checking, about 60 automated rules (out of ~800), heavily manual, every team for themselves, and some hastily written rules, but otherwise done! ✅
- Flight rule disposition reporting, Waiver reporting and processing ✅ ← *check out RuleCheck!*
- Last mile software such as COCPIT helper scripts and uplink review reports generation ✅

Key Take Aways

1. **There is no backup for the systems engineering function**
   - It is shocking how few of people can have such a huge impact on how we explore another planet. No pressure… actually pretty extreme pressure

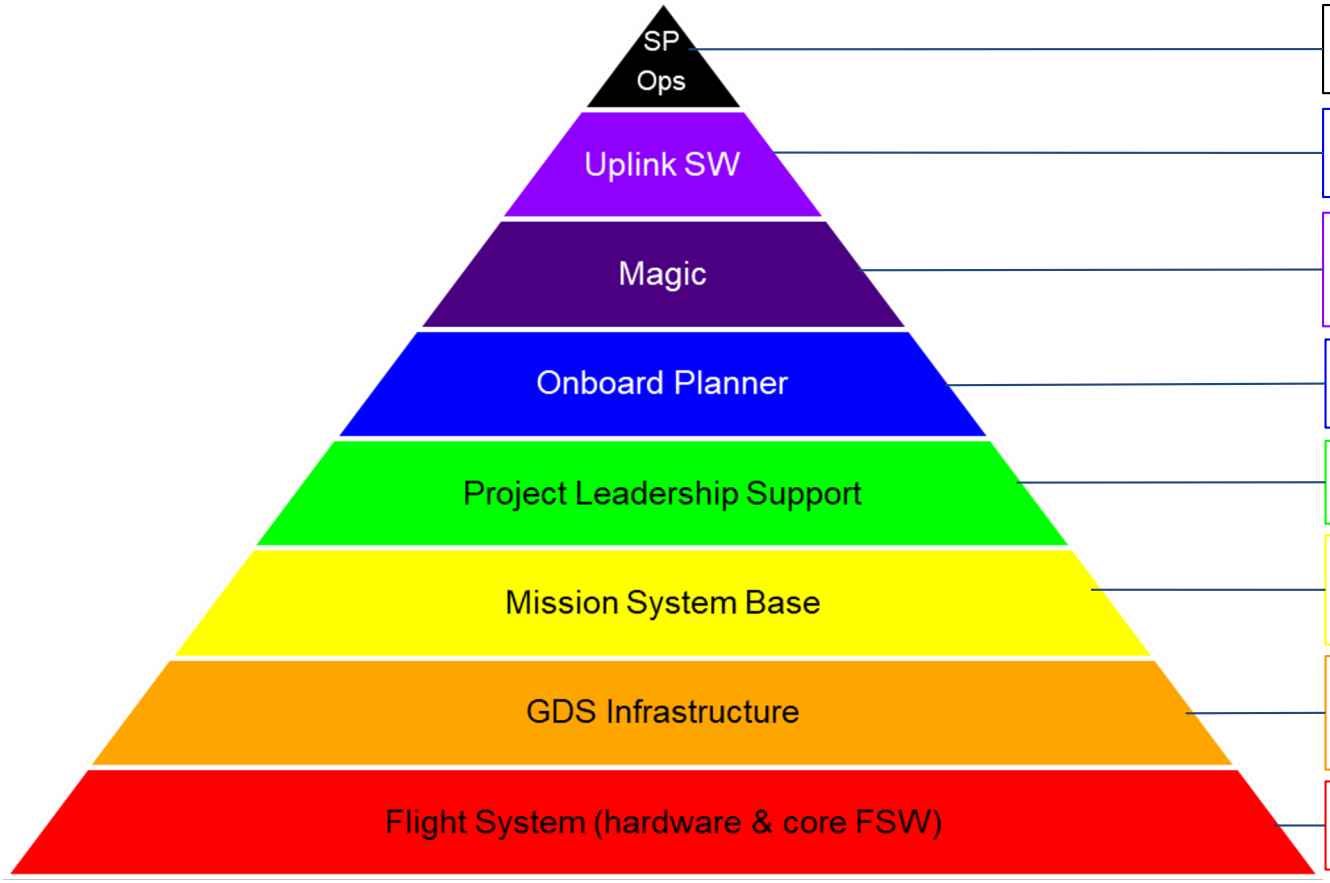
2. **Playing the long game:**
   - Building a new system around simple planner meant that adoption or refinement of current practices taken from an MSL 2.0 approach was descoped or deferred, usually for years

3. **Stay out of corners and boxes:**
   - I can count on one hand number of divergent designs that applied only to MSM mode in order to keep ops engine running smoothly

"If there is not a flight system problem, there's no money.
If there is a flight system problem, there's no time."
-Akin's Law #42

How to enable all of this to work

What it takes



Critical mass of top tier operators with a can-do attitude, and LOT of Perseverance

Great UI/UX, focus on operability and functionality, E2E system with FSW-in-the-loop

Thousands of hours of collaboration between talented gds and fsw devs and MOSE with deep understanding and the power to set terms

A robust OBP FSW design with an understanding of operability

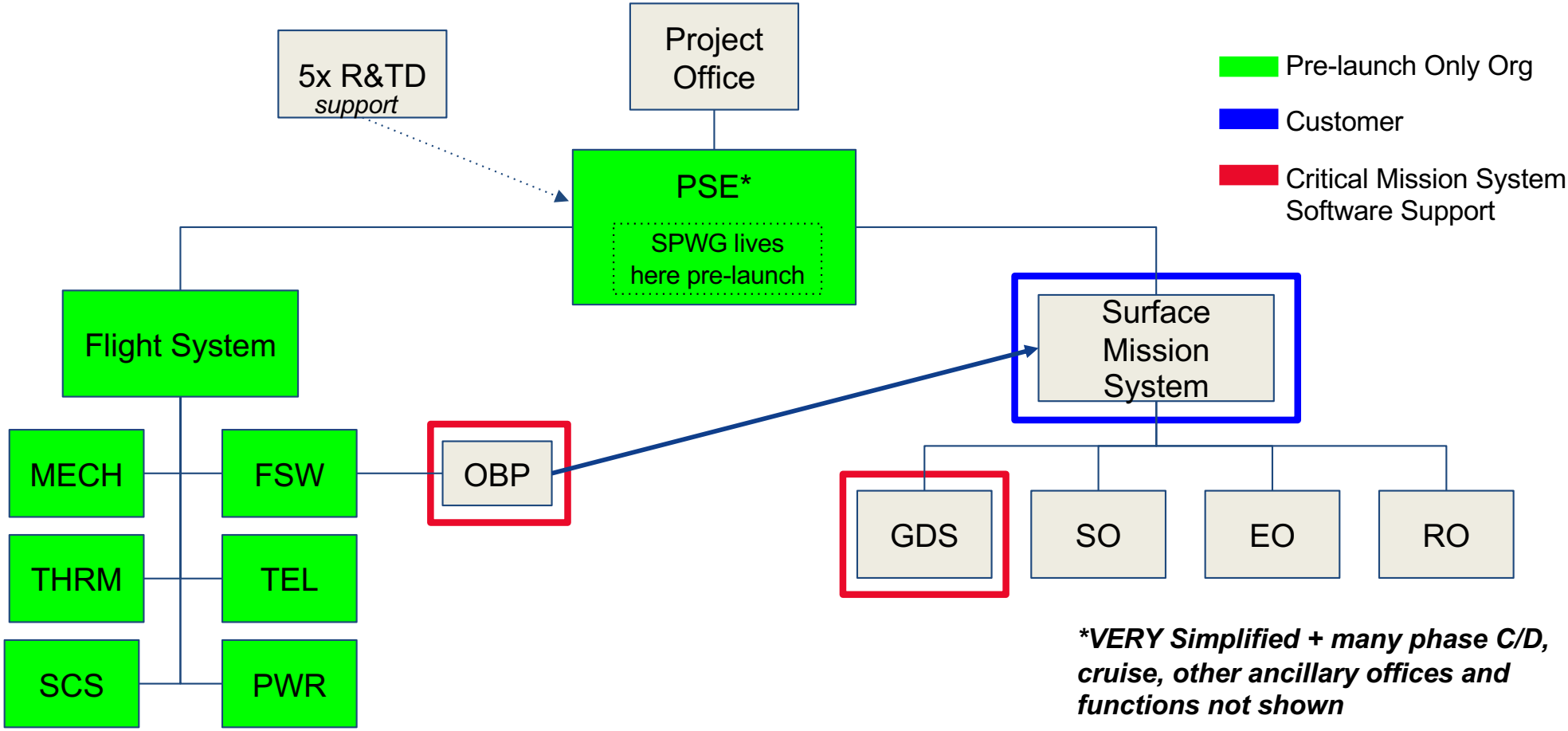
Consistent support and trust in the vision from development through operations

A worlds class ground system and process that has to be faster than previous missions and work with and without Simple Planner

A robust GDS Infrastructure that is simple for the end-user to access but has backend support for advanced development

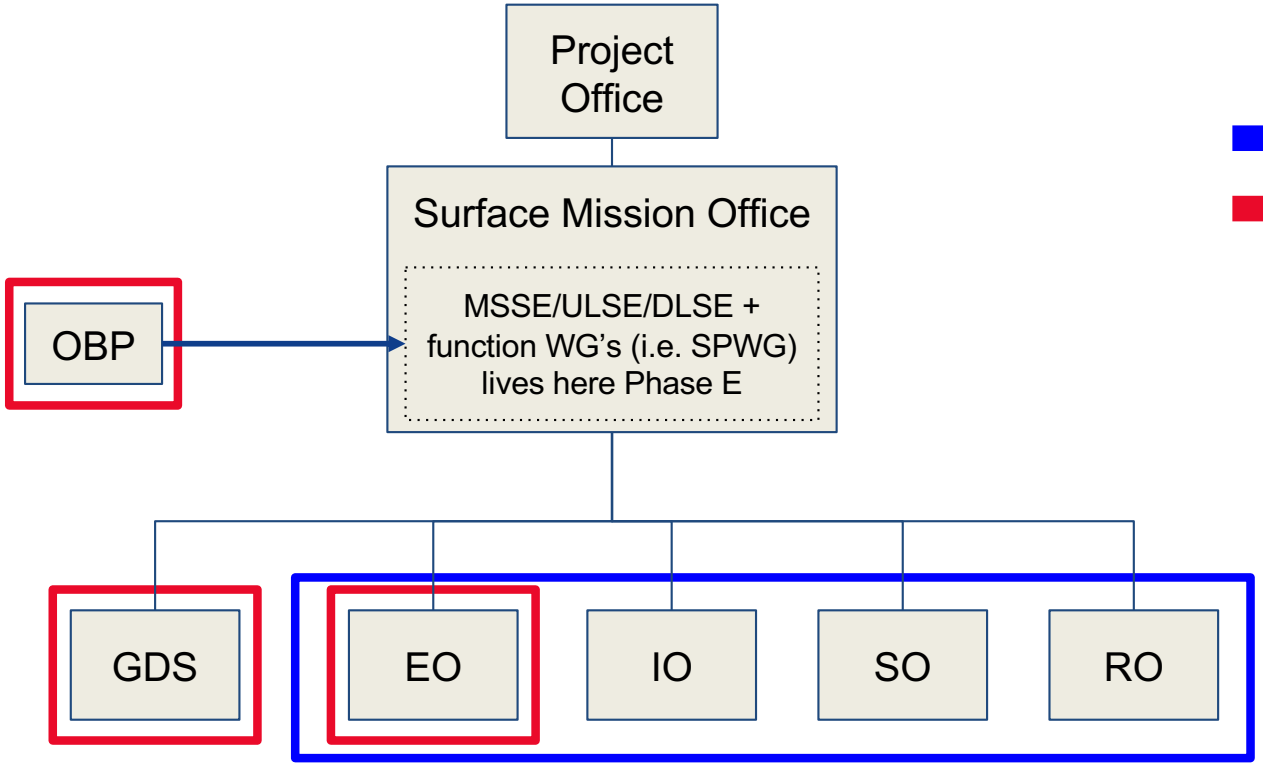
A robust flight system with stable, bug free core FSW and no hardware failures

Org Chart* & Interfacing Pre-Launch



VERY Simplified + many phase C/D, cruise, other ancillary offices and functions not shown

Org Chart* & Interfacing Launch through Surface



VERY Simplified + many phase E, cruise, other ancillary offices and functions not shown

General Uplink System

"Design is an iterative process. The necessary number of iterations is one more than the number you have currently done. This is true at any point in time."

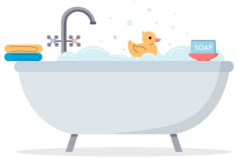
-Akin's Law #3

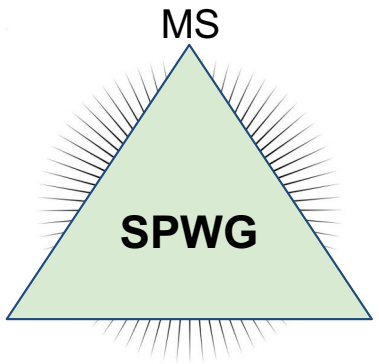
Standing meetings, generally since 2019 - Present general uplink ~10 hours per week not including SPWG, most now bi-weekly since 2024

- 2 — Stakeholder meetings with Devs, SE, & ops customers review & update **priorities**, initial triage of new issues, status updates
- 4 — Software specific **design & requirements** gathering working groups
- 2 — Reserved blocks for pre-code freeze **user testing** with developers present
- 2 — Formal GDS, ops customers, & management **acceptance**, testing & release schedule **coordination**, other UL stakeholder meetings

Simple Planner Working Group

Oct 2017 - Aug 2018
Nov 2019 - May 2020
April 2022 - Present





Some Highlights of FS-MS Collaboration and Feedback Loop 🔁

- Defining FGICD (Vol 3) for OBP: forced understanding of what to send to vehicle and what MS should control
  This led directly to requirements for SSIM, Copilot, COCPIT
- Talking through conops meant GDS could scope and develop user workflows that captured intent with confidence and translate it to something the rover understands to produce the same intended behavior in flight
- Figuring out how to switch between MSM-OBP modes by Sol 0 forced CBP to be applicable to both
- Figuring out how to map MS AD claims + compatibility requirements into limited resource bits for parallelism
- MS system could set priorities for FSW features to get VnV'ed - wishlist items or bust!
- MS could provide the missing piece of the puzzle on how to get buy-in from users vs what's a bridge too far, etc.
- Bug fixes and changes requests were more readily accepted because OBP is a component in FSW
  Where FSW wasn't sure of impact on user experience, SPWG was the venue to get that answer
- Having the brain trust to work through each challenge case one-by-one:
  i.e. "you'll never do Sampling with OBP, too complex and risky" - in fact the energy OBP recoups on drill sols means the ability to reduce turnaround time both for nominal and especially off-nominal attempts; first sample acquisition in simple planner happened only one week after we were using it ops!

How to enable both modes to work

Many Inter-Org Interfaces, Minimal Changes

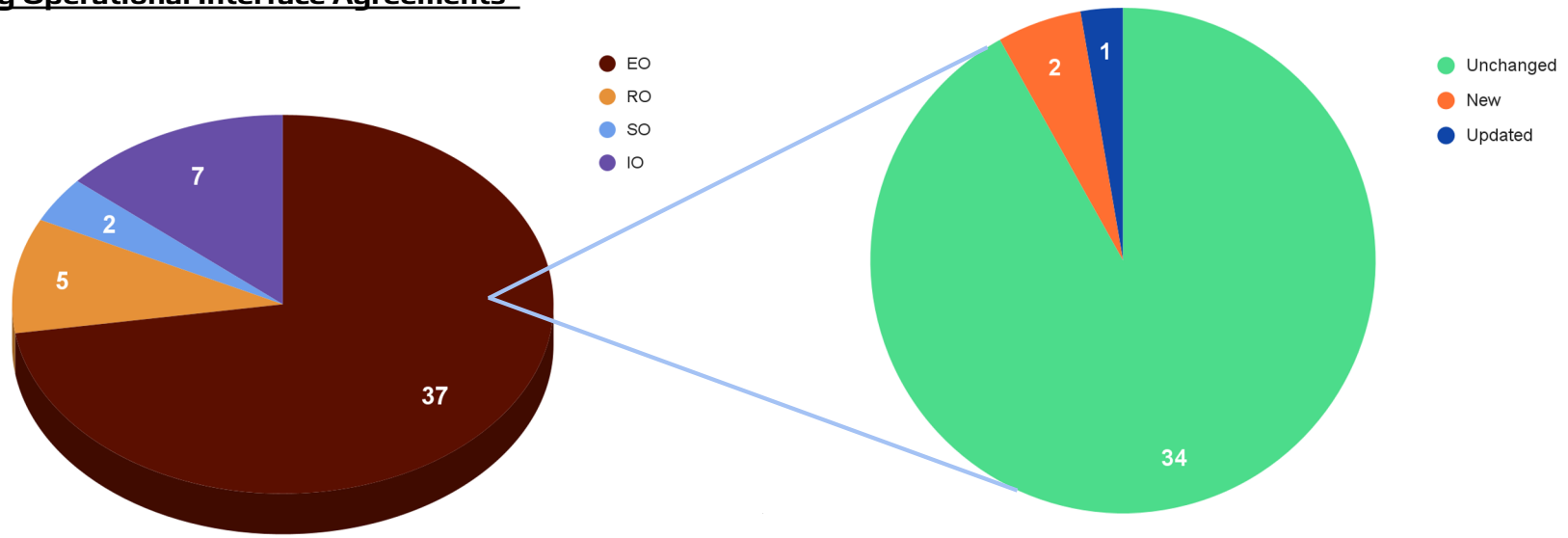
"The ability to improve a design occurs primarily at the interfaces. This is also the prime location for screwing it up."

-Akin's Law #15

Scope of Simple Planner changes Post-Landing:

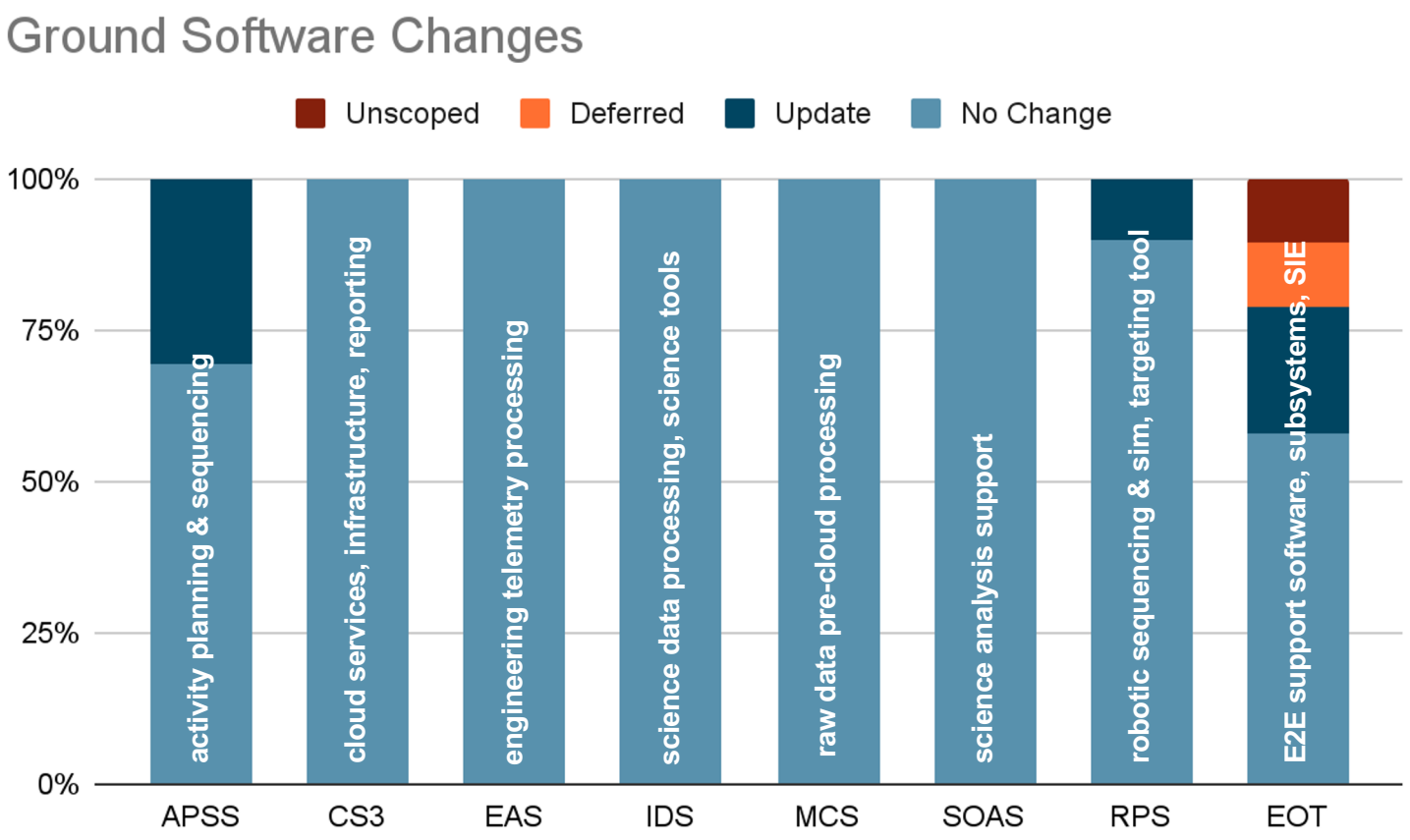
- 1 update needed once final GDS design closed
- 2 new OIA's created for adoption of proposed OBP Thermal Ops

Planning Operational Interface Agreements*



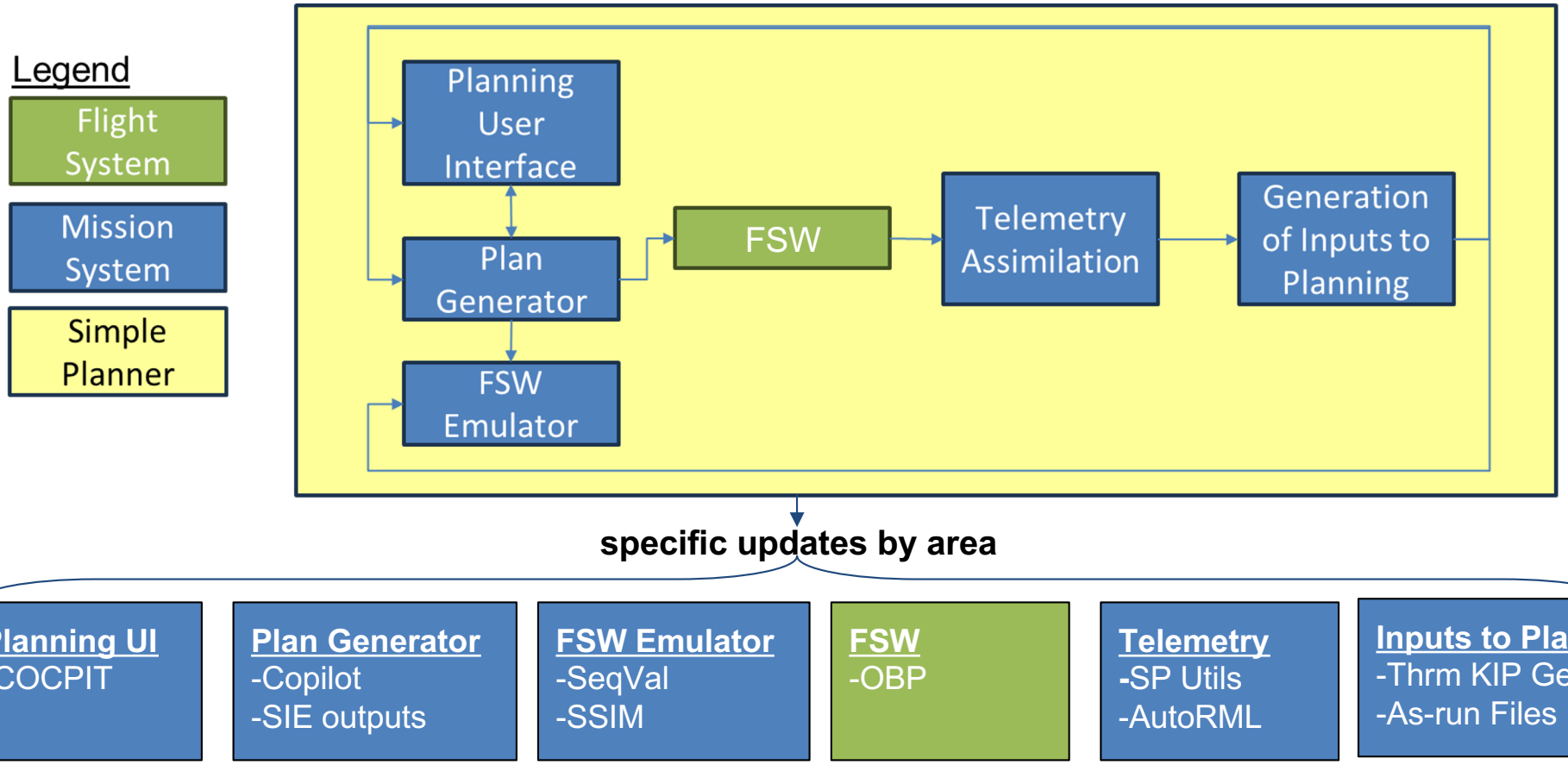
**Software Interface Specifications inclusive with OIAs, not enumerated separately*

Software Deltas by GDS Subsystem



Small amount of new scope, but the design requirements closure was focused almost exclusively on activity planning, integrated sequencing, and generation of key inputs to planning

End-to-End Software Flow Changes



Constraint Based Planning (CBP)

To fully leverage the gains of an autonomous scheduler, encoding user intent is paramount; the UI has to work with how operators already think.

- Luckily, users already know what they want, the trick is to get them to be upfront about it.
- **Importantly, the planning UI in COCPIT supported CBP for *both* MSM and SP**
  - This meant users could practice
  - We got several iterations of how to improve the UI and hone-in the required capability

Operability is equal weight to functionality

Anything that could be automated “behind the scenes” was best, anything that asked the user to modify behavior took a commensurate amount of time to train. Simple concepts took a seminar, complex ones took that plus additional tool automation to ease workload, but to fully trust autonomous scheduling in a constraint-based planning paradigm took giving time for operations leads to see how it worked for themselves, and there is a saturation rate that I believe we maintained to continue progress without steamrolling ops.

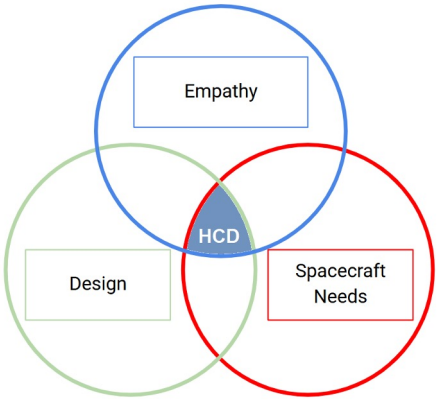
Some lessons learned in early SP ops

- The avoid the fear of breaking promises to science and instrument teams and thus losing trust in the autonomy, time window rollup was originally intersection only. However, that felt draconian and teams were all willing to accept a “close enough” but easily operable network rollup that should still work 99% of the time.

- Handling brittleness - thermal modeling trade between excessive waste in heat vs torquing start times so far to the right an OBA no longer fits in its allowable execution window, aka nuisance dropped activity in flight. Manual workaround is not only cumbersome and tedious, trying to train the FSW behavior across the broader ops team for even application is a nightmare. Better to automate concerns away from the user so they don’t go down the FSW behavior rabbit hole.

Human Centered Design

In order to adopt more autonomy, or advanced AI in the future, understanding that there is a human, or usually many humans, in the loop, means you cannot simply make a blanket statement of "we will automate that person out of the job!" instead understand the limits of what you are trying to accomplish, get as close as you can, and then allow for a little bit of grace, that's where humans fit in.



Quotes from SPWG when asked about replacing people with autonomy:

- "Have to have someone to put intent into the system."
- "The goal is not replace humans, but to give better tools; increased productivity turns into better science/more science"
- "What would *I* want as an operator?"

Some Design Considerations for Simpler Planner:

- What to show users, what not to; where to hide the get-out-of-jail knobs
- COCPIT has to be both more powerful than MSLICE for planning and sequence setup *AND* easier to use!
- Don't lie to the ground system or users (generally) - show the system bias so they can work with it rather than against it
- Need to build trust in the ground schedule and flight execution behavior that isn't so counterintuitive to experienced operators that people revert back to MSM
  - Example, Simple Planner is not an "optimal" planner, so where a human might expect the rover to make a smart choice about minimizing heating or awake time, they might be surprised by how "dumb" the system actually is. So instead, train the users to accept that 9 out 10 times their activities will start as early as they have allowed, even if that is suboptimal for energy. Better still, create an automated way for them optimize energy on the ground within their defined windows in a way the rover *will* understand!

Hindsight is 2020

Any views and opinions expressed herein do not necessarily state or reflect those of NASA, JPL, or the California Institute of Technology.

Lie, Cheat, and Steal

Lie

Despite best efforts, there is still oodles of idle time margin in the ground power model that paint a conservative picture, to make use of that margin, activities which can be marked optional aren't counted against the ground power model, knowing flight will do the right thing. This is a core feature of simple planner, but lo and behold, those optionals really have a tendency to schedule even when the ground model would otherwise say there isn't energy for them!

Cheat

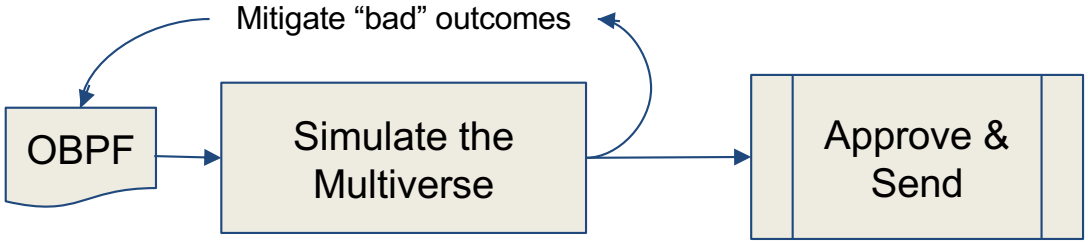
We can get away with a lot of things because the number of possible permutations in the way the rover operates vastly simplifies when you make a statement such as "most of our cameras are used during daylight hours" or "three instruments share the remote sensing mast (RSM), all of which use RSM heating", or "three instruments share the turret, the placement of which all use arm heating and RSM cannot be used during that placement."
Another example, the heater story for mobility simplifies greatly when you promise not to blind drive at midnight
Operations constraints like this are absolute cheat codes to solving potentially gnarly problems.

Steal

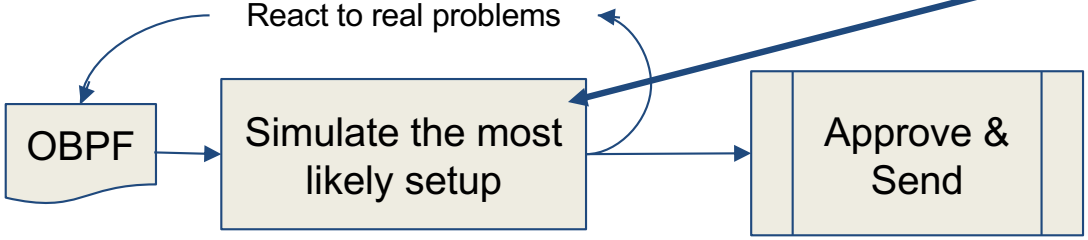
Any time you can keep key players on the project, steal their time because they will invariably help you close out issues a lot faster than trying to grow a clone from scratch. We were very fortunate on M2020 to continue to find ways to keep the smart people around for as long as possible. i.e. there is no more OBP FSW dev nor VnV, but yet Dan Gaines, Kevin Reich, Nick Waldram, etc are still on the project,

Validating non-deterministic system

Original Thinking (Strategic Validation)



Refined Thinking (Tactical Validation)



The amount of science planning we have time to plan and sequence day to day means we don’t allow for branching sol outcomes, i.e. we won’t come back the day after a drive sol and find out we actually drilled instead!

The rover use cases vastly simplify the problem:

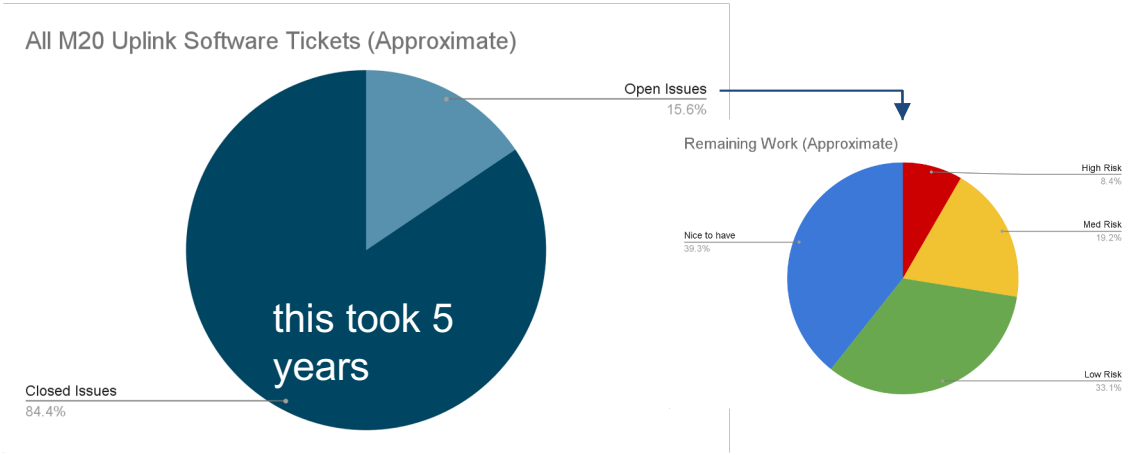
- complexity limits
- parallelism constraints
- dependencies
- thermal constraints
- science time windows

Generally this is enough to determine in timeline layout how the rover day on Mars is going to go, even accounting for switched order.

Additionally there are layers of protection on the ground and in flight:

- high fidelity simulation
- sequencing strategy
- fsw arbitration
- fault responses

Setting Priorities



Ground fixes and features were governed generally in order* of:

1. **make it work**
2. **make it fast**
3. **make it robust**

* for backlog fishing order, all three of these interleave with one another when it comes to implementation. any capability that requires phasing gets split into multiple tickets/issues

Priorities for OBP capabilities selected by “usefulness” (see right)

Reprioritizing Flight Features, in MS!
where usefulness is a combination of energy and time savings on the ground and/or on the rover as well as total science return over time

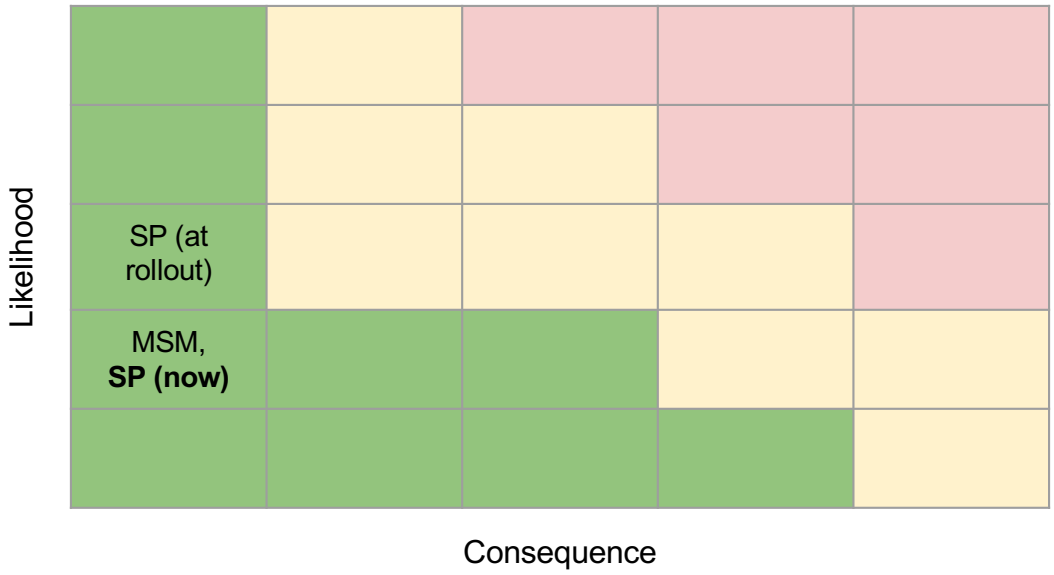
1. autonomous wake/shutdown
   a. w/shunt mitigation
2. autonomous thermal
3. snap/throttle/push/veto
4. parallelism
5. activity hopper
6. allow activity order switching
7. optional activities

8. preferred time
9. commit groups
10. delta dv acquired
11. switch groups
12. expanding oba (general)
13. pause resume
14. shutdown sequence
15. manual wakeup/shutdown
16. ddi checking for state requirements

Risk Compared w/MSM Baseline

Jet Propulsion Laboratory
California Institute of Technology

- When sp rolled out, past mvp for operations, but highest level of open risk
- msm had decades of heritage and 2.5 years of operation on m2020 already
- but now… SP in the same box as MSM
- Since its formal rollout in October 2023, **we have yet to go back to a master-in-control regime for nominal planning***



*off-nominal planning in safe mode and FSW updates require MSM control

MARS 2020 PROJECT

Pay no attention to the man behind the curtain

When things are going well, Simple Planner seems like a magic black box that allows for a much higher rate of mission productivity, however because people, time, and money were, and still are, limited, invariably latent risks will be realized and the new flavors of failure modes can shake the confidence of experienced operators because they are not as immediately known as simple command errors nor as easily mitigated against to return to “normal.” That sense of helplessness is only tolerated by the weight of benefit.

Put another way, you need a lot of goodwill and “wins” to overcome the FUD of an autonomous system when it does not work right.

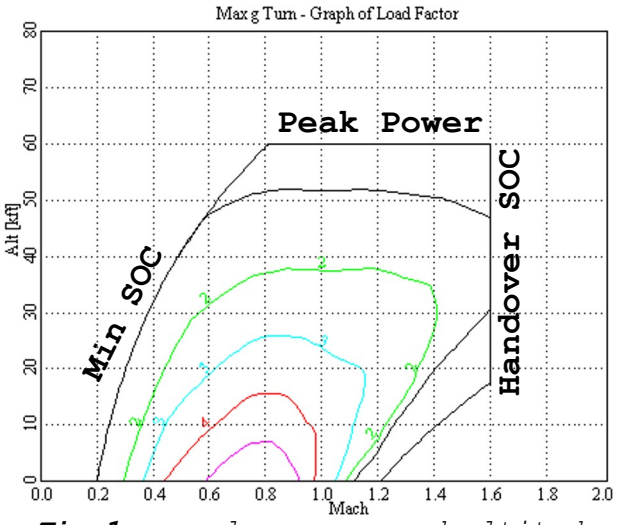
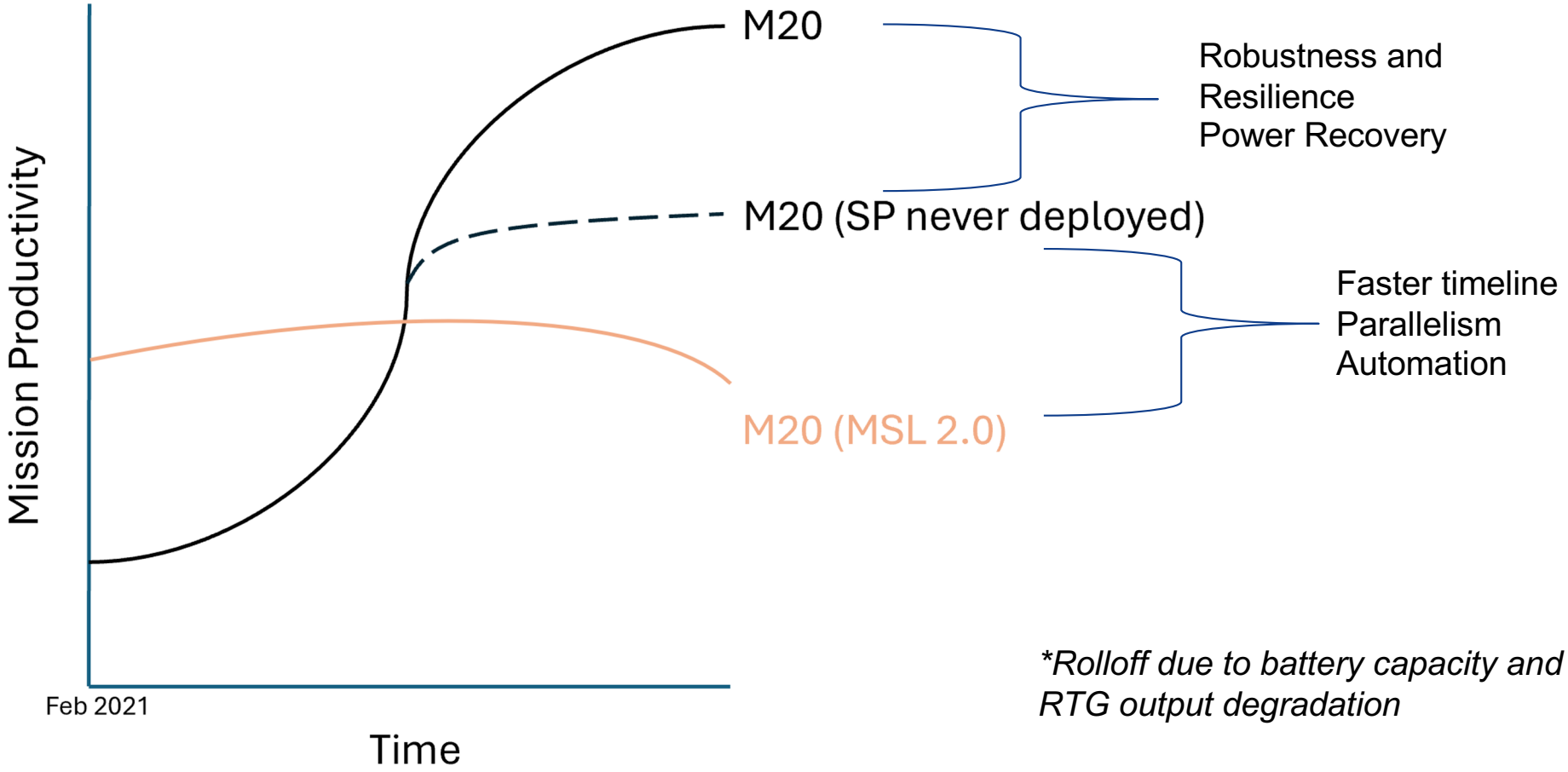


Fig 1: *poorly repurposed altitude envelope graph*

Hardest challenges to overcome

- **Thermal modeling:**
  Existing FSW and hardware limitations required a host of OBP implementation workarounds which lowered margins, sometimes inducing negative margins, none of which is visible to the user so it seems like the rover doesn’t obey the laws of physics from time to time!
- **Power modeling:**
  Still haven’t reconciled the three different power models being used (yes 3!). Existing hardware limitations puts heavy constraints on raw processing power that we take for granted on the ground, which removes margin from the system that operators otherwise expect

The Trade, Visualized (Qualitative Only!)



**Rolloff due to battery capacity and RTG output degradation*

Closing Remarks

1. If you want autonomy onboard your spacecraft,
   - MS cannot come at the “traditional” project dev timeline
   - You need a high fidelity FSW Emulator, like SSIM
2. Growing a good MOSE is an extraordinarily lengthy process, requiring multiple mission iterations, the correct discernment of what worked and what didn’t, the continuity of those lessons learned across multiple generations, and a dose of radical ideas from time-to-time to force growth without grievously wounding the heart of operations (the people)
   - At any point this chain of events can go wrong and that is why M20 ops is special
3. Since its formal rollout in October 2023, we have yet to go back to a master-in-control regime for nominal planning
4. All of the ways in which we normally operate the rover have been exercised numerous times
   - Including, and especially, collecting and caching core samples
5. The timeline has continued to shrink an additional hour since rollout, averaging <6 hours for single sol plans
6. We have made it through a Martian Winter with in-situ thermal management
   - In fact it has prevented several nuisance faults due to flexibility in scheduling
7. Simple Planner has enabled activity plans that would outright break power constraints under any other paradigm, and still with margin to be recouped by optional activities or otherwise
   - In short, M20 is now capable of accomplishing whatever science campaign lays ahead
8. The concept of sophisticated onboard autonomy and constraint-based planning, with a complex fast-paced multi-faceted mission has now been proven definitively… on a RAD750, ready for the next challenge :-)

Acknowledgements

MOS Dev

- James Biehl
- Nick Peper
- Matt Keuneke
- Sean McGill
- Nick Waldram
- Bekah Siegfriedt

Salesman of the Year

- Raymond Francis

OBP Leadership

- Dan Gaines
- Stephen Kuhn
- Elyse Moffi

GDS Leadership

- Usha Guduri
- Guy Pyrzak

APSS

- Andrea Connell
- Matt Hurst
- Evan Davis
- Sean Wenzel
- Roxana Gonzalez Burgos
- Ivy Deliz
- Wayne Chi
- Jagriti Agarwal
- Amruta Yelamanchelli
- Jason Swope
- Sara Schnadt
- Natalie Rezek
- Hoan Luu
- Dylan Stewart
- Dragos Guta
- Bob Kanefsky
- Jimin Zheng
- Anthony Robertson

EO Software Dev

- Nick Rossocheva
- Matt Hurst
- Blake Charlton
- Nick Peper
- Kevin Reich
- Laura Mann
- Flora Ridenhour
- Catherine Christiansen
- Shannon Towey

M2020 SO Pilot/SEL Team

M2020 Payload Uplink Leads

M2020 Project & Mission Management

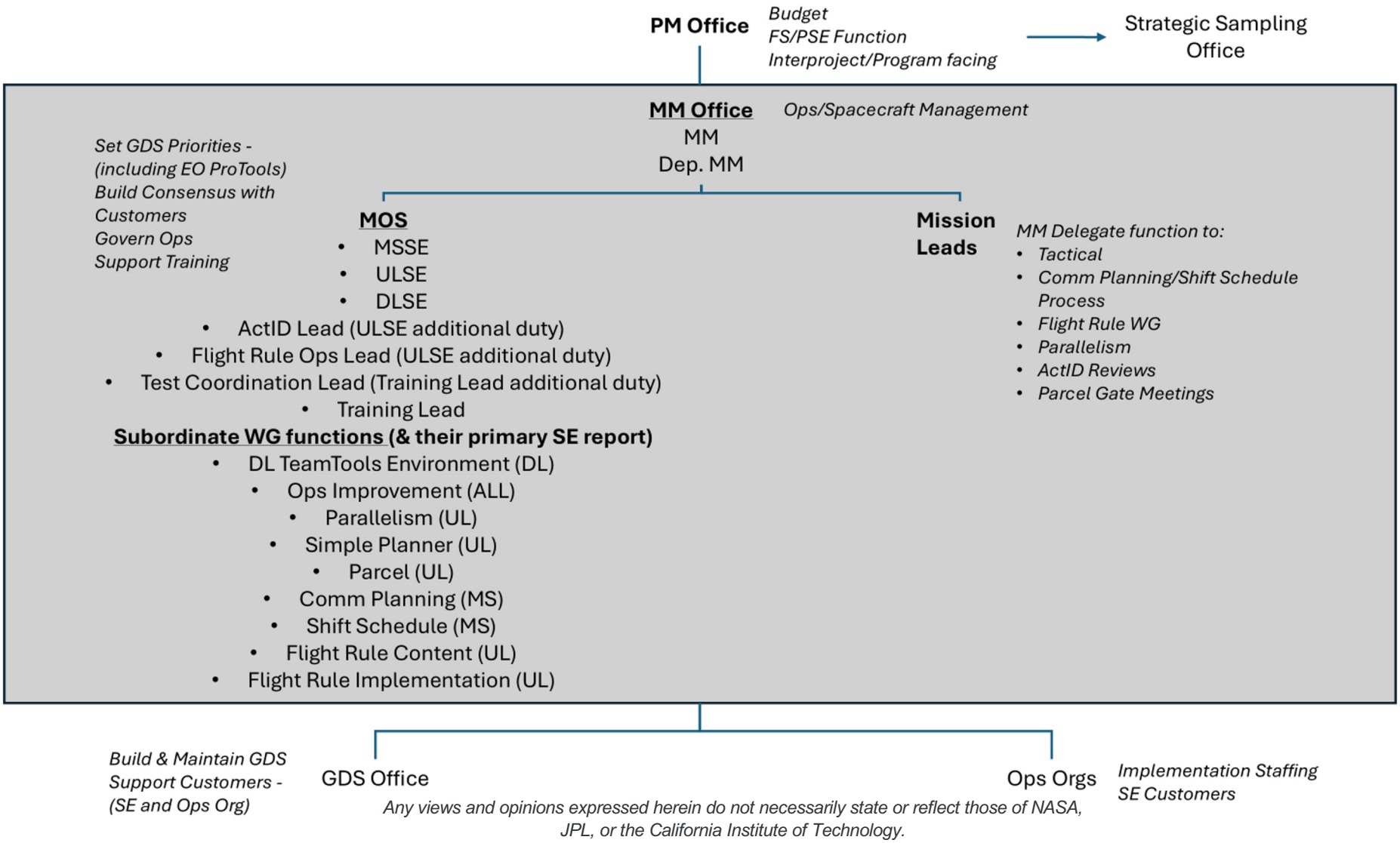
M2020 GDS Architecture Wizards

M2020 Flight System Wizards

Jet Propulsion Laboratory
California Institute of Technology

jpl.nasa.gov

Any views and opinions expressed herein do not necessarily state or reflect those of NASA, JPL, or the California Institute of Technology.



Any views and opinions expressed herein do not necessarily state or reflect those of NASA, JPL, or the California Institute of Technology.

Planning Operational Interface Agreements*

Scope of Simple Planner was baked in Pre-landing

- 1 update needed once final GDS design closed
- 2 new OIA's created for adoption of proposed OBP Thermal Ops

OIA unchanged

OIA had to be updated

OIA had to be created

EO (36)

- –OIA-Surface-EO-Comm-006: DSN Keywords File
- –OIA-Surface-EO-Comm-007: Master Comm Schedule
- –OIA-Surface-EO-DMX-002: Daily DM Sequences for Surface
- –OIA-Surface-EO-DMX-003: Onboard data product status
- –OIA-Surface-EO-MECH-001: RKSML Snapshot File
- –OIA-Surface-EO-MECH-002: NPM Parameters and State
- –OIA-Surface-EO-MECH-003: Playback RKSML File
- –OIA-Surface-EO-MECH-004: NPM Values JSON File
- –OIA-Surface-EO-PWR-001: Power Model for Surface
- –OIA-Surface-EO-PWR-002: MMPAT Modeling Files
- –OIA-Surface-EO-PWR-003: Power Initial Conditions
- –OIA-Surface-EO-SIE-001: Surface Seqs, Files, & Review Products
- –OIA-Surface-EO-SIE-002: Surface SCMF
- –OIA-Surface-EO-SIE-004: Forward-Link Command Files
- –OIA-Surface-EO-SIE-006: Sequence Delivery Guidelines
- –OIA-Surface-EO-SIE-007: File Load Delivery Guidelines
- –OIA-Surface-EO-SIE-008: SCMF Reverse Translation
- –OIA-Surface-EO-SYS-003: SPICE SCLK Delivery Guidelines
- –OIA-Surface-EO-SYS-004: SPICE Spacecraft FK Delivery Guidelines
- –OIA-Surface-EO-SYS-005: SCLK_SCET_Time Correlation
- –OIA-Surface-EO-SYS-006: SPICE LSK Delivery Guidelines
- –OIA-Surface-EO-SYS-007: Executed Activity Update
- –OIA-Surface-EO-THRM-002: Thermal Model and Predictions
- –OIA-Surface-EO-THRM-003: Battery Diurnal Temperature File
- –OIA-Surface-EO-THRM-004: MMRTG Diurnal Temperature File
- –OIA-Surface-EO-THRM-005: OBP Strategic Thermal Tables
- –OIA-Surface-EO-THRM-006: OBP Tactical Thermal Tables
- –OIA-Surface-EO-TUL-002: Tactical Predict Files
- –OIA-Surface-EO-TUL-003: Tactical Waiver Request Guidelines
- –OIA-Surface-EO-TUL-004: Tactical Waiver Submission
- –OIA-Surface-EO-TUL-005: Onboard List Update
- –OIA-Surface-EO-TUL-006: Mech Initial Conditions for Copilot
- –OIA-Surface-EO-TUL-007: Integrated RKSML Delivery
- –OIA-Surface-EO-TUL-008: Tactical Flight Rule Disposition Reporting
- –OIA-Surface-EO-VSE-005: Propagated RKSML Files
- –OIA-Surface-EO-VSE-006: Propagated NPM

IO (7)

- –OIA-IO-IDSO-001: EDR & RDR Files
- –OIA-IO-IDSO-002: Image and Terrain Mesh RDRs
- –OIA-IO-IDSO-004: IDSO Horizon Profiles
- –OIA-IO-IDSO-005: Localized Products for RO
- –OIA-IO-IDSO-006: Localized Products for EO
- –OIA-IO-IDSO-007: Geotiff for Localization
- –OIA-IO-IDSO-008: Contextual Mesh

SO (2)

- –OIA-SO-PILOT-001: Tactical Activity Plan
- –OIA-SO-PILOT-002: Activity Review Products and Files

RO (5)

- –OIA-RO-004: Uplink Command Sequences
- –OIA-RO-005: Uplink File Loads
- –OIA-RO-010: Duration/Resource Estimate for Plan
- –OIA-RO-011: Simulated Uplink Animation
- –OIA-RO-012: Simulated Uplink RKSML

Software Interface Specifications inclusive with OIAs, not enumerated separately

Criteria for MSM Rollback

- How much longer of a shift are we okay with?
  - For the first day – 10 hours is ok as long as the problems identified are considered to be a one-off/transition growing pain
- How many sols are we willing to allow the shift length to be longer before we decide to switch back (this is probably dependent on the duration of the shifts)
  - Long shift = Single sol plans, 8-8.5 hours multisol plans 9+ hours
  - If multiple single sol plans in a row are stubbornly at 8-8.5 hours then reassess
    - Identify if it is a single area/process or the accumulation of systemic problems (i.e. workaround stack up)
  - Decide by Thursday declare plan if Friday 3-sol plan go/no-go.
    - Determine if additional complexity reduction is needed (temporary for first couple 3-sol plans)
- Is "long shift" a reason to switch a tactical plan from SP1 to MSM? (meaning is there a cutoff time in the planning day in which we pivot to MSM or do we keep trying to the shift cutoff)
  - See above
- What other criteria might we use to decide to switch back to MSM mode? (this will depend on where we land with the various things on the checklist)
  - Ground: SSIM issues that leave an open vulnerability in command validation
    - eg: considered set vs gatekeeper vulnerability
  - Ground: Bin 3 things just add up to be too many workarounds
    - workaround math: (complexity) X (duration of workaround) summed by role = role pain, role pain summed by shift
  - Flight: If thermal even so much as looks at us weirdly, we will likely go back to M/SM (MSTRIAGE-10074)

Ground Software Used in Planning

APSS	CS3	EAS	IDS	MCS	RPS	EOTT	EOTT
AD Editor	Cyber Security	dplib	CAMP	AMPCS	ASTTRO	DKF	SIE Reports
COCPIT	EOP	Downshfit Dasboard	CRISP	DMTK	ROCS Spice API	System Status	RuleCheck
Copilot	OCS	MASH	SciLo	MCS Pass Monitor	RSVP	AutoRML	Publish Predicts
CrossCheck	OCS API	Parasol	C-POSE	DMT API	SSIM	NPM Propagation	Timeline Stats
RuleHub	OCS CLI	ROUNDS	DataDrive		TargetDB	Publish AsRun	API Test
SeqVal	GDS Infrastructure	ROUNDS Rule Editor	EDRgen	SOAS	CASPIAN	Publish Copilot Incon	
SeqGen	CACHER	Transpire	IDS Pipeline	iROCS	ACA Viewer	TUL Morning Report	
Sequencer	Meeting Tracker	Comm Tracker	LandForm	Science Dashboard	ArmSketch	COCPIT Helpers	
Uplink Dashboard	RASCAL	Channel Viewer	MarsViewer	Viewpoint	MobSketch	APAM Reports	
Uplink Services	CACHER API	EVR Viewer	PLACES	Jupyter NB	Periscope	Thrm Relation Gen	
Uplink Tools	CSSO	Epoch Viewer	RDRgen			Thermal Selection	
Waiverly			Terrain Mesh			Warmup Table Mgr	
MSEQ			Outreach Lambda			SPOS	
			ROLO			Prep Bundle	

Scoped but deferred to post-landing

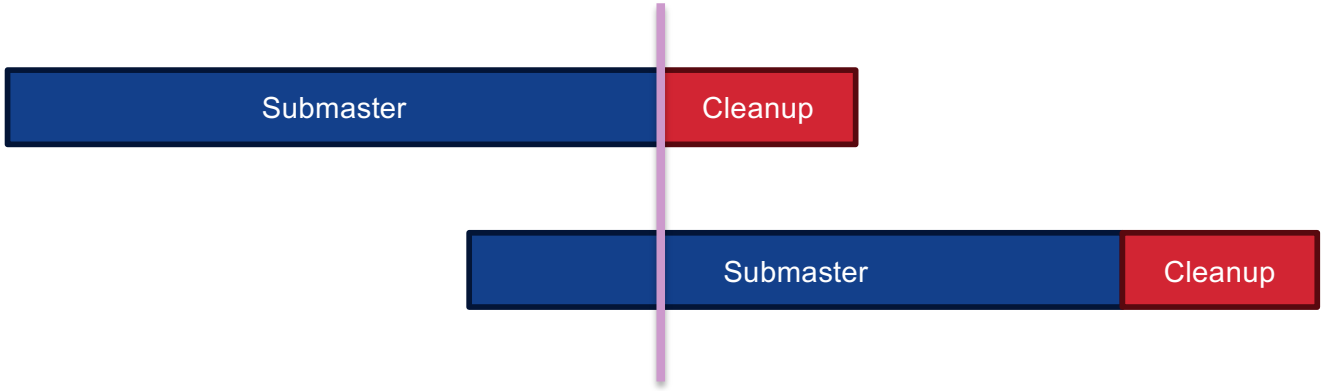
Unscoped Pre-landing Tool

Updated for Simple Planner

No change

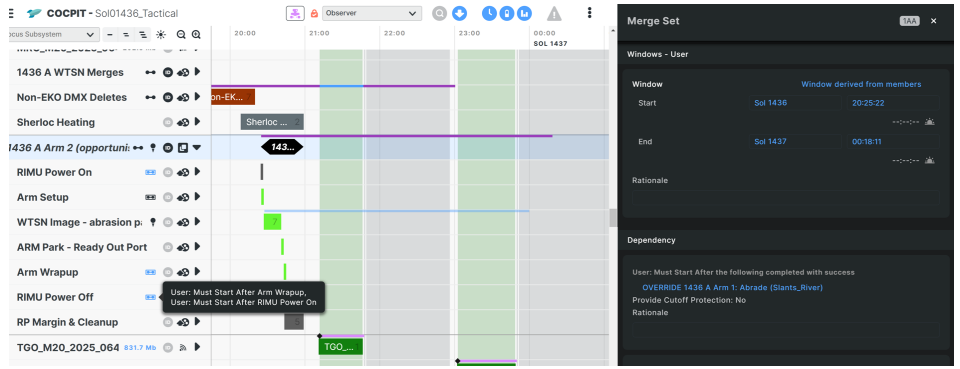
M20 Cleanup Structure

In order to allow for more flexibility in scheduling we restructured the deactivates in cleanups to be specific to the sequences being called in the submaster.
In previous missions cleanups would do a global deactivate which would cut off any sequences (other than the master) that were running at the time of the cleanup.
This had the side affect of allowing easier parallelism which generally benefited the planning process regardless of MSM or OBP modes.



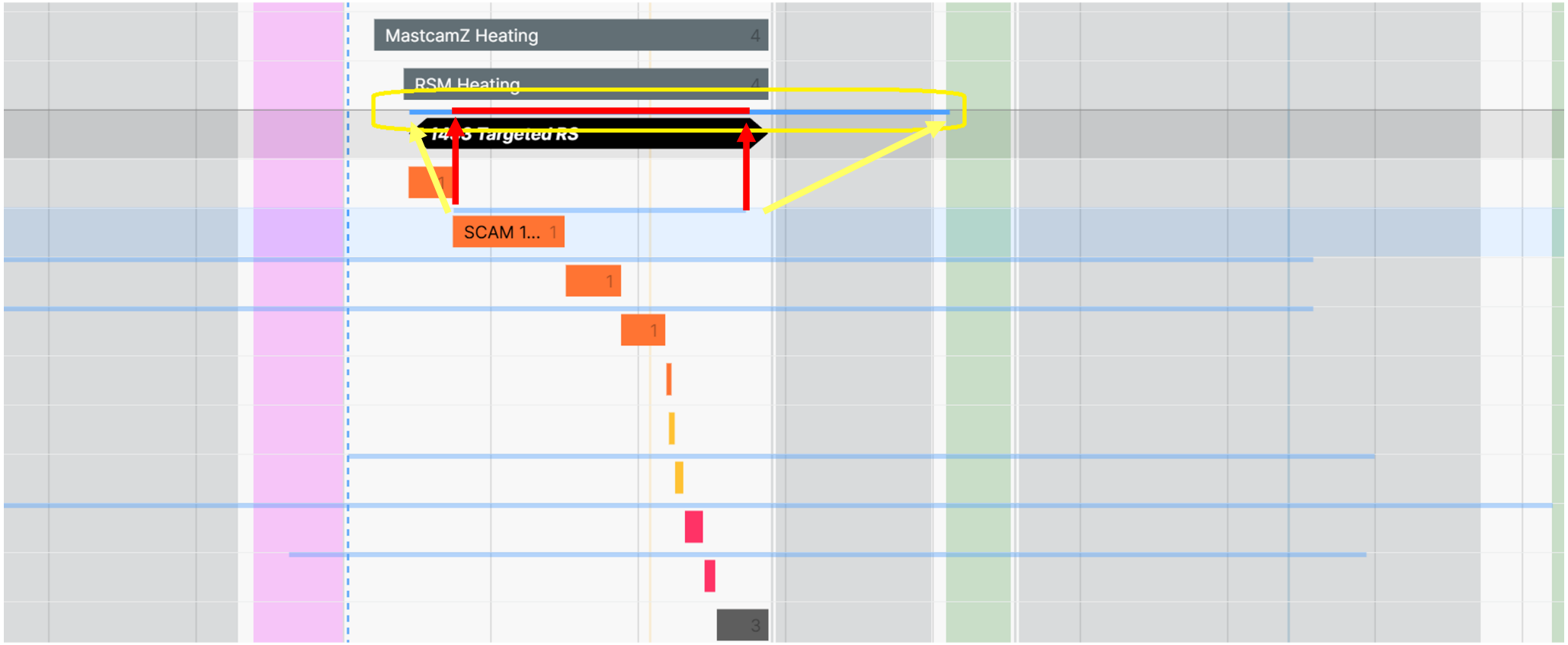
M2020 CBP

- Uses Copilot, which forces constraints, models power and automatically adjusts the plan at runtime. Under the hood uses Surrogate, which uses same algorithms as OBP FSW
- Onboard autonomy takes copilot generated planfiles and schedules what it can, when it can, adjusting this schedule throughout the martian day
- Feedback through CrossCheck and AutoPilot
- AEGIS autonomous targeting operational 2022+
- OBP in control 2023+
- End-to-end for rover FSW, file system, and robotic operations, limited automation for instruments

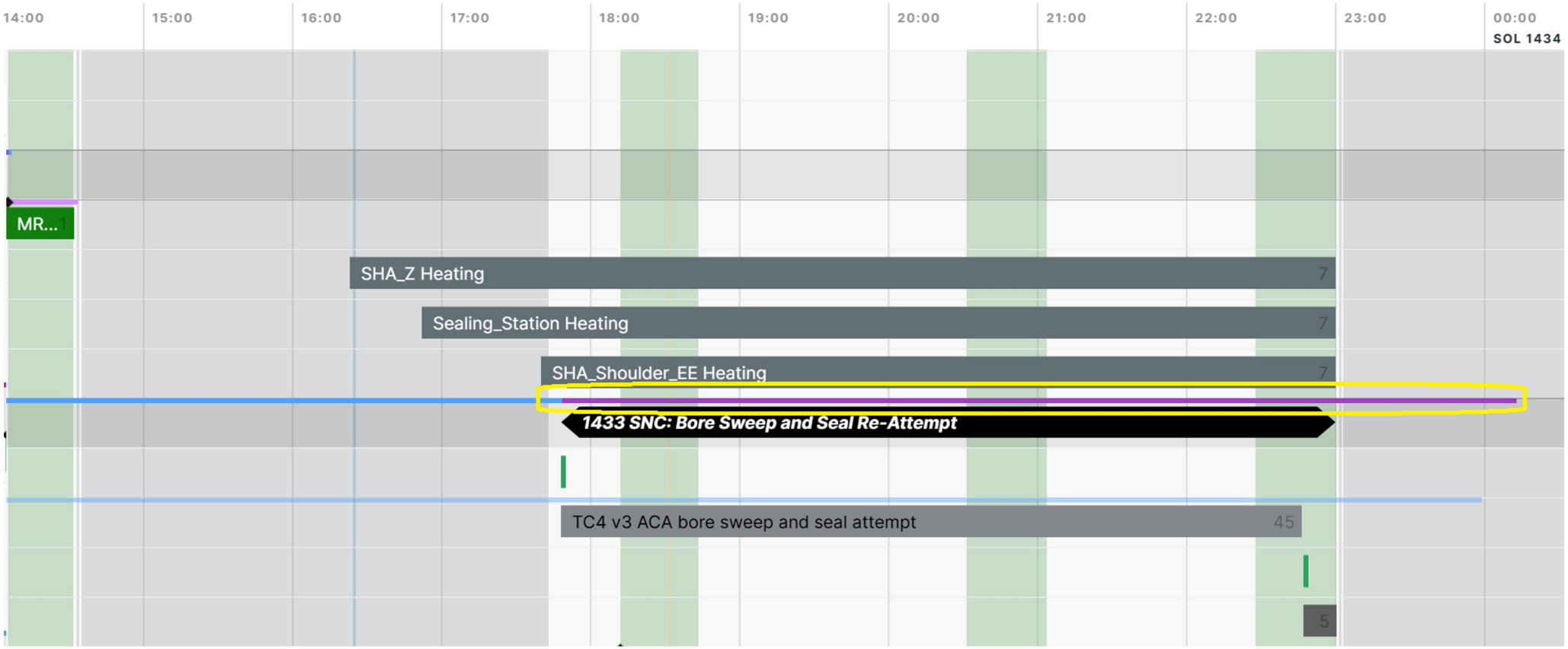


Any views and opinions expressed herein do not necessarily state or reflect those of NASA, JPL, or the California Institute of Technology.

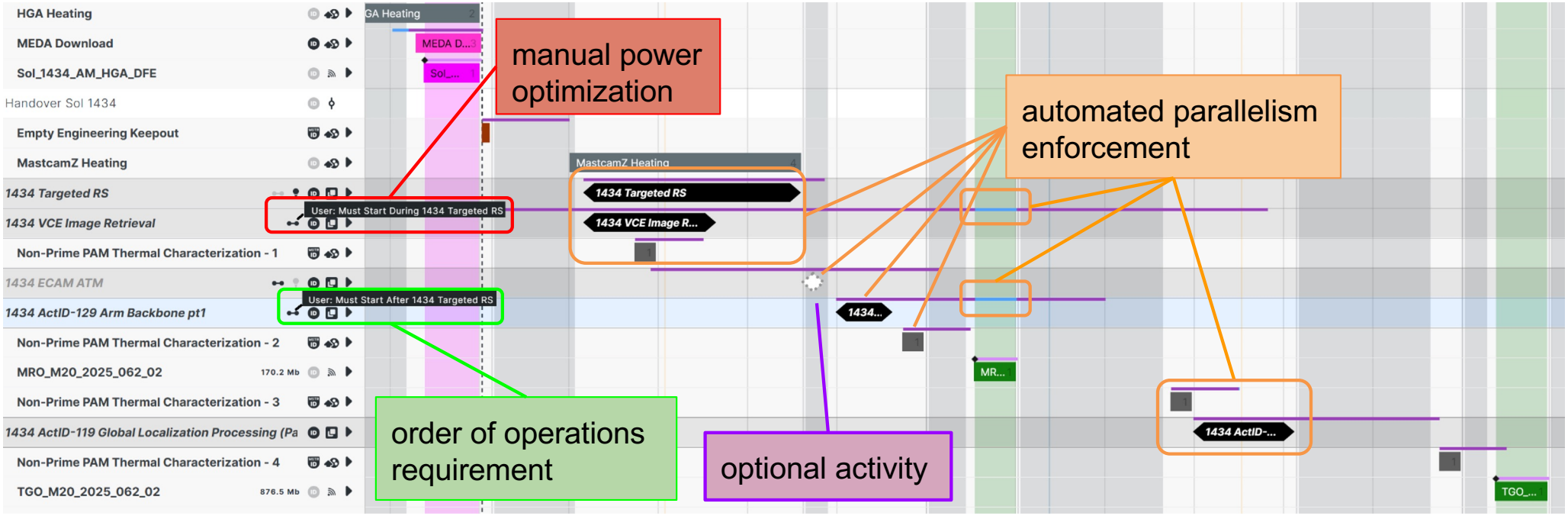
Automated Windows and Rollup



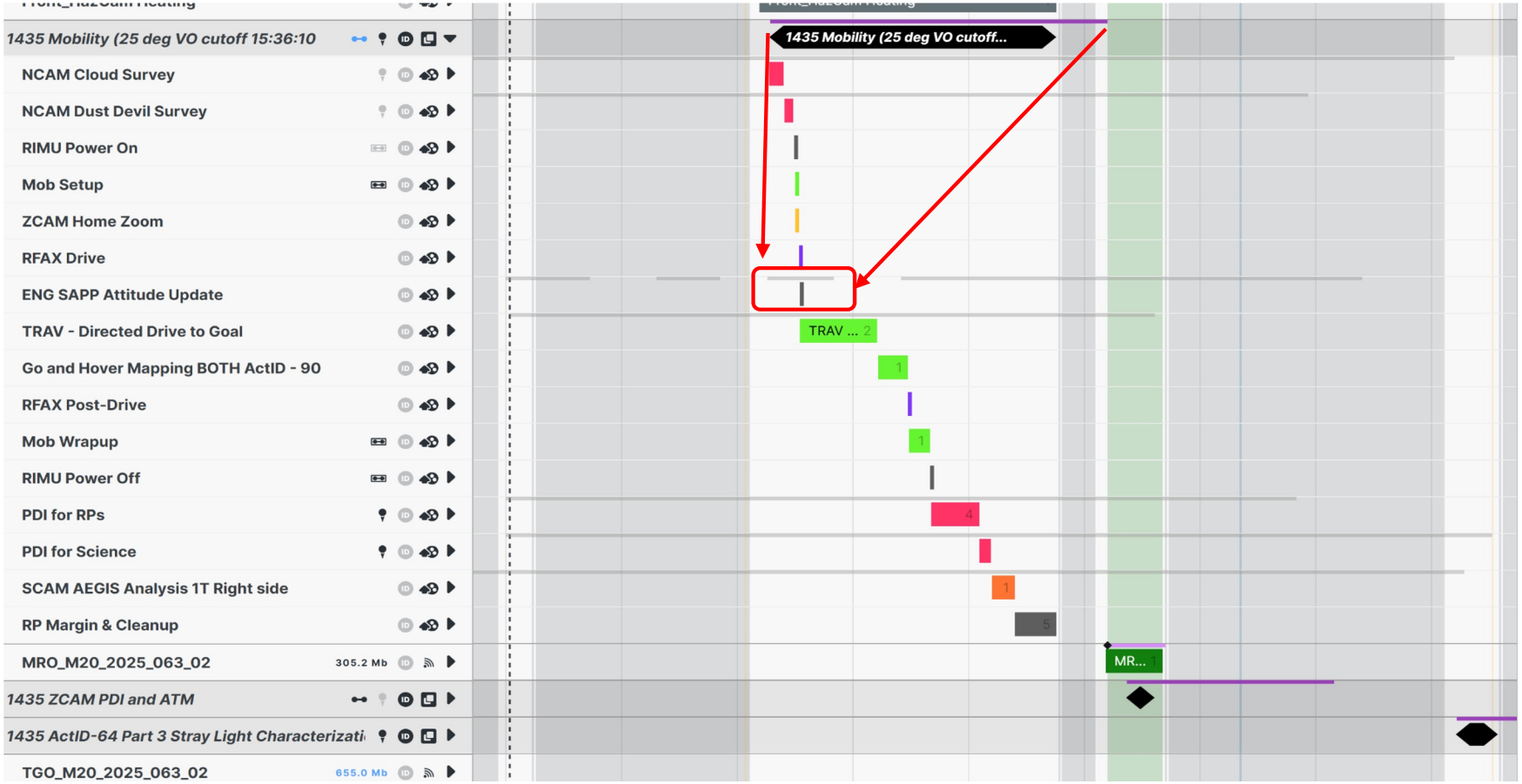
Automated Power Optimization



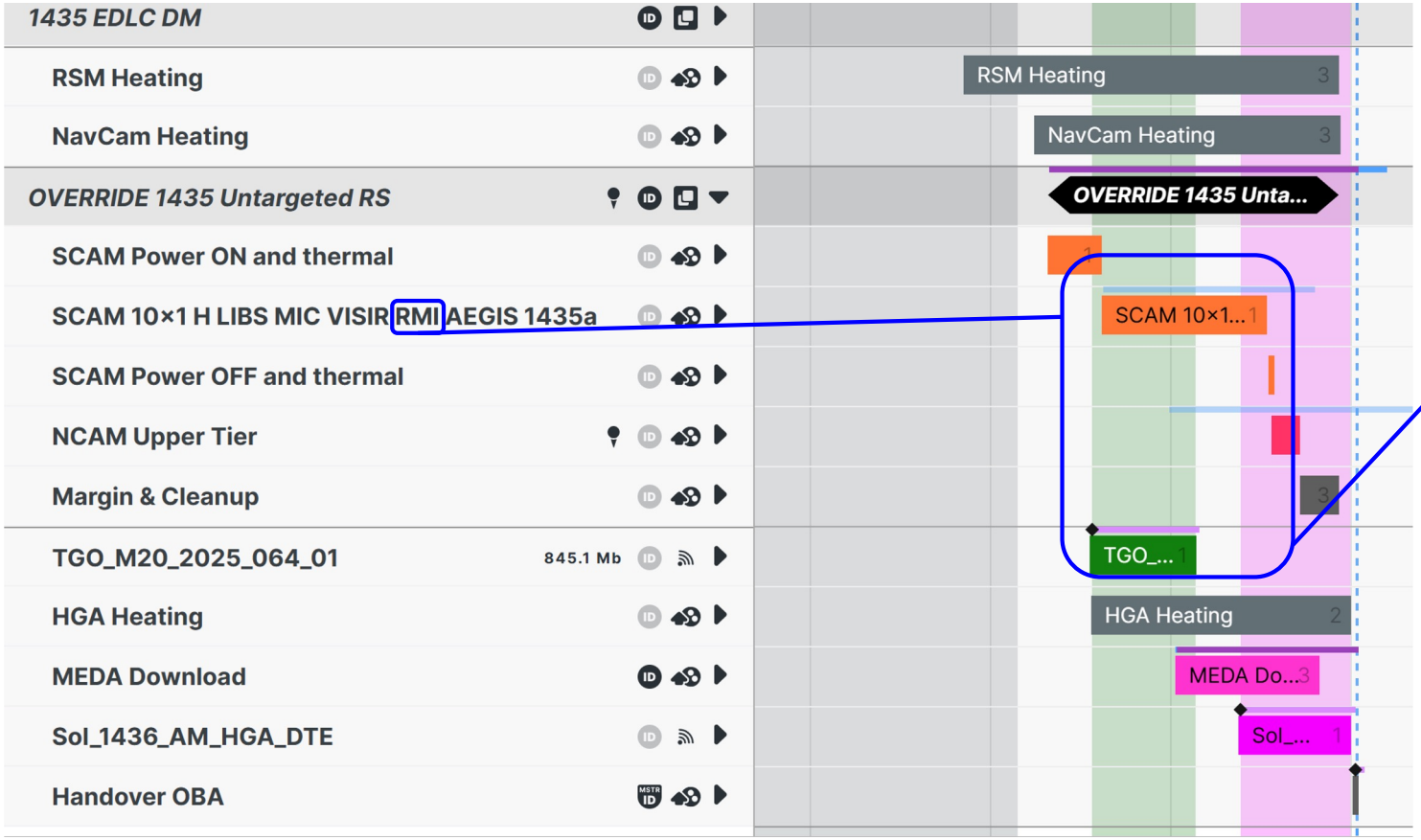
Multiple OBA Relationships



Manual “Eye-Balling”

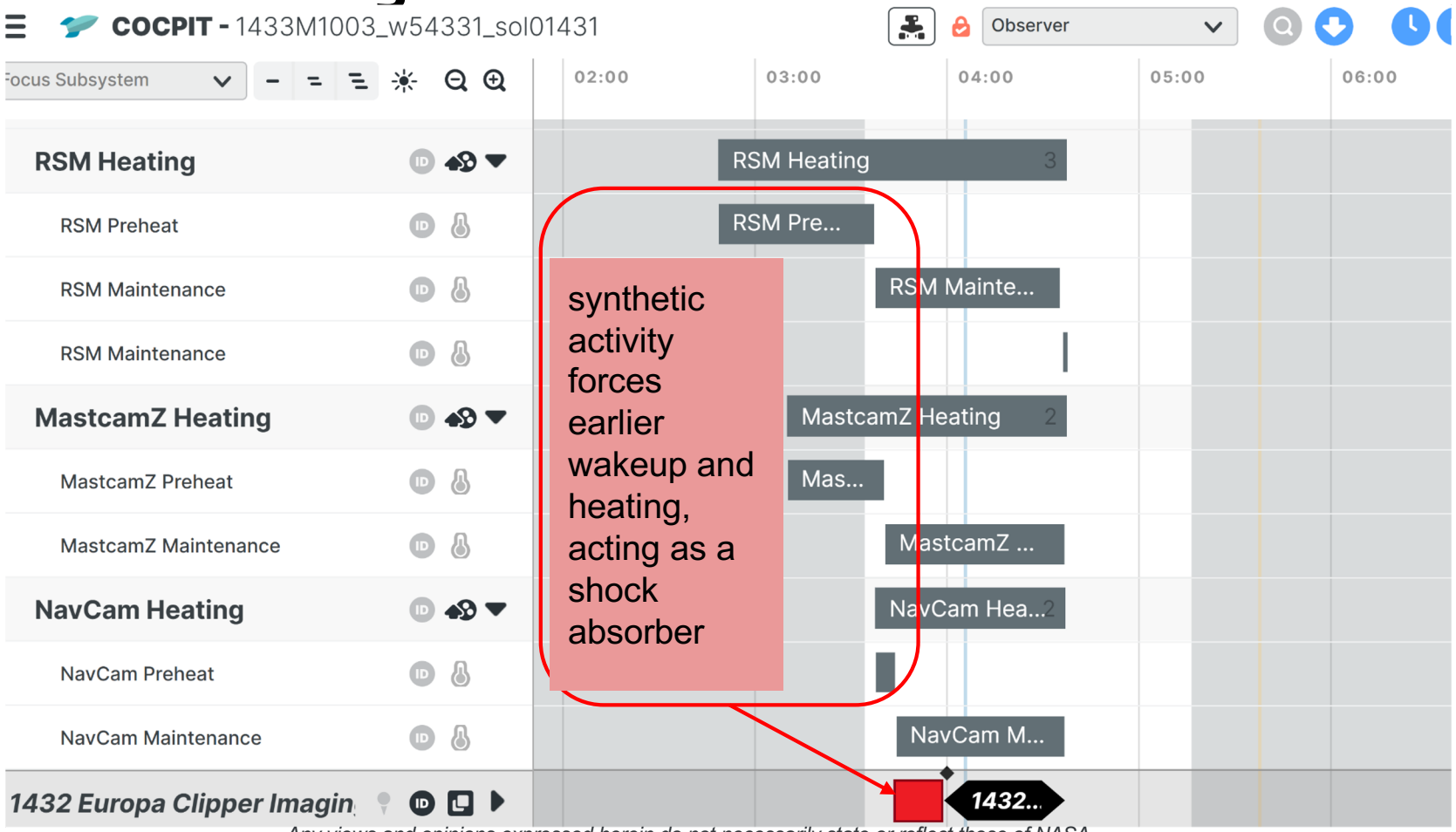


User Manual Override



Recent ActID re-enabled SuperCam RMI with UHF windows, in the donut hole period before automation catches up (in ~1-2 months)

Brittleness Mitigation



SP ORT day 1