fetch robotics

Physical Continuous Integration

CI with Real Robots!

Alex Henning

10.09.2016

The Problem

- **Using best practices**
  - Continuous integration
  - Code reviews
  - Linters
- **Releasing took a lot of time and *manual* effort**
  - Verifying robots behaved appropriately
  - Slow feedback cycle
  - Real world interactions caused issues
- **We want releases to be *fast* and *robust***

The Solution

Software Continuous Integration

The process of automatically verifying each change, allowing teams to detect problems early.

Physical Continuous Integration

The process of automatically verifying each change on ***real robots***, allowing teams to detect problems early.

What are We Testing?

fetchcore: fleet management

freight: mobile platform

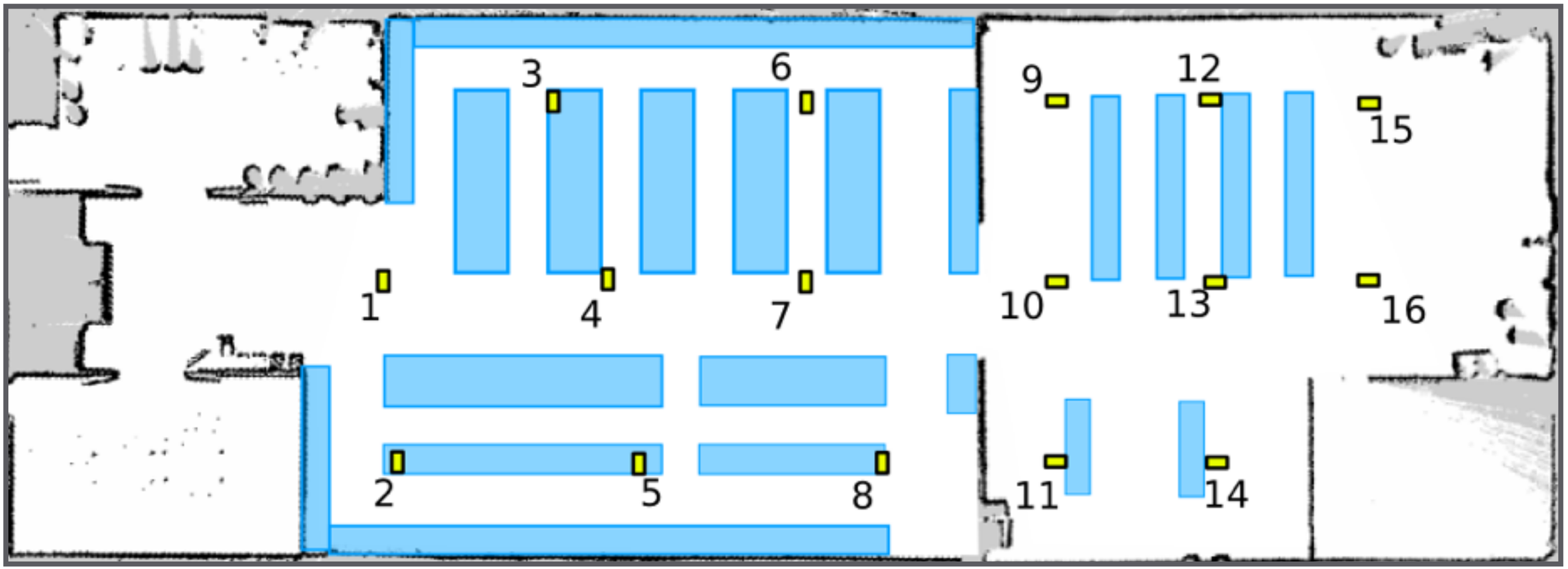
Our Setup

Test Facility

- 7500 Sq. Ft. warehouse
- Multiple robots
- 16 cameras

Servers

- Bag file server
- Video server
- ***fetch***core servers



Testing a Change

- Pull in changes from ROS
- Test locally
- Start physical continuous integration

And…

- Some robots refuse to move

What's Wrong?

Available resources:

- Robots are always bagging and logging
- Cameras are always recording
- Tools to get relevant bags and video

What's Wrong?

Let's investigate:

- Logs:

```
[WARNING] Global plan in collision, replanning
[WARNING] Global plan in collision, replanning
```

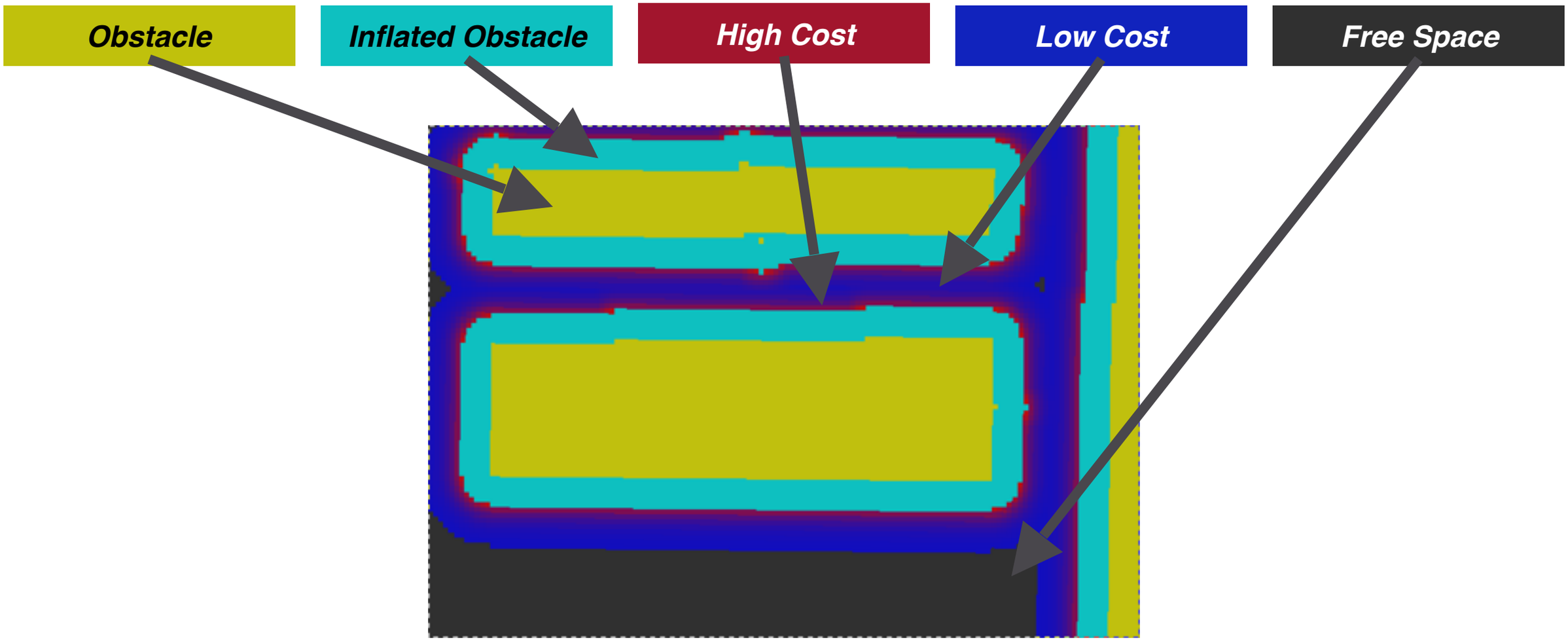
  …

- Cameras:

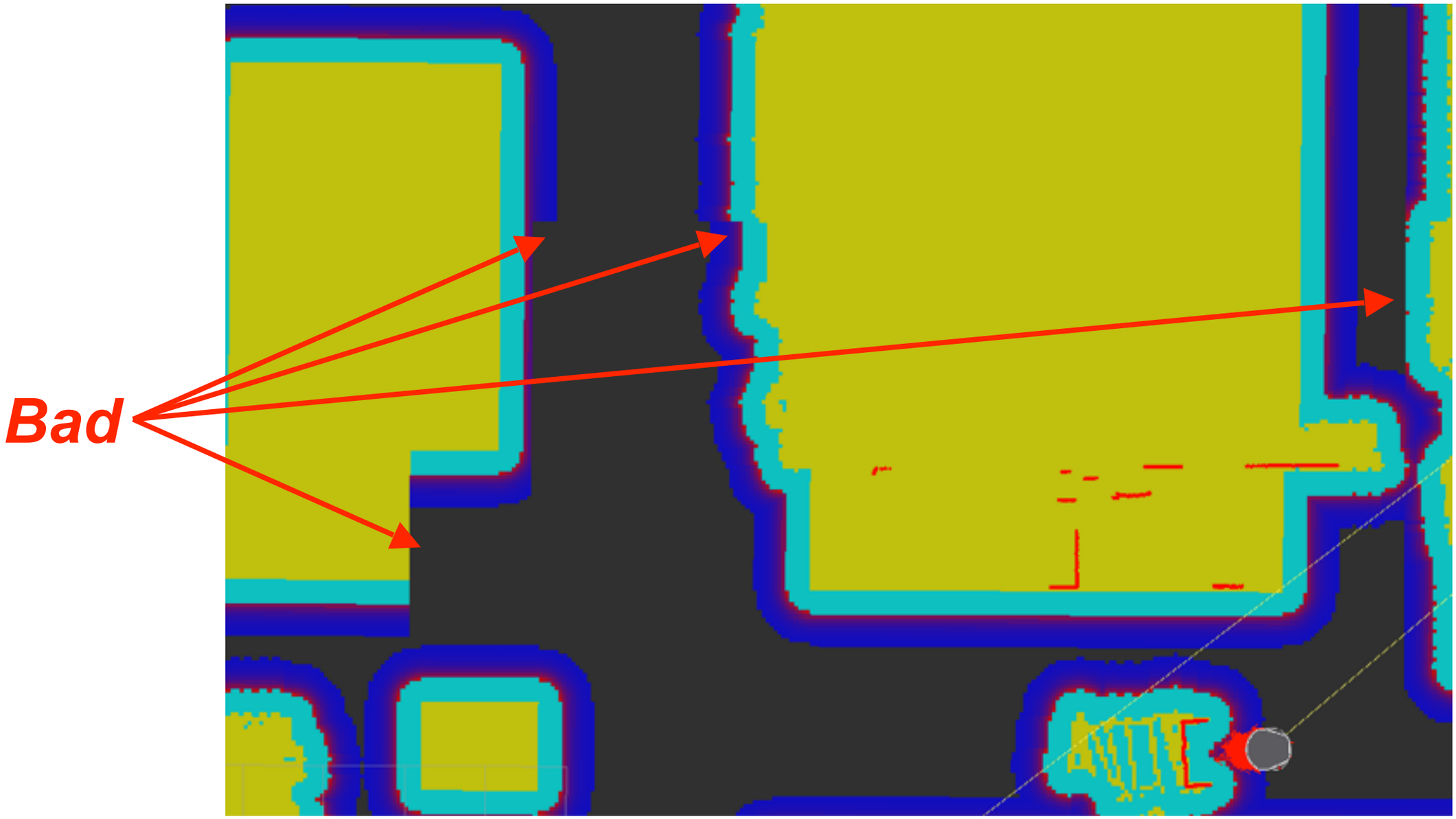
- Bag files and RViz

A Brief Introduction to Costmaps

- Part of the ROS navigation stack
- Used in planning paths
- Helps avoid crashing



What the Bug Looks Like



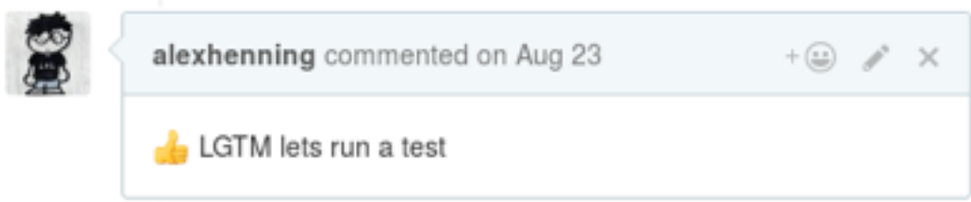
Now What?

We know what to look for

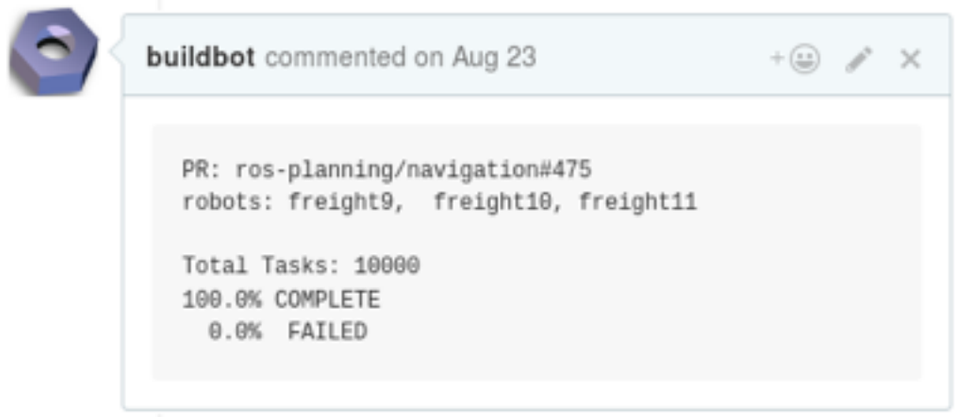
1. Reproduce the bug
2. Develop a fix
3. Verify fix
4. Software continuous integration
5. Code review

Physical Continuous Integration

- Need a test
  - Automated
  - Catches the problem
  - Avoid regressions
- Run test in the warehouse
- Results get posted on github




```
>>> Preparing test of ros-planning/navigation#475
>>> freight9 is ready
>>> freight10 is ready
>>> freight11 is ready
>>> fetchcore is ready
>>> Starting test
...
>>> Test is done
>>> Posting results to github
```



Problem Solved

All tests pass

Fix costmap artifacts #475 — Edit

Merged **mikeferguson** merged 2 commits into `ros-planning:indigo-devel` from `alexhenning:fix-costmap-artifacts` on Jun 7

Conversation 0 · Commits 2 · Files changed 2

Changes from **all commits** · **2 files** · +6 −6 — Unified | Split — Review changes

8 costmap_2d/plugins/inflation_layer.cpp — View

```
@@ -150,10 +150,10 @@ void InflationLayer::updateBounds(double robot_x, double robot_y, double robot_y
150 150     last_min_y_ = *min_y;
151 151     last_max_x_ = *max_x;
152 152     last_max_y_ = *max_y;
153     -   *min_x = std::min(tmp_min_x, *min_x);
154     -   *min_y = std::min(tmp_min_y, *min_y);
155     -   *max_x = std::max(tmp_max_x, *max_x);
156     -   *max_y = std::max(tmp_max_y, *max_y);
    153 +   *min_x = std::min(tmp_min_x, *min_x) + inflation_radius_;
    154 +   *min_y = std::min(tmp_min_y, *min_y) + inflation_radius_;
    155 +   *max_x = std::max(tmp_max_x, *max_x) + inflation_radius_;
    156 +   *max_y = std::max(tmp_max_y, *max_y) + inflation_radius_;
157 157   }
158 158 }
159 159
```

Your Turn

Ansible
- Configuration
- Orchestration
- Manages many machines

ZoneMinder
- Records video
- Detects motion
- 18 TB of storage

Custom Software
- Ties it all together
- Integrates with github
- Gathers data
- Tools for investigating

Pitfalls

Need to deal with the real world

- Crashing
- Delocalization
- Batteries
- Deadlocks

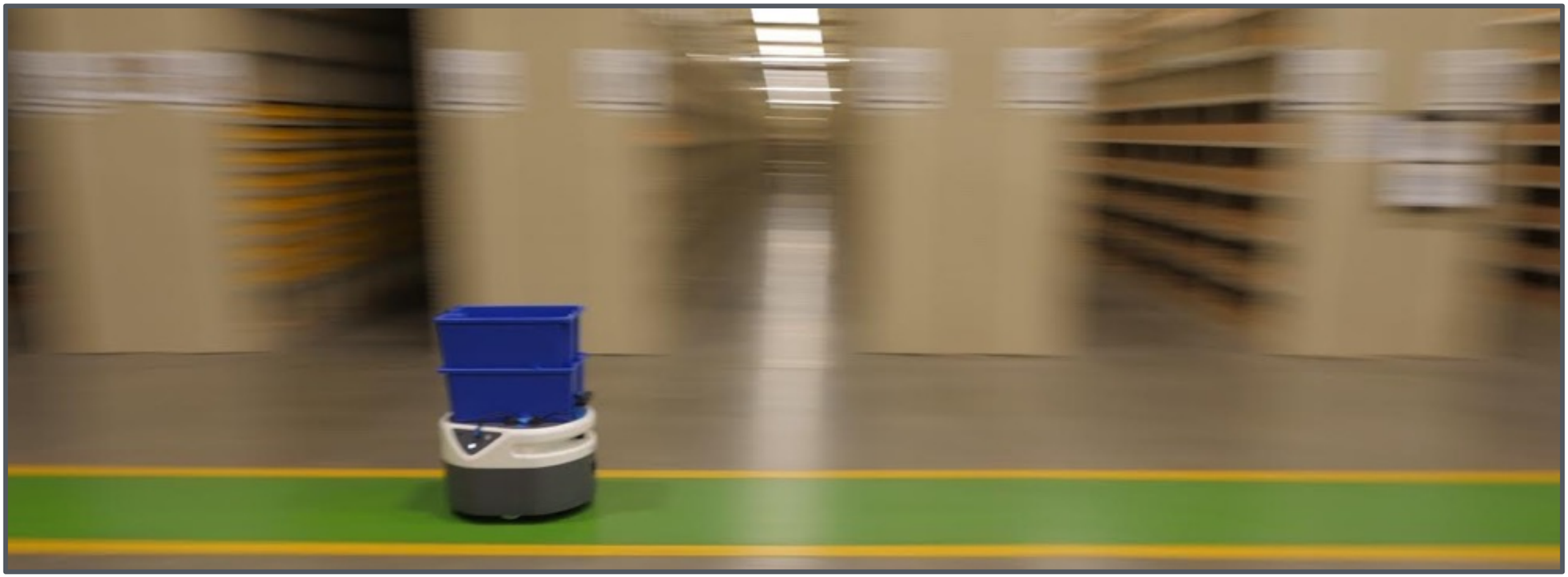
To reduce problems

- Code review
- Software continuous integration
- Early termination

Final Remarks

Physical Continuous Integration

- Part of the a good development workflow
- Not a silver bullet
- Anyone shipping robots should integrate it



Questions