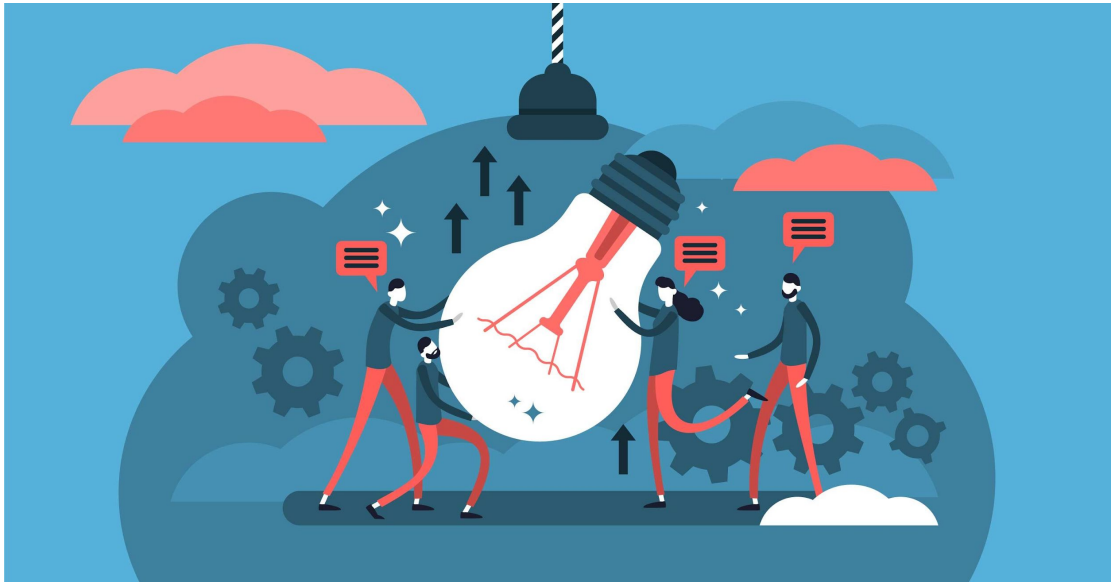
Navigation Field Triage Retrospective

Ammar

Agenda:

- Sorty: First contact with the real world
- Scaling up for the real world
- Where we are now
- Lessons Learned

Perception + Continuous Operations | another successful collaboration



Chaos! | making contact with the real world

Sorty@REWS

# of Robots	~ 10
Collective driving time / week	~ 15 hours

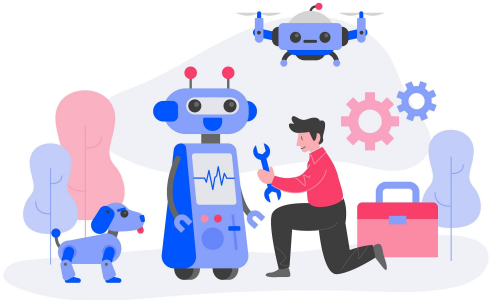
Cadence Mismatch | making contact with the real world

Strategic
~Months / Quarters

Tactical
~Hours / Days

Cadence Harmony | scaling up contact with the real world

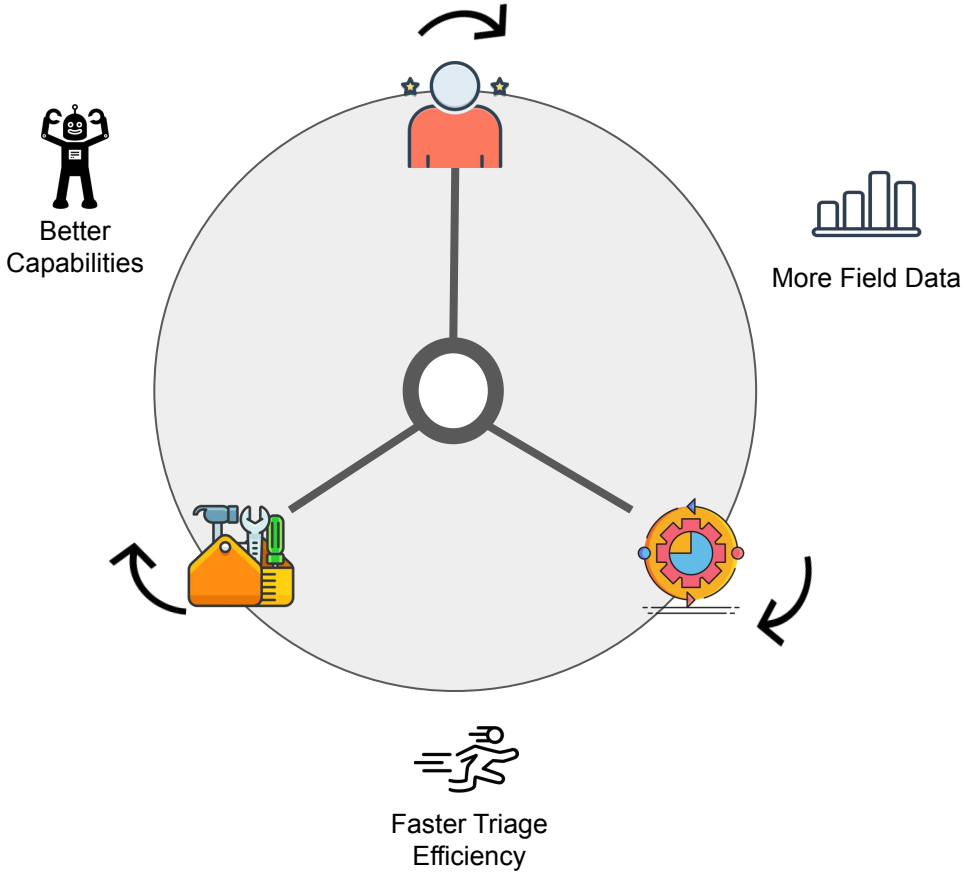
Strategic
~Months / Quarters



Field Triage
Happy Medium :)

Tactical
~Hours / Days

Procedural Flywheel | getting ready for 10X growth



Where We Are Now | spearheading field triage for navigation

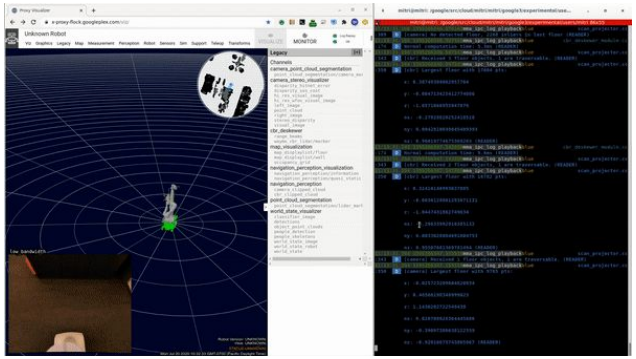
- An established protocol and a dedicated (& terrific) triage champion leading the charge!
- Detailed living documentation on triage operating procedures thereby alleviating the bottleneck to information.
- All bugs (≤ P2) addressed within 48 hours.

mitri@

How to Debug Navigation Bugs

Process:

1. **Log Visualizer**
   a. Open Log Visualizer URL link from the bug.
   b. Check **information** and **quasi_static** channels for any ghost or unusual points near the robot that might potentially be blocking the robot from navigating.
   c. Check the **occupancy_grid** channel to look for any map issue and double check robot localization.
2. **Download Logs**
   a. Open Timeline URL link from the bug.
   b. Select the ipc logs that correspond to the same timing of the incident
   c. Copy the directory path starting with "*proxyfleetcontrol-logs/*"

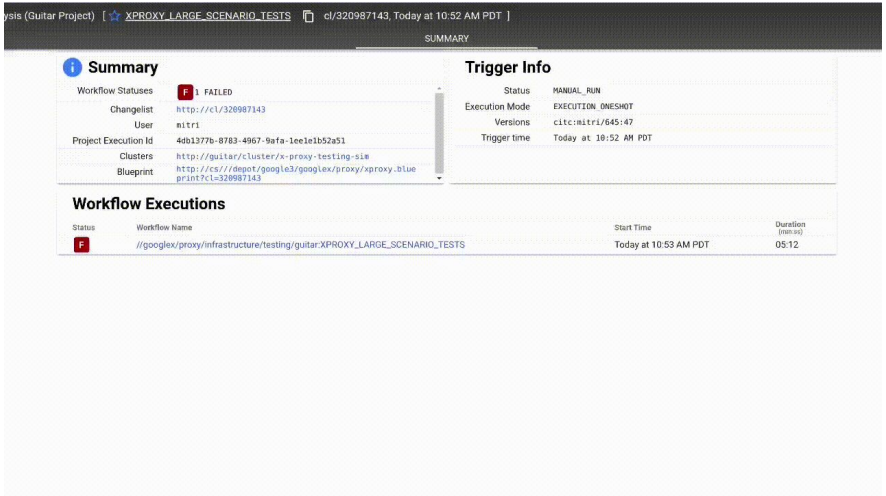


Where We Are Now | spearheading field triage for navigation

- Every incident meticulously categorized & logged.
- Regressions are now spotted earlier for more immediate action.
- Guided the design of scenario evaluation & testing for navigation perception.

Weekly summary of Nav bugs

	Navigation Run Time (hours)	Total count of Bugs	Capability Limitation							Upstream noise				Regression	True failure	Miscellaneous
Week			Ego Motion / Jerkiness	Missing Semantics	CBr sensing limitation	Camera sensing limitation	Sensing blindspot	Ego occlusion	Total	CBr	Camera	Map	Total			
7/20/2020	0.17	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0
3/9/2020	2.91	6	0	0	0	0	0	0	0	0	0	0	0	0	5	1
3/2/2020	7.85	14	1	0	0	0	2	0	3	4	1	0	5	0	4	2
2/24/2020	9.71	14	1	3	0	0	0	0	4	1	0	0	1	0	5	4
2/17/2020	5.59	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2/10/2020	8.76	9	1	0	0	0	0	0	1	0	0	0	0	0	1	7
2/3/2020	9.01	18	4	0	0	0	0	0	4	1	11	0	12	0	1	1
1/27/2020	7.54	3	0	0	0	0	0	0	0	0	3	0	3	0	0	0
1/20/2020	6.03	6	0	0	0	0	1	0	1	0	5	0	5	0	0	0
1/13/2020	14.15	4	0	0	0	0	0	0	0	0	2	0	2	0	0	2
1/6/2020	13.46	5	0	0	0	0	1	0	1	0	4	0	4	0	0	0



Lessons Learned | how can other domains in Proxy benefit from a field triage process?

- Categorized field data is already guiding the design & development of new features.
  - Eg: Pan-tilt calibrations, sensor blindspots etc.

- Room for **10X** improvement in shortening triage times with the right set of tools, thereby making the flywheel go even faster.
  - **Goal:** <7 mins / bug

- Specialization is important! Given the complexity of the system it's better to train a few people to dive deeper than training all of CO.

- Helped us make the transition from direct real world contact where most of the bugs are immediately fixable to a stage of addressing the long tail of issues that need to be statistically prioritized in upcoming feature releases.

Some useful info to apply to your domains:

- go/proxy-nav-triage-doctrine
- go/proxy-nav-field-report
- go/proxy-nav-debug process
- go/proxy-nav-triage-retrospective