# Eye In The Sky

PRESENTATION 2 – DECEMBER 4, 2014

SPRINT 1&2 RECAP, SPRINT 3 REPORT

# Team

Colter Assman: UAV Specialist

Julian Brackins: Scrum Master

Charles Parsons: Nav Specialist

Alex Wulff: Technical Lead / Demolitions Expert

Product Owner & Team Advisor: Dr. Jeff McGough

# Overview

Project Overview

Sprint Timeline

Open CV

SLAM

GPS

ROS + Gazebo

Testing

Deliverables

# Project Overview

# Terms

# Terms

UAV – Unmanned Aerial Vehicle

UGV – Unmanned Ground Vehicle

ROS – Robot Operating System

- Node – ROS subroutine or program for UAV control

OpenCV – Open Computer Vision

- Library for real-time computer vision

SLAM – Simultaneous Location And Mapping

SVO – Fast Semi-Direct Monocular Visual Odometry

# User Stories

As a member of a search and rescue team:

- Vehicle to assist with finding missing persons in areas of low radio communication / high human risk
  - Black Hills
  - Forest Fires
  - Radioactive Locations

As a user of the UAV:

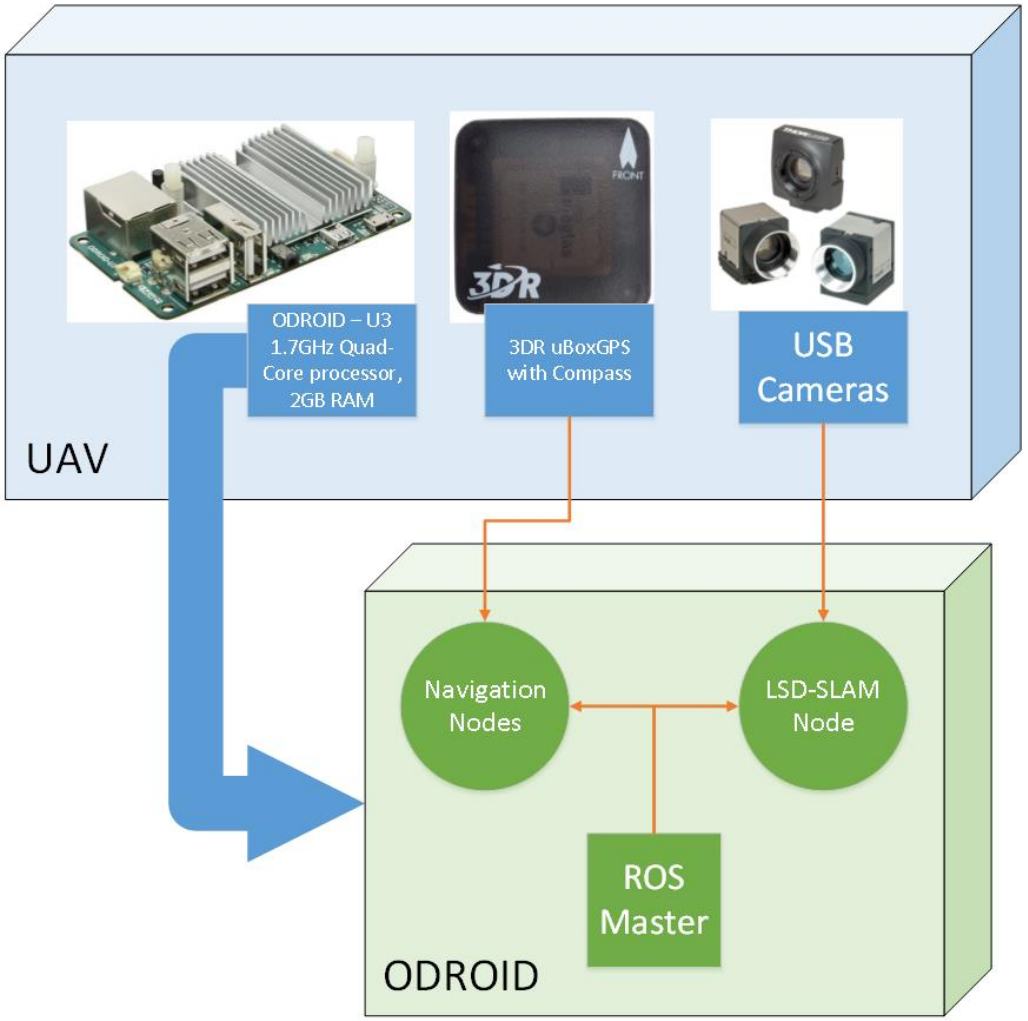
- UAV launches from landing pad located on vehicle
- UAV collects data and sends to base station
- UAV autonomously locates landing pad
  - Landing Pad situated on UGV
  - automatically recharges UAV battery

# Intellectual Property

The claim to the IP is shared between the Sponsor Jeff McGough and the members of the Eye In the Sky Senior Design Team: Colter Assman, Julian Brackins, Charles Parsons, and Alex Wulff.

# System Diagram

Note: omitted UGV design



# UAV / UGV Design Details

# UAV Specifics

- The UAV is currently equipped as:
- AMP 2.6 - flight control board
- 3dr ublox GPS with Compass - compass and gps
- R5800X receiver and TXV582 - video transmitting
- Sony HAD 520 line camera - on board camera
- Spektrum AR7000 - 7 channel receiver
- 915 MHz radio - telemetry with ground station

# UGV

- The UGV is to be equipped as:
- GPS – Make and model still under investigation
- SLAM Cameras (Asus Xtioc & Microsoft Kinect)
- Recharging Station with mounts
- Tilting platform with LED indicators for low light environments.
- Power supply
- Odroid computation.

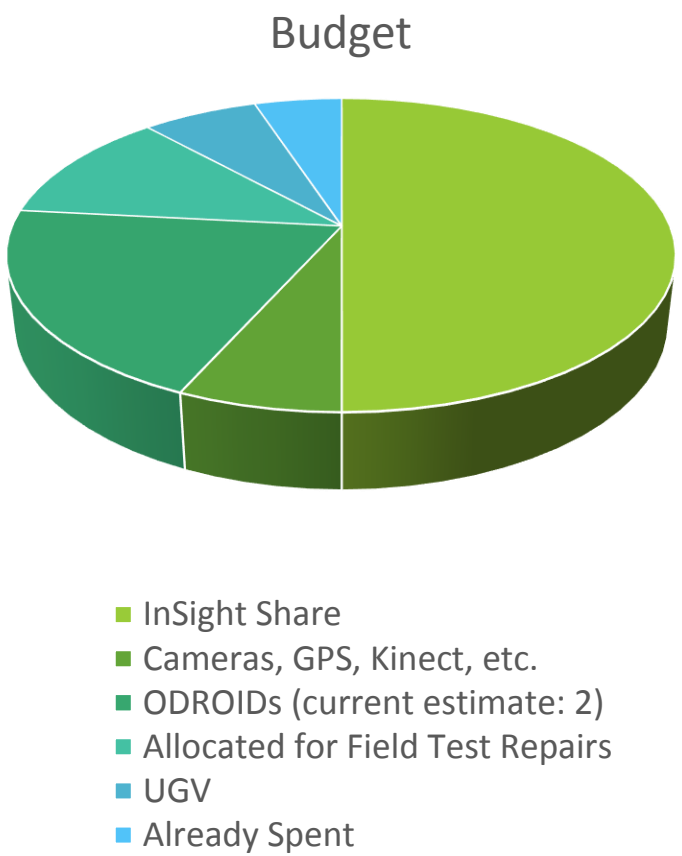
# Risk Assessment

BUDGET, OBJECTIVES/CONSTRAINTS

# Budget

$1,500 shared between inSight / EITS

- GPS for UGV + other cameras
- ODROIDS (more than 2 = over budget)
- Allocate budget for vehicle repairs
- UGV Construction
- Colter has bought 2 new battery's and XT-50 connectors for a total of $75.50.



# Budget Notes

Most of our budget has been mitigated through certain means:

- UAV from Department
  - Needs repairs but cheaper than designing outright
- Cameras from Department
- Simulation
  - Gazebo
  - Lower costs of repairs by fine tuning software in sims

# Objectives / Constraints

Objectives:

- Autonomous UAV flight
  - In Progress:
    - Designing ROS nodes to handle UAV nav behavior, routines
      - Python code – nodes publish Twist messages through cmd_vel
- Landing Pad detection
  - In Progress:
    - Explored multiple methods for object detection
      - OpenCV (Homography)
      - SLAM (mapping + object detection)
- UAV Auto-Charging
  - Induction plates or charging mechanisms.  Not on schedule until Sprint 5.

# Objectives / Constraints

Constraints:

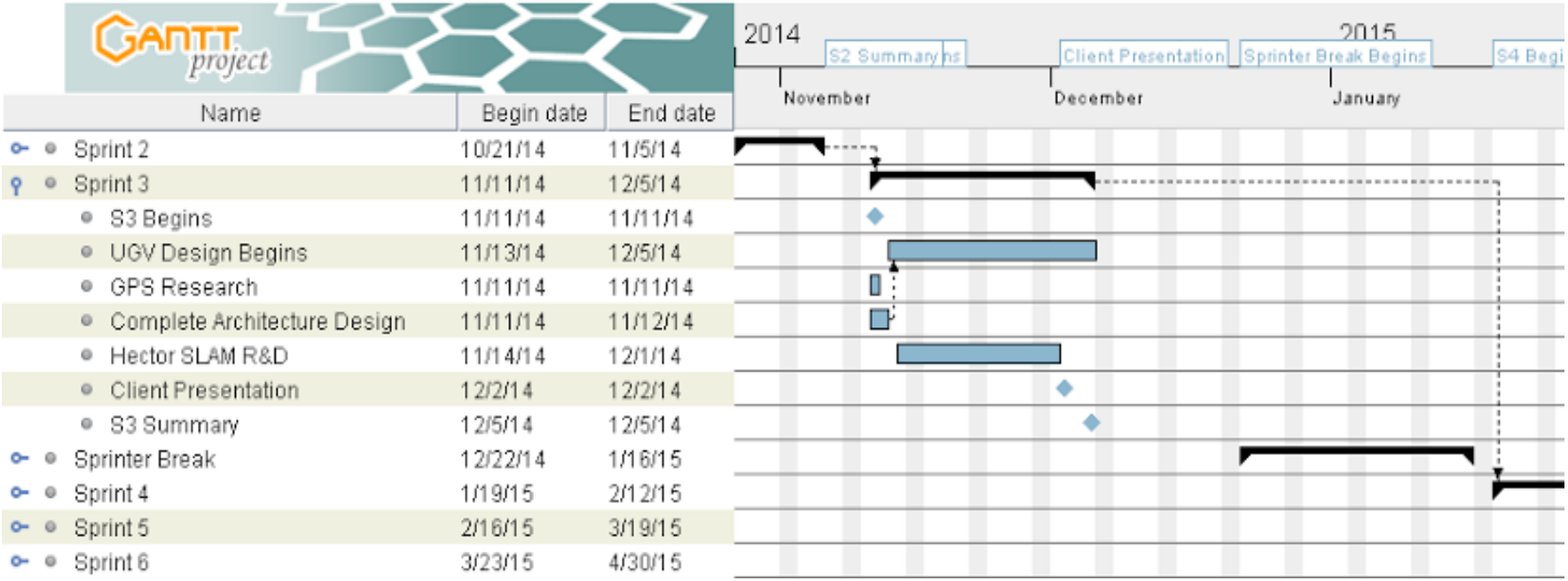
- ROS = HARD
  - Setting up environment
  - Ubuntu setup, etc.
  - Familiarizing team with ROS
- UAV needs repairs
- Testing the UAV / UGV could result in vehicle damages
  - Sims will reduce expenditures.
- UAV Weight
  - How many cams, how many computers, gps weight,  battery, etc.
  - Stress testing ODROIDs to gauge how may we'll need
    - Running SLAM, navigation nodes, cameras, etc.
- ODROID
  - Current Ubuntu 14.04 does not come with graphical desktop environment on ODROID
    - No graphics, no camera feed...

# Sprint Timeline

# Sprint 1 & Sprint 2 Recap

- Sprint 1:
  - Largely research / requirements gathering
  - ROS environment setup
- Sprint 2:
  - OpenCV research
    - Object Distance Estimation
  - SVO Analysis

# Sprint 3




| Name | Begin date | End date |
|---|---|---|
| Sprint 2 | 10/21/14 | 11/5/14 |
| Sprint 3 | 11/11/14 | 12/5/14 |
| S3 Begins | 11/11/14 | 11/11/14 |
| UGV Design Begins | 11/13/14 | 12/5/14 |
| GPS Research | 11/11/14 | 11/11/14 |
| Complete Architecture Design | 11/11/14 | 11/12/14 |
| Hector SLAM R&D | 11/14/14 | 12/1/14 |
| Client Presentation | 12/2/14 | 12/2/14 |
| S3 Summary | 12/5/14 | 12/5/14 |
| Sprinter Break | 12/22/14 | 1/16/15 |
| Sprint 4 | 1/19/15 | 2/12/15 |
| Sprint 5 | 2/16/15 | 3/19/15 |
| Sprint 6 | 3/23/15 | 4/30/15 |

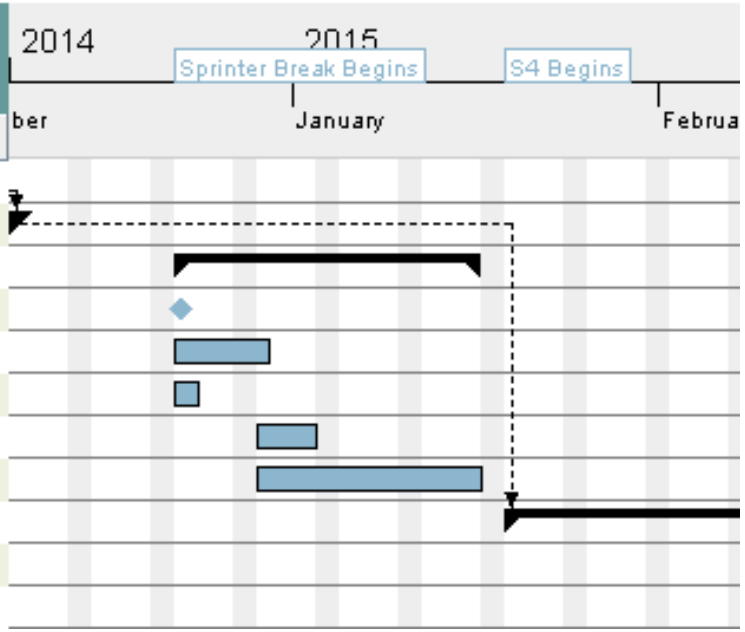
# Sprint 3

- Tasks Completed:
  - GPS Tracking publishing through common protocol
  - SLAM analysis
  - UAV Simulation in Gazebo
  - Point cloud generated with Microsoft Kinect (OpenNI Freenect)

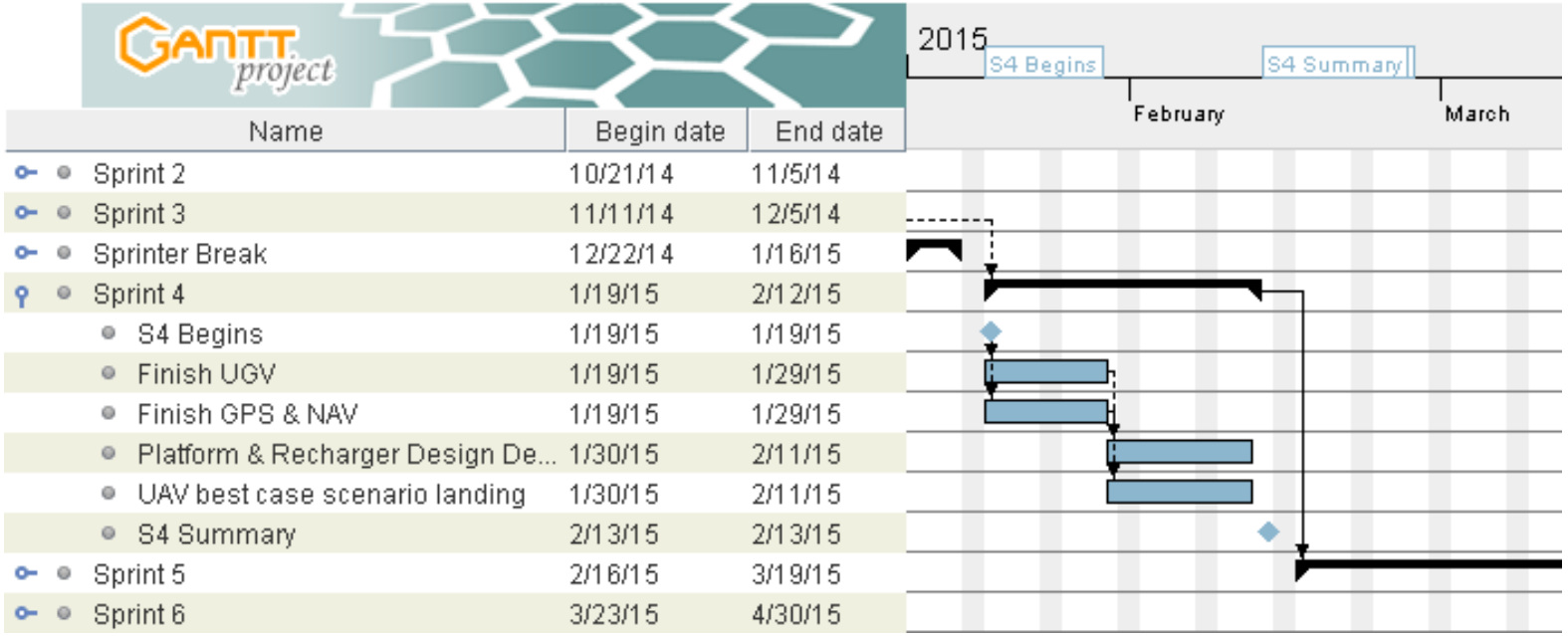
- Tasks Pushed Back:
  - Hardware Stress Testing
  - OpenCV Homography for UAV Landing
  - UGV
    - Design complete, still need to build
  - UAV still in disrepair

# Sprinter Break

| Name | Begin date | End date |
|---|---|---|
| Sprint 2 | 10/21/14 | 11/5/14 |
| Sprint 3 | 11/11/14 | 12/5/14 |
| Sprinter Break | 12/22/14 | 1/16/15 |
| Sprinter Break Begins | 12/22/14 | 12/22/14 |
| UGV Construction | 12/22/14 | 12/29/14 |
| OpenCV Homography | 12/22/14 | 12/23/14 |
| ROS Node Architecture Setup | 12/29/14 | 1/2/15 |
| Testing Environment Setup | 12/29/14 | 1/16/15 |
| Sprint 4 | 1/19/15 | 2/12/15 |
| Sprint 5 | 2/16/15 | 3/19/15 |
| Sprint 6 | 3/23/15 | 4/30/15 |



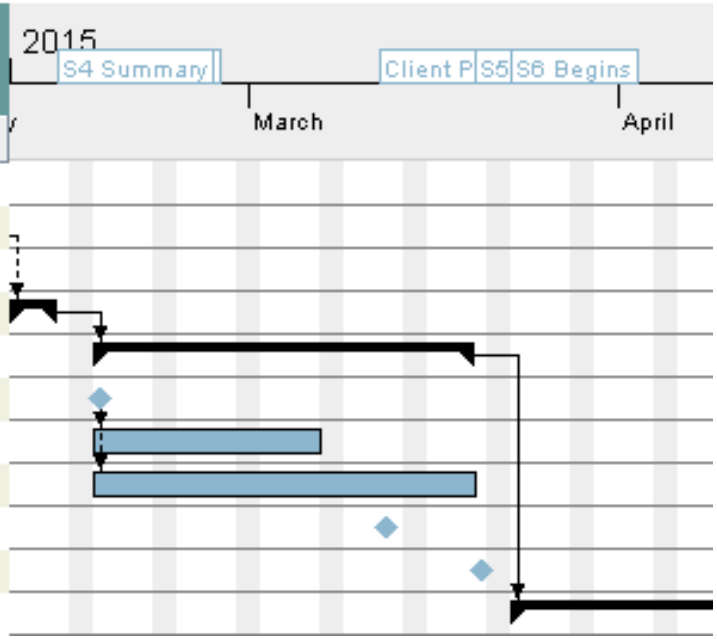
# Sprint 4



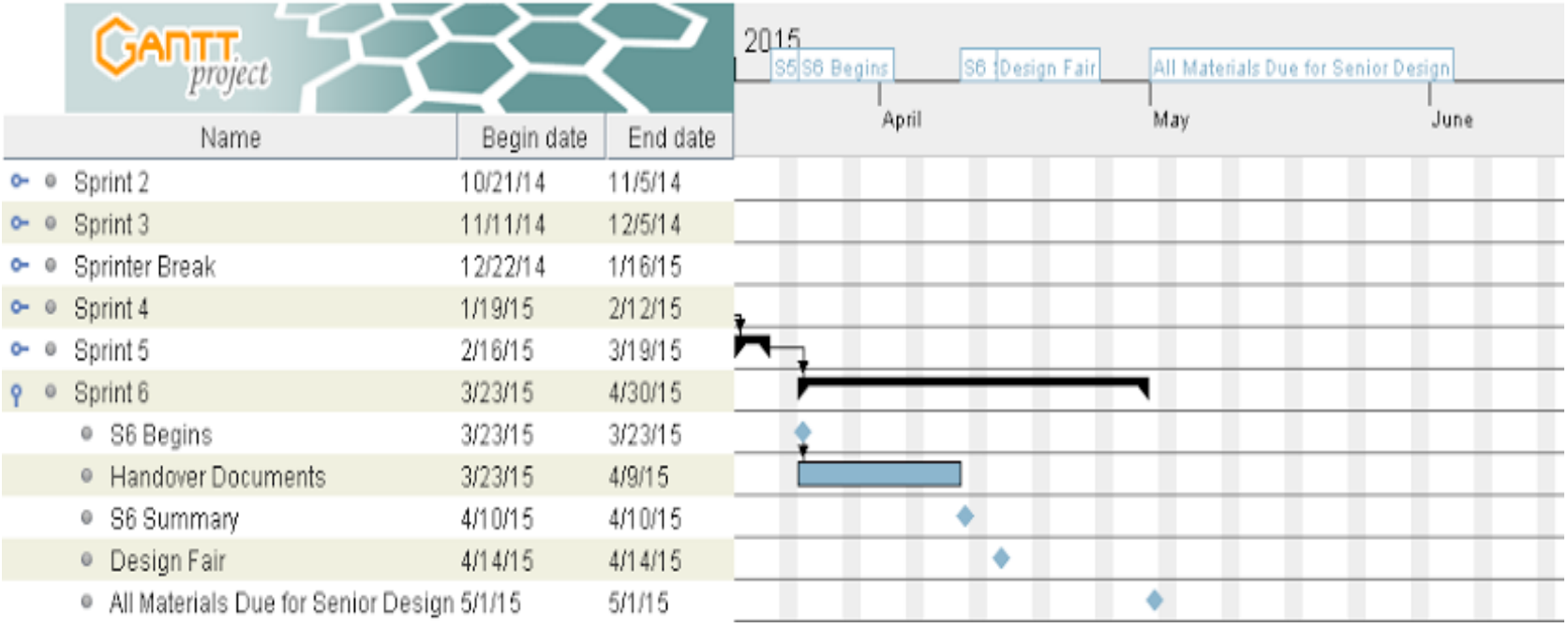

| Name | Begin date | End date |
|---|---|---|
| Sprint 2 | 10/21/14 | 11/5/14 |
| Sprint 3 | 11/11/14 | 12/5/14 |
| Sprinter Break | 12/22/14 | 1/16/15 |
| Sprint 4 | 1/19/15 | 2/12/15 |
| S4 Begins | 1/19/15 | 1/19/15 |
| Finish UGV | 1/19/15 | 1/29/15 |
| Finish GPS & NAV | 1/19/15 | 1/29/15 |
| Platform & Recharger Design De... | 1/30/15 | 2/11/15 |
| UAV best case scenario landing | 1/30/15 | 2/11/15 |
| S4 Summary | 2/13/15 | 2/13/15 |
| Sprint 5 | 2/16/15 | 3/19/15 |
| Sprint 6 | 3/23/15 | 4/30/15 |

# Sprint 5

| Name | Begin date | End date |
|---|---|---|
| Sprint 2 | 10/21/14 | 11/5/14 |
| Sprinter | 11/11/14 | 12/5/14 |
| Sprinter Break | 12/22/14 | 1/16/15 |
| Sprint 4 | 1/19/15 | 2/12/15 |
| Sprint 5 | 2/16/15 | 3/19/15 |
| S5 Begins | 2/16/15 | 2/16/15 |
| Platform Implementation | 2/16/15 | 3/6/15 |
| Intensive Field Testings | 2/16/15 | 3/19/15 |
| Client Presentation | 3/12/15 | 3/12/15 |
| S5 Summary | 3/20/15 | 3/20/15 |
| Sprint 6 | 3/23/15 | 4/30/15 |



# Sprint 6




| Name | Begin date | End date |
|---|---|---|
| Sprint 2 | 10/21/14 | 11/5/14 |
| Sprint 3 | 11/11/14 | 12/5/14 |
| Sprinter Break | 12/22/14 | 1/16/15 |
| Sprint 4 | 1/19/15 | 2/12/15 |
| Sprint 5 | 2/16/15 | 3/19/15 |
| Sprint 6 | 3/23/15 | 4/30/15 |
| S6 Begins | 3/23/15 | 3/23/15 |
| Handover Documents | 3/23/15 | 4/9/15 |
| S6 Summary | 4/10/15 | 4/10/15 |
| Design Fair | 4/14/15 | 4/14/15 |
| All Materials Due for Senior Design | 5/1/15 | 5/1/15 |

# OpenCV

RECAP, HOMOGRAPHY, ANALYSIS

# OpenCV

Plan:

- Distance Algorithm:
  - Captures distance but not angle/skew
  - Only works in limited distance.
- Homography:
  - Change in angle could be extracted from Homography Matrix
- However, we're using SLAM…
  - SLAM for mapping, would also give us info of the landing pad
- Redundancy
  - OpenCV + SLAM would give us two checks
  - ODROIDs
    - Computing analysis after stress testing

# SLAM

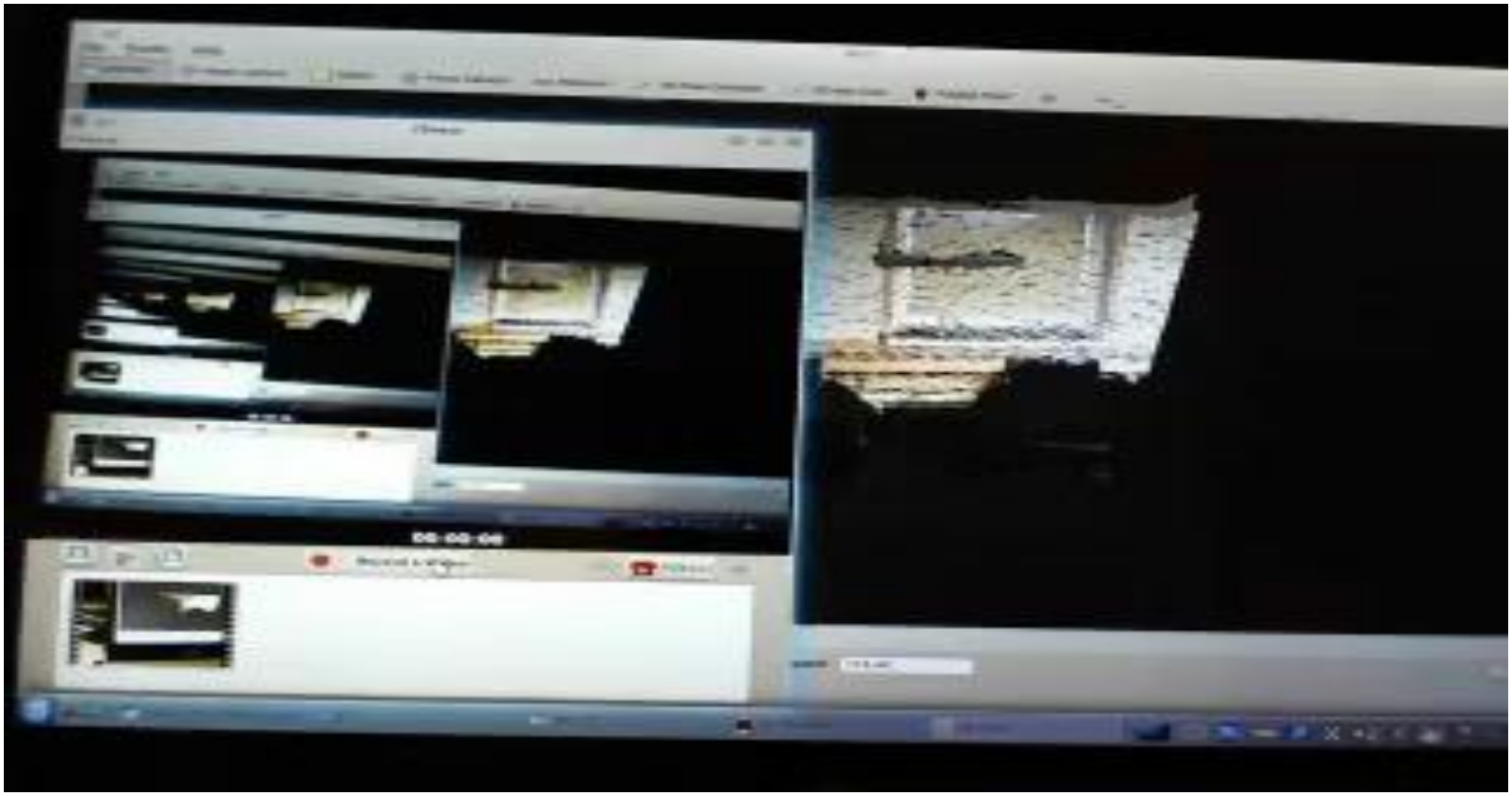
# SLAM

- With some short fallings of OpenCV by itself, we've found coupling SLAM technology with our ROS nodes, which integrate Hector SLAM with existing UAV functionality.

- Different Types
  - Various SLAM methodologies were considered for the UGV including Hector-SLAM, LSD-SLAM, etc..

- For additional obstacle detection of the UGV, point cloud nodes have been assembled leveraging both Freenect and OpenNi. This combination allows for both Asus Xtion and Microsoft Kinect connectivity.

# Point Cloud of McL108



# GPS

# GPS

- Common Ground
  - As the UAV is using a 3dr ublox GPS and the UGV's GPS is still undecided, it has been our decision to try and find a way to incorporate an interpretive node that will take many forms of GPS pose and marker topics and relay them to a common topic.
  - To test and validate these nodes' data, the Android_Sensors node was added to the build and was then modified to relay data out as specific data using topic names like "GPS_data". This custom topic pushes data as received or every 2 seconds.
  - Sample Data
    - header: seq: 25087 stamp: secs: 1417577925 nsecs: 862000000 frame_id: /gpsstatus: status: 0 service: 1 latitude: 44.08800569 longitude: -103.22106451 altitude: 955.5 position_covariance: [256.0, 0.0, 0.0, 0.0, 256.0, 0.0, 0.0, 0.0, 256.0] position_covariance_type: 1

# ROS

ROBOTICS IMPLEMENTATION AND SIMULATION

# ROS and Gazebo

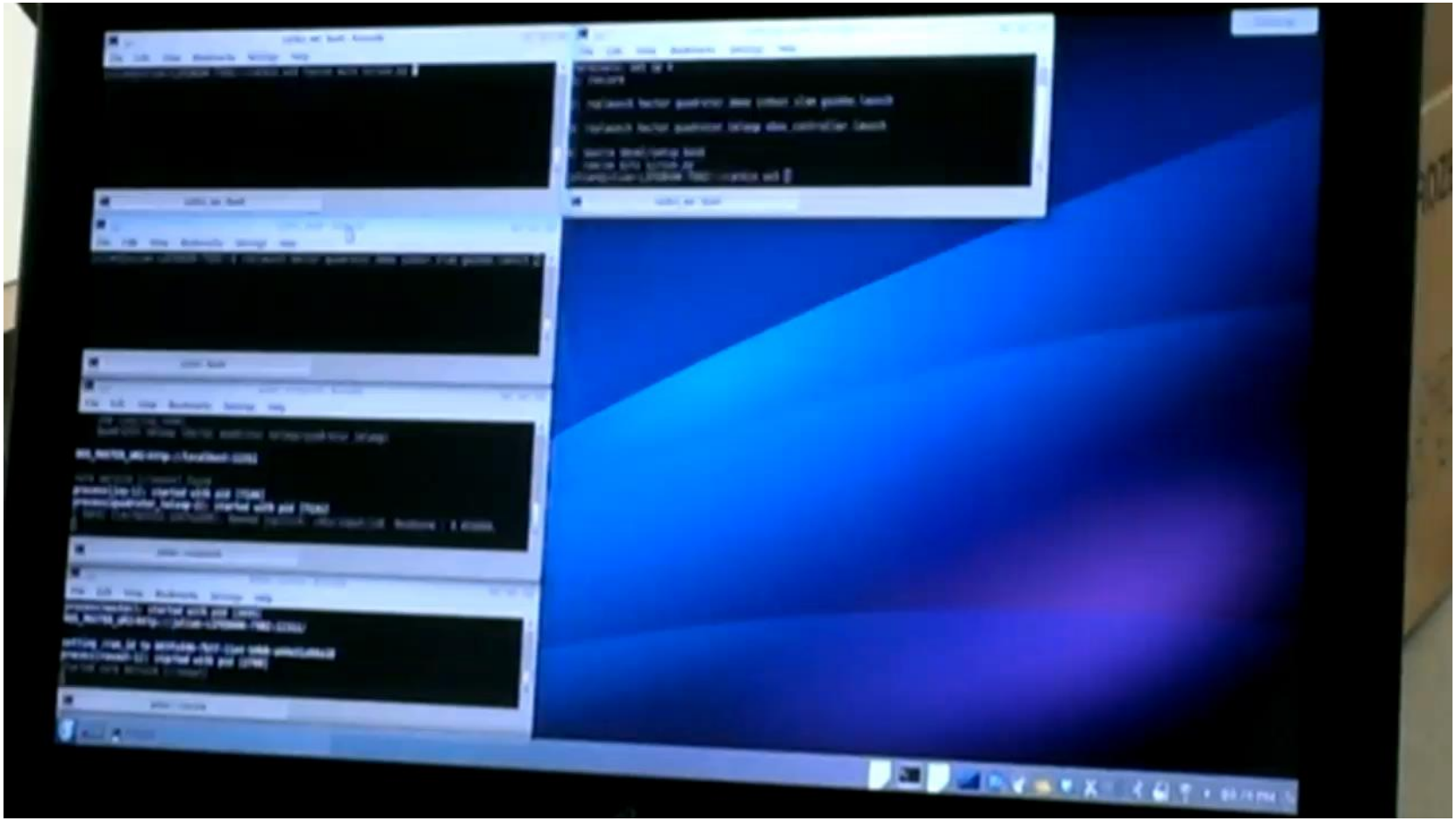
ROS + Gazebo for simulating UAV + UGV

- Gazebosim.org
- 3D graphics
  - using OGRE (OpenGL 3D game dev)
- Generate sensor data
  - (cameras, gps etc)
- Works with ROS


Simulation helps fine-tune software while keeping costs down

# ROS + Gazebo Demo

# ROS + Gazebo Demo (video)

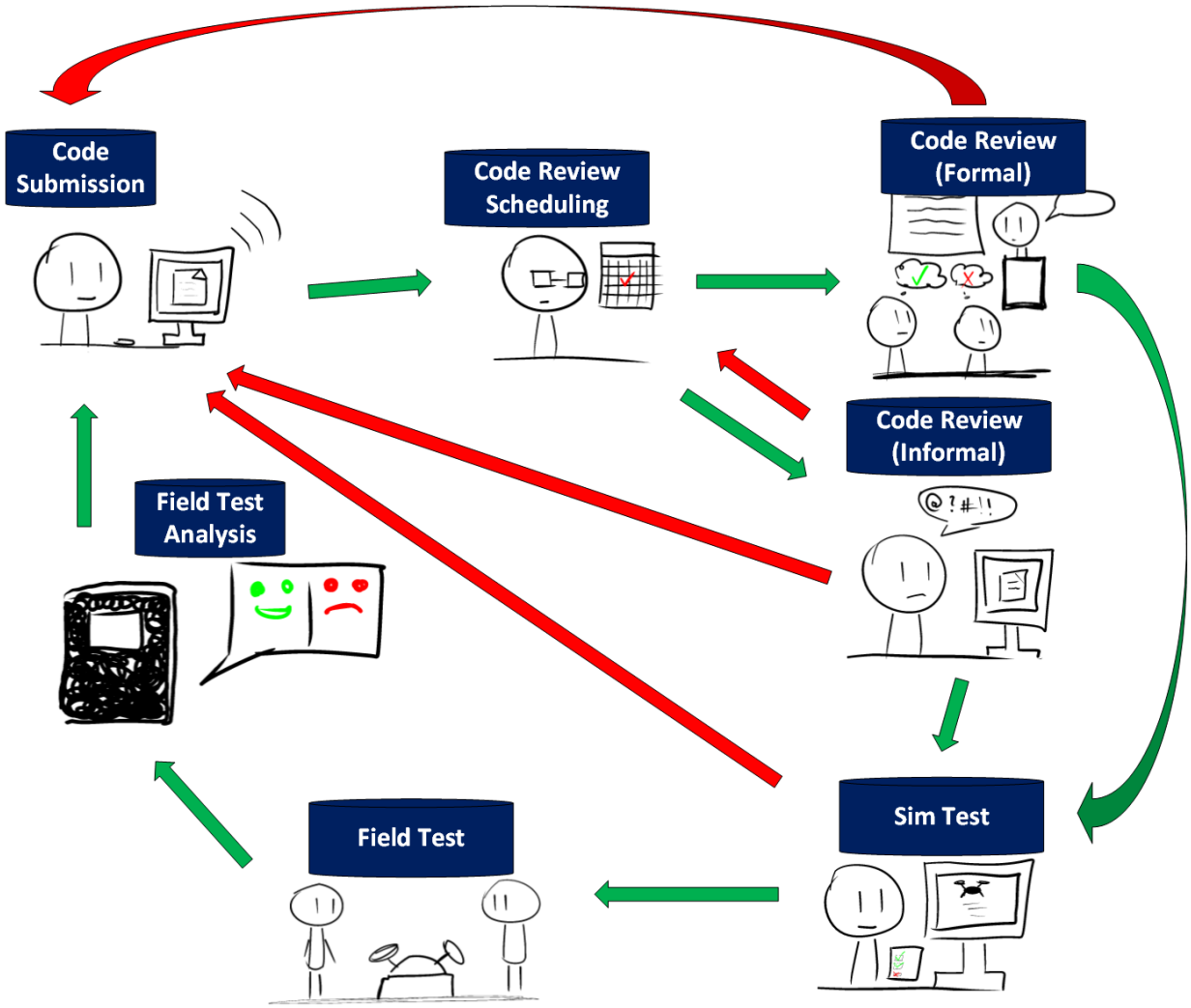


# Testing

# Testing Software

- Project has heavy focus on Research first 3 sprints
  - We had set up in our Sprint timeline that 1-3 were research heavy, with 4-6 being dev / testing
  - As a result, we haven't done a lot with testing
  - Winter break focuses on transitioning from research to development
- Plan for testing UAV
  - Testing Plan on next slide
  - Before Sprint 4:
    - Set up ROS node architecture
    - Complete Unit Test scripts for environment
    - Finish Gazebo environment for robust simulation testing

# Testing Diagram

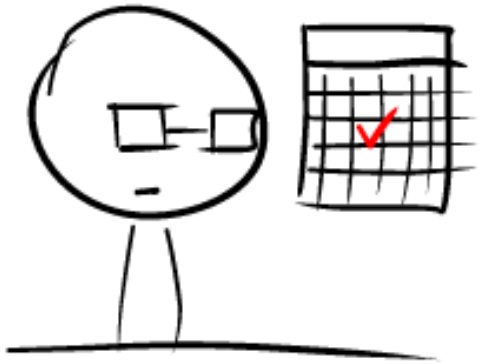


# Code Submission

Development Team submits code to the repo

- Documented changes
- Scrum Master notified by email
- Changes based on Sprint Backlog requirements or Field Test Analysis

# Code Review Scheduling



Done by the Scrum Master

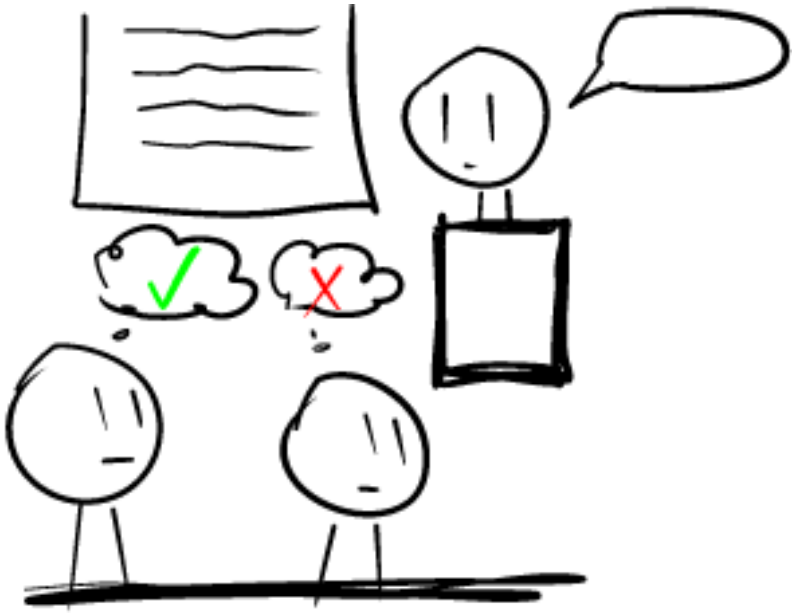
- Determines Formal or Informal review based on content
  - At least 2 Formal Reviews per Sprint
- For Informal Reviews, team member is picked based on availability
  - Scrum Master can perform Informal Review as well

# Code Review (Informal)


Code Reviewed by Team member chosen by Scrum Master

- Approves code for Sim Testing
- Can promote to a Formal Code Review (scheduled by Scrum Master)
- In rare instances, can outright reject code submission

# Code Review (Formal)



Code Review performed by entire team

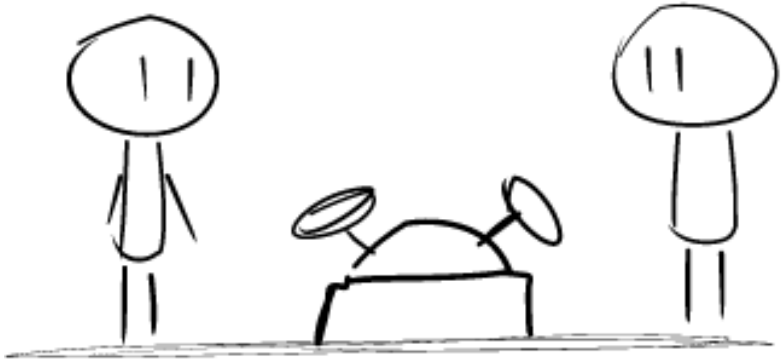
- Less Frequent than Informal Reviews
- Author(s) of code revision detail changes to software to entire team
- Surbeck Board Rooms, or Business Dev. Center
- If the ENTIRE team does not agree on changes, code must be resubmitted.

# Sim Test

Team Member tests new build on the Gazebo Simulator

- Newest build does not get added to the physical hardware without passing a Sim Test
- All Code Reviews (Formal and Informal) must be followed up with a Sim Test
  - Sim Test can be performed at Formal Code Review
- Depending on Scheduling issues, Sim Tests will not always be followed by Field Tests

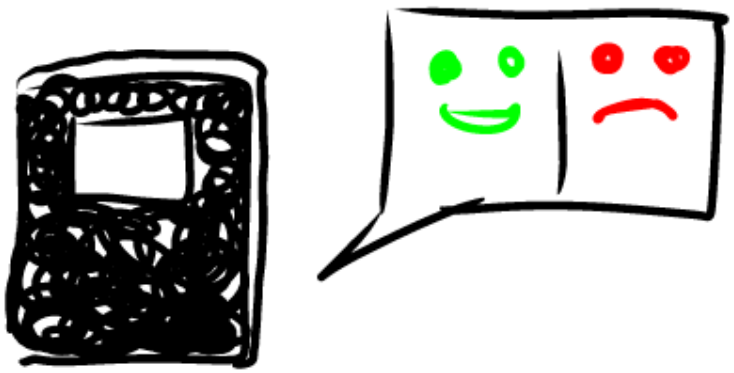
# Field Test



Minimum of two Members of the team test new software build on UAV / UGV

- Keep Lab Notebooks, document results, report to Scrum Master if they are not present
- Code that has not been approved by a Sim Test absolutely can not be used on vehicles in a Field Test

# Field Test Analysis



Team reports Field Test

- What went well, what did not
- Problems in software are added to Sprint Backlog
- Field Notes are integrated into the Design Document
- Field Test Team and Code Dev Team collaborate to determine changes for next build

# Deliverables

# Deliverables

What we got:

- Testing Environment - Sirius.py
  - Node “template” for navigation
  - Verifies  scripting works in sim
- Node structure for Nav, SVO, Ptam, SLAM & PointClouds
  - Architecture design will bridge these together
- UAV Design
- UGV Design
- Document
  - Still in progress

# Conclusion