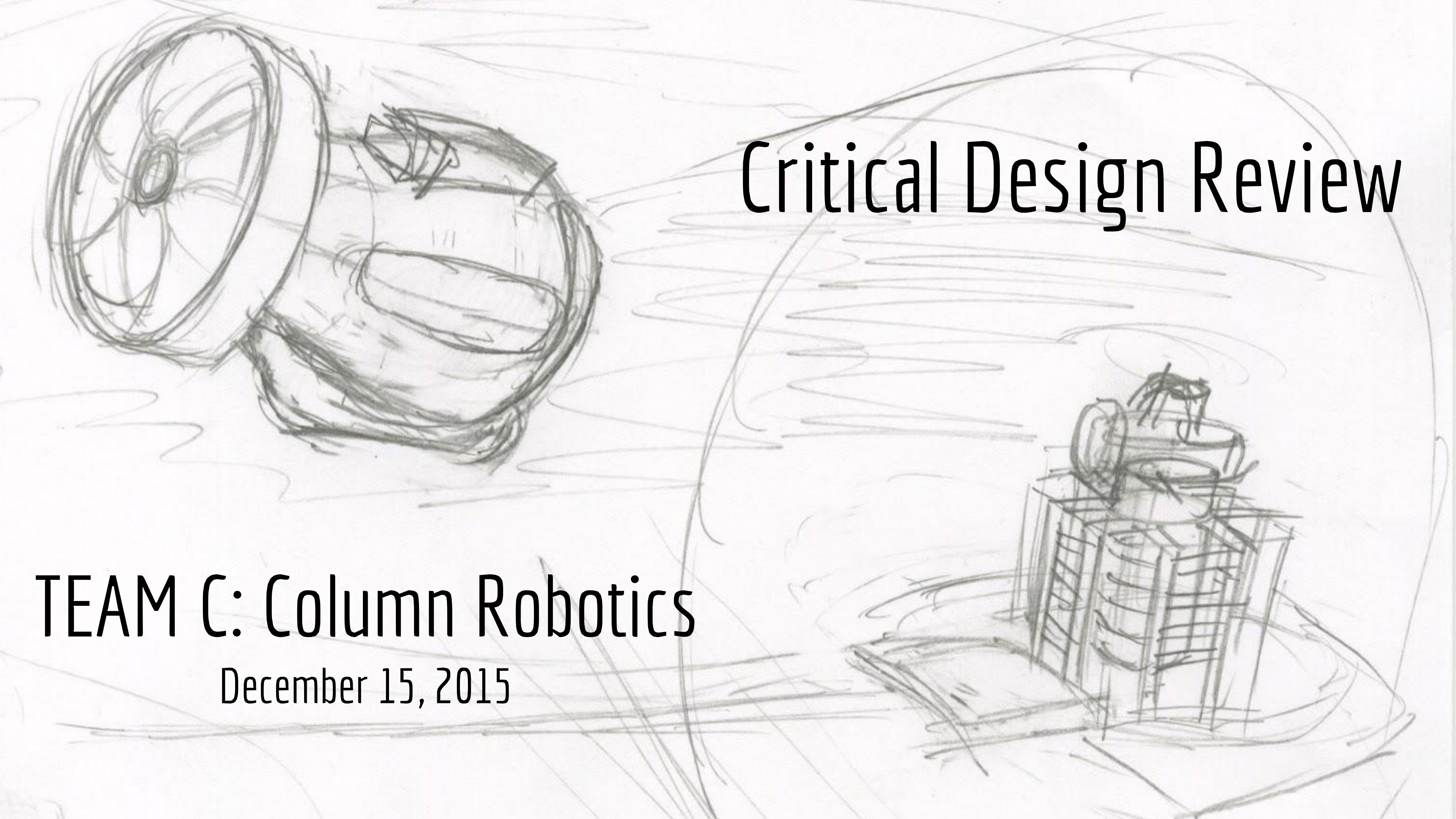
Critical Design Review



TEAM C: Column Robotics

December 15, 2015

TEAM C: Column Robotics

Job Bedford

Cole Gulino

Erik Sjoberg

Rohan Thakker

Sponsor: Dr. Hagen Schempf

Structure of Presentation

1. Project Description
2. Use Case
3. System-Level Requirements
4. Functional & Cyberphysical Architecture
5. Current System Status

   Requirements | System / Subsystem Descriptions | Modeling / Analysis / Testing

   FVE Performance Evaluation | Video | Strong / Weak Points
6. Project Management

   WBS Summary | Schedule Status | Test Plan | Budget Status | Risk Management
7. Conclusions

TEAM C: Column Robotics

1. **Project Description**
2. Use Case
3. System-Level Requirements
4. Functional & Cyberphysical Architecture
5. Current System Status

   Requirements | System / Subsystem Descriptions | Modeling / Analysis / Testing

   FVE Performance Evaluation | Video | Strong / Weak Points
6. Project Management

   WBS Summary | Schedule Status | Test Plan | Budget Status | Risk Management
7. Conclusions

Project Description

Autonomous Exploration and Docking

Failure of undersea oil and gas infrastructure has resulted in billions of dollars of damages.

Current maintenance of these wellheads often requires a specialized ship and manual ROV crew, costings hundreds of thousands of dollars per intervention.

We propose to demonstrate **a terrestrial analogue** to an underwater vehicle capable of autonomously searching for, identifying, and docking with underseas wellhead.

http://persistentautonomy.com/

http://3drobotics.com

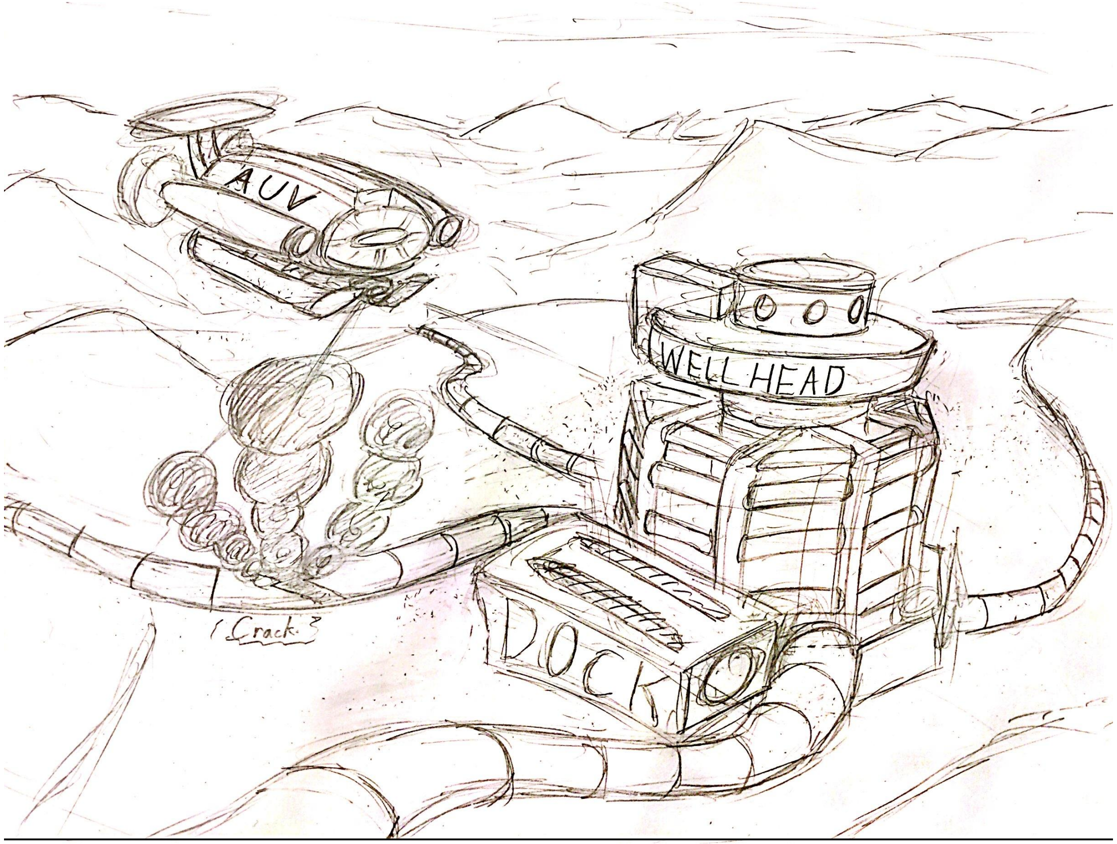
TEAM C: Column Robotics

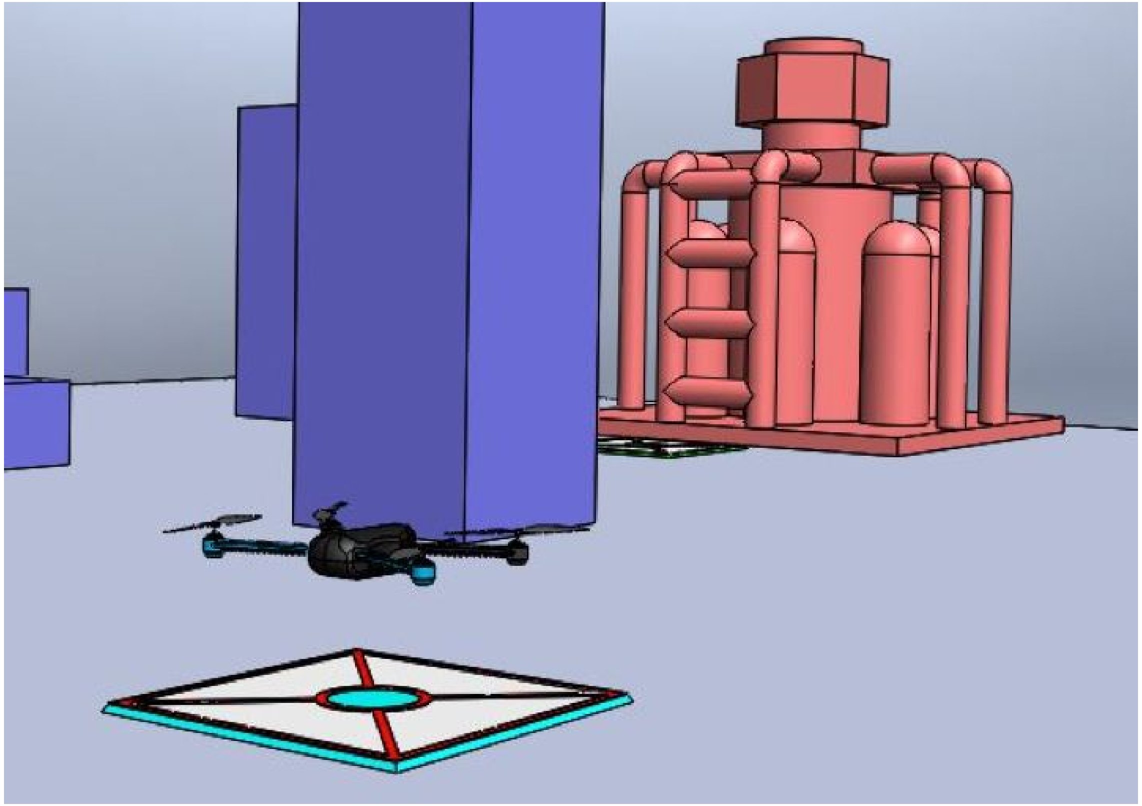
1. Project Description
2. **Use Case**
3. System-Level Requirements
4. Functional & Cyberphysical Architecture
5. Current System Status

   Requirements | System / Subsystem Descriptions | Modeling / Analysis / Testing

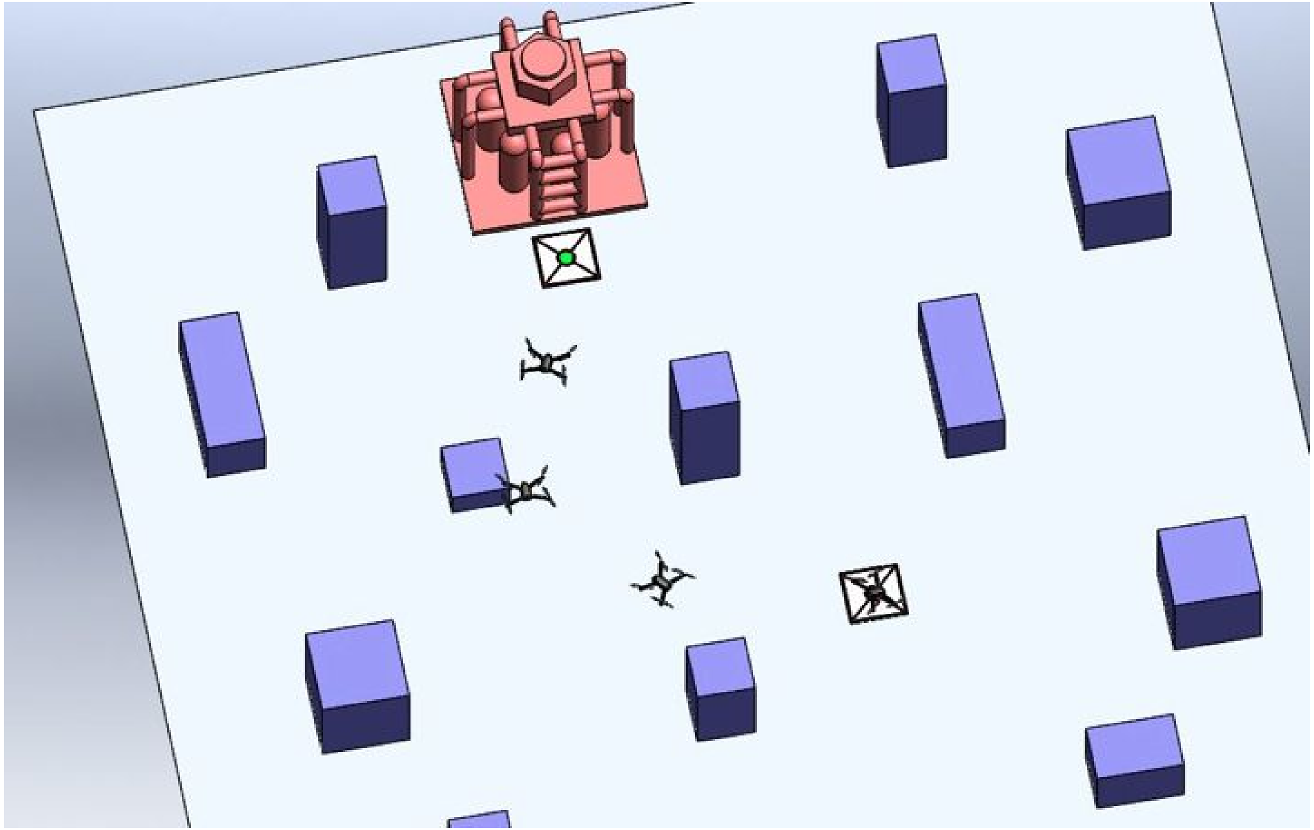
   FVE Performance Evaluation | Video | Strong / Weak Points
6. Project Management

   WBS Summary | Schedule Status | Test Plan | Budget Status | Risk Management
7. Conclusions

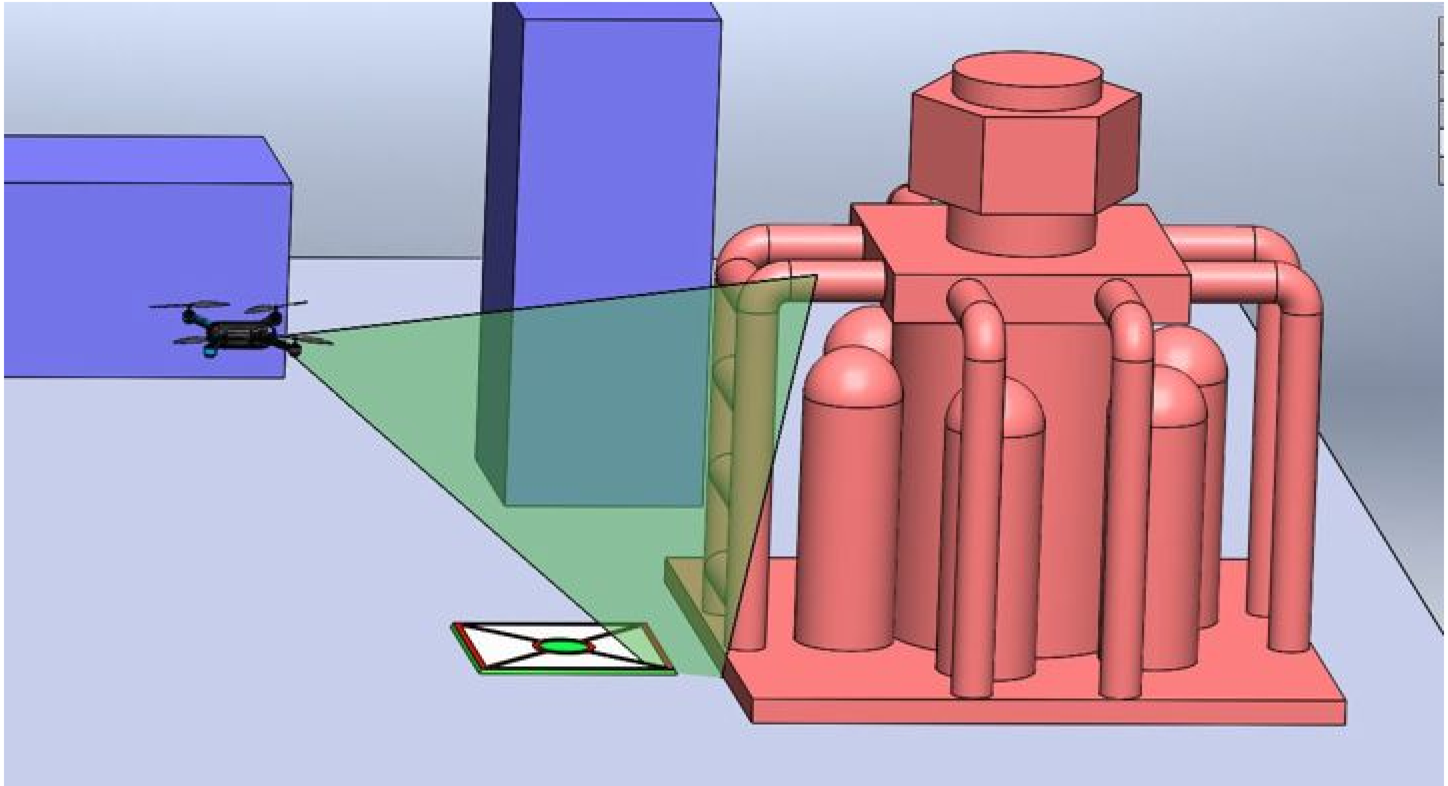

AUV
WELL HEAD
DOCK
Crack



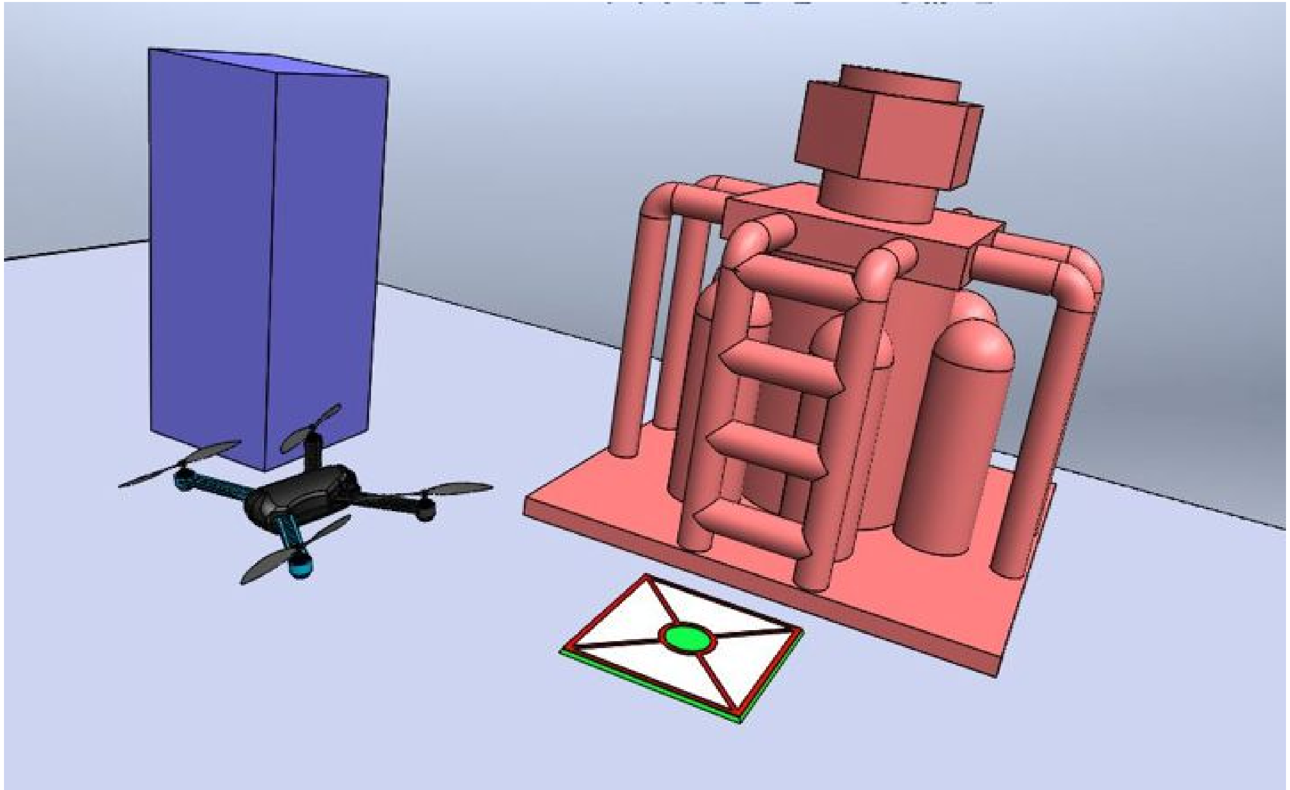
Liftoff from initial Landing Pad



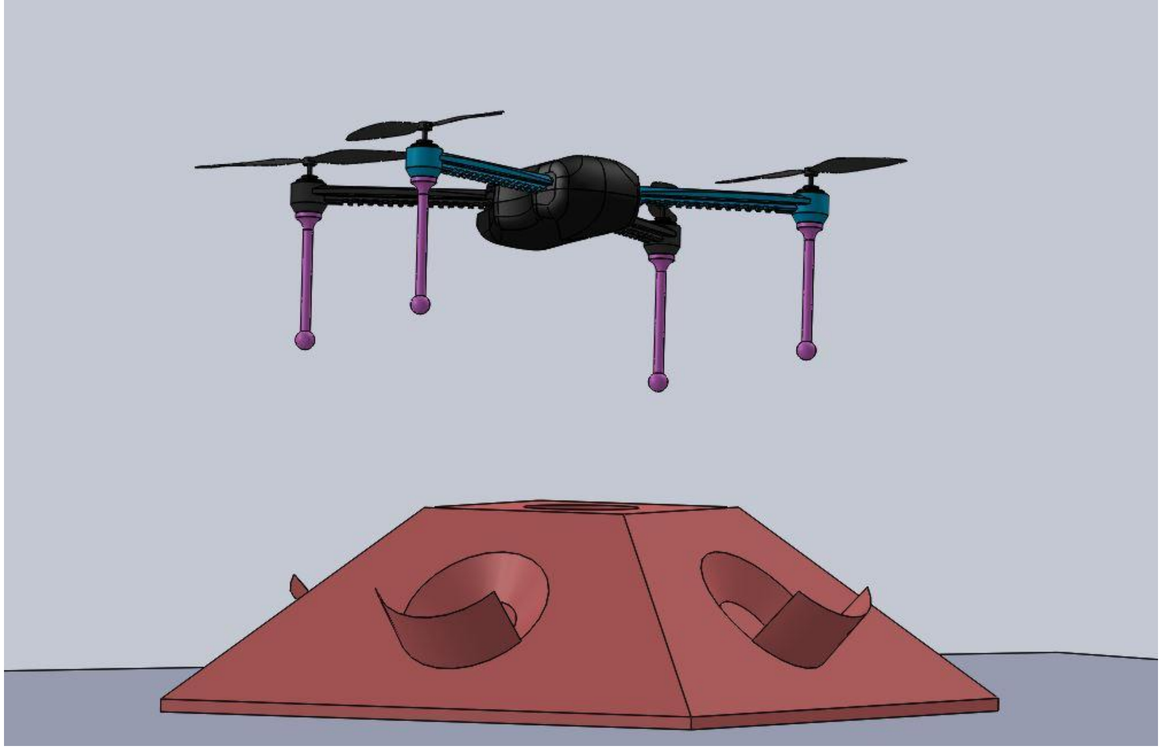
Searching for Wellhead



Identify Wellhead



Orientation before Docking



Docking

TEAM C: Column Robotics

1. Project Description
2. Use Case
3. **System-Level Requirements**
4. Functional & Cyberphysical Architecture
5. Current System Status

   Requirements | System / Subsystem Descriptions | Modeling / Analysis / Testing

   FVE Performance Evaluation | Video | Strong / Weak Points
6. Project Management

   WBS Summary | Schedule Status | Test Plan | Budget Status | Risk Management
7. Conclusions

MANDATORY FUNCTIONAL REQUIREMENTS

MF1. Locate Oil/Gas wellhead infrastructure with known heading in 25 m^2 area

- Change: Area shrunk due to testing constraints

MF2. Autonomously maneuver to wellhead within 1 hour

MF3. Positively ID as correct wellhead with 90% confidence

MF4. Maintain hover position over dock within +/- 1m

- Change: Added performance metric

MF5. Rigidly dock in 5 DOF

MF6. Provide status feedback to user of current state at 0.1 Hz

DESIRED FUNCTIONAL REQUIREMENTS

DF1. Locate oil/gas wellhead infrastructure in low visibility with unknown heading in 25 m^2 area

- Change: Added performance metric

DF2. Positively ID as correct wellhead from visual object recognition with 90% confidence

- Change: Added performance metric

DF3. Align with dock located at known radius but unknown angle from wellhead within +/- 1m

- Change: Added performance metric

~~DF4. Rigidly dock in 6 DOF with electrical connection~~

- Change: Removed

MANDATORY NON-FUNCTIONAL REQUIREMENTS

MNF1. Operable with simple graphical user interface

MNF2. Provides emergency stop for system with less than 1 second lag

- Change: Added performance metrics

MNF3. Operable by a single person

DESIRED NON-FUNCTIONAL REQUIREMENTS

DNF1. Reduce operator cost by at least one-half

- Change: Added performance metric

DNF2. Simulate low-visibility: Unable to get visual feed beyond 3m from camera/quadrotor

- Change: Made more concrete, added performance metric

TEAM C: Column Robotics

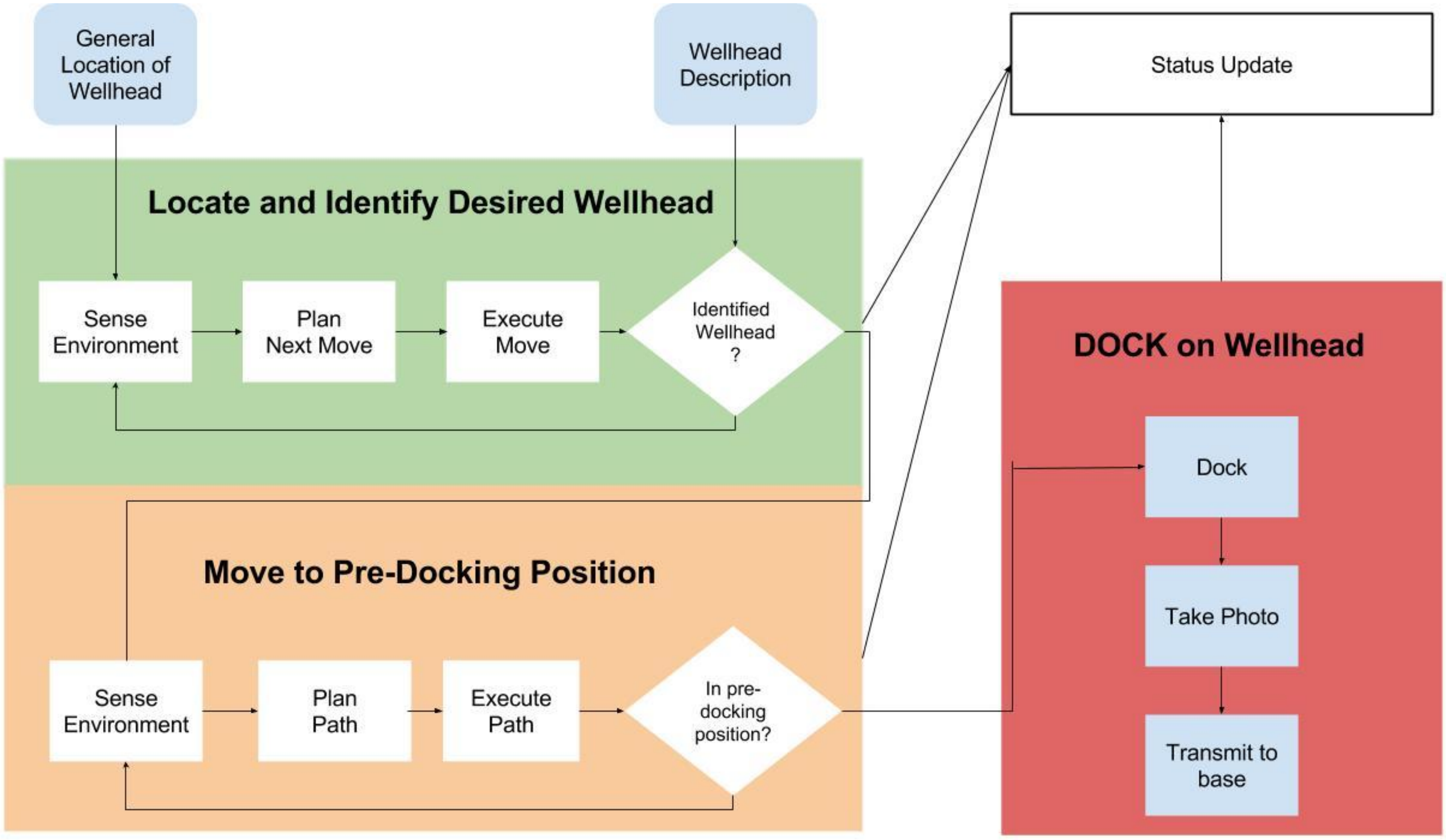
1. Project Description
2. Use Case
3. System-Level Requirements
4. **Functional & Cyberphysical Architecture**
5. Current System Status

   Requirements | System / Subsystem Descriptions | Modeling / Analysis / Testing

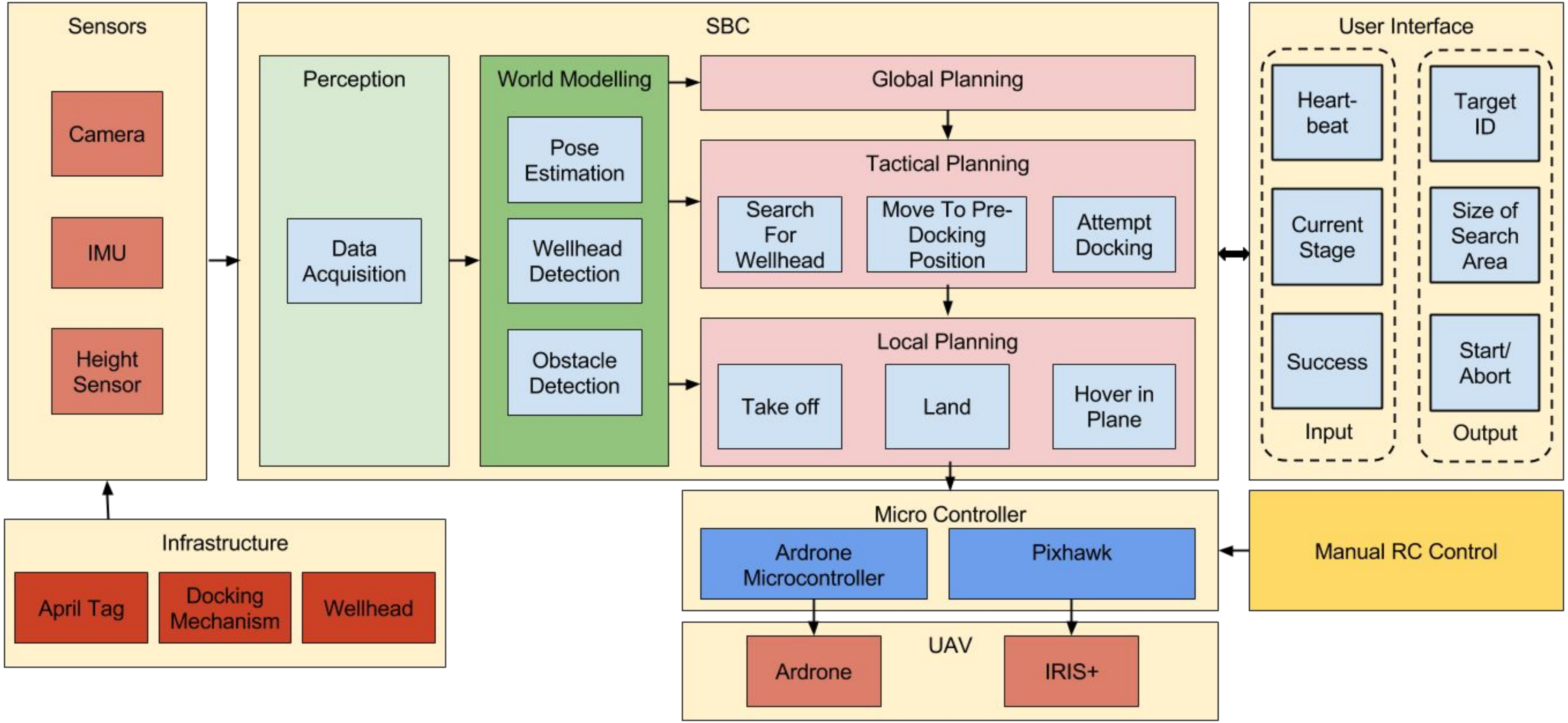
   FVE Performance Evaluation | Video | Strong / Weak Points
6. Project Management

   WBS Summary | Schedule Status | Test Plan | Budget Status | Risk Management
7. Conclusions

FUNCTIONAL ARCHITECTURE



CYBERPHYSICAL ARCHITECTURE



TEAM C: Column Robotics

1. Project Description
2. Use Case
3. System-Level Requirements
4. Functional & Cyberphysical Architecture
5. **Current System Status**

   Requirements | System / Subsystem Descriptions | Modeling / Analysis / Testing

   FVE Performance Evaluation | Video | Strong / Weak Points
6. Project Management

   WBS Summary | Schedule Status | Test Plan | Budget Status | Risk Management
7. Conclusions

TARGETED REQUIREMENTS

MF1. Locate Oil/Gas wellhead infrastructure with known heading in 25 m^2 area

MF2. Autonomously maneuver to wellhead within 1 hour

MF3. Positively ID as correct wellhead with 90% confidence

MF4. Maintain hover position over dock within +/- 1m

MF5. Rigidly dock in 5 DOF

MF6. Provide status feedback to user of current state at 0.1 Hz

MNF1. Operable with simple graphical user interface

MNF2. Provides emergency stop for system with less than 1 second lag

MNF3. Operable by a single person

Completed | In-Progress | Not Started

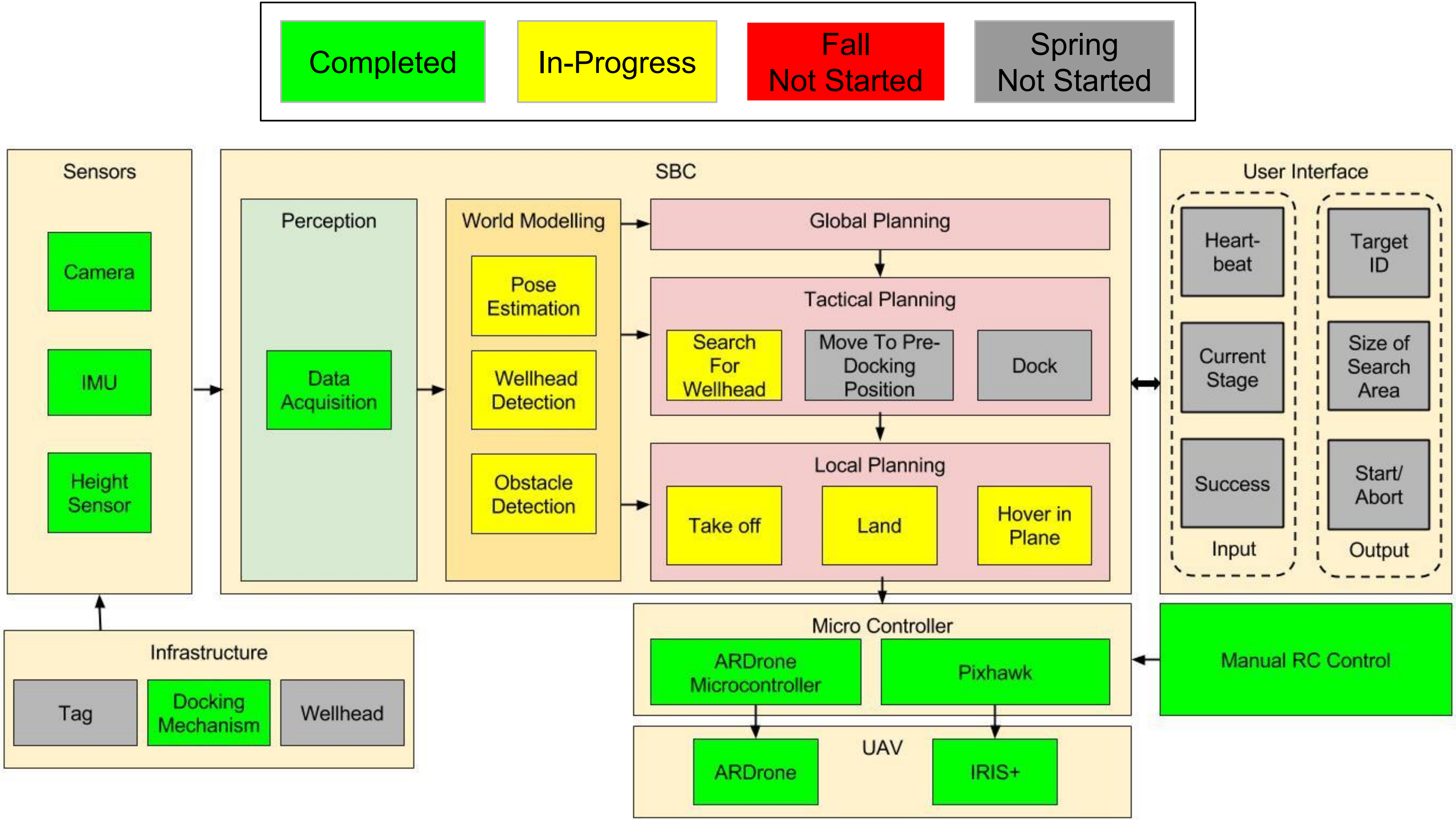
TEAM C: Column Robotics

1. Project Description
2. Use Case
3. System-Level Requirements
4. Functional & Cyberphysical Architecture
5. **Current System Status**

   Requirements | **System / Subsystem Descriptions** | Modeling / Analysis / Testing

   FVE Performance Evaluation | Video | Strong / Weak Points
6. Project Management

   WBS Summary | Schedule Status | Test Plan | Budget Status | Risk Management
7. Conclusions



Completed

In-Progress

Fall Not Started

Spring Not Started

Sensors

Camera

IMU

Height Sensor

SBC

Perception

Data Acquisition

World Modelling

Pose Estimation

Wellhead Detection

Obstacle Detection

Global Planning

Tactical Planning

Search For Wellhead

Move To Pre-Docking Position

Dock

Local Planning

Take off

Land

Hover in Plane

User Interface

Heart-beat

Current Stage

Success

Input

Target ID

Size of Search Area

Start/ Abort

Output

Infrastructure

Tag

Docking Mechanism

Wellhead

Micro Controller

ARDrone Microcontroller

Pixhawk

Manual RC Control

UAV

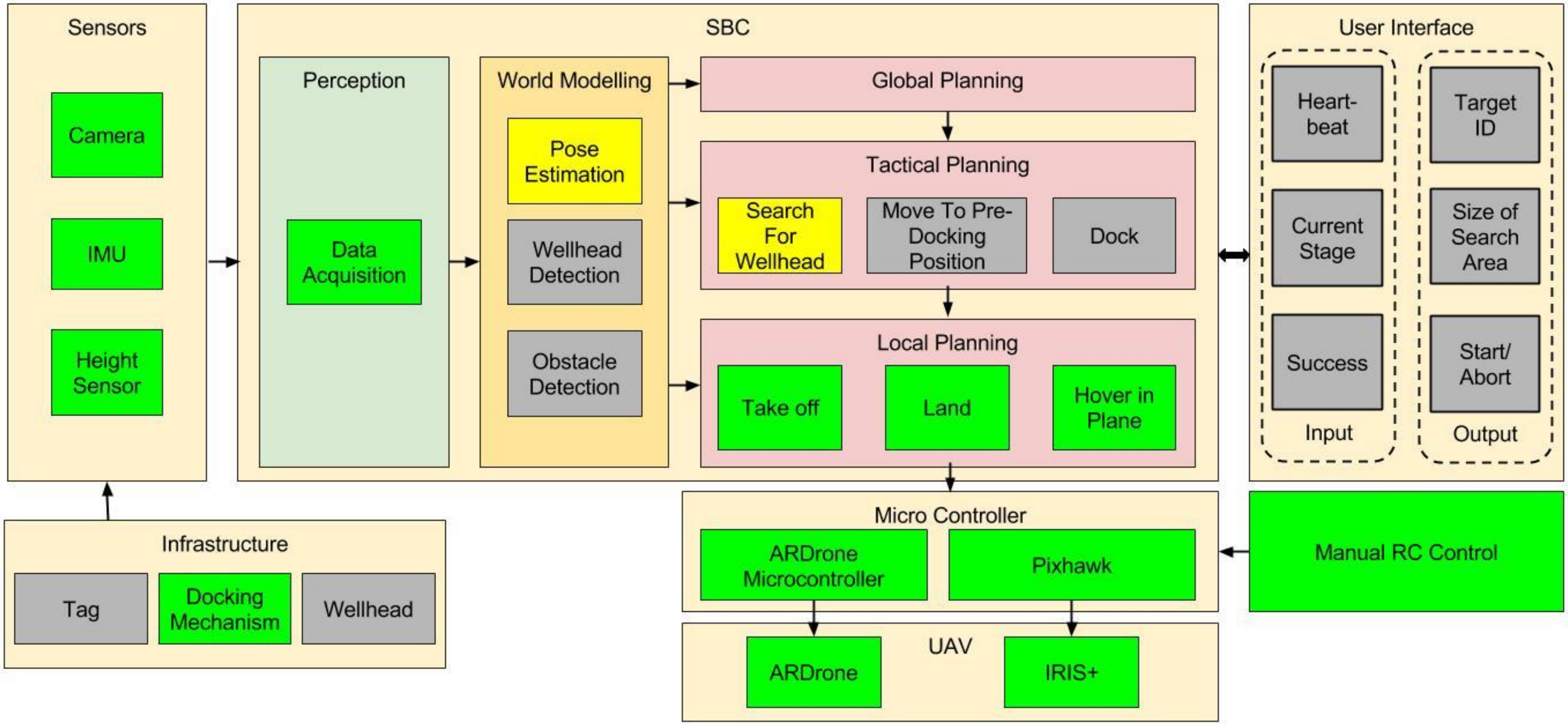
ARDrone

IRIS+

Completed

In-Progress

Not Started



Sensors
- Camera
- IMU
- Height Sensor

SBC
- Perception
  - Data Acquisition
- World Modelling
  - Pose Estimation
  - Wellhead Detection
  - Obstacle Detection
- Global Planning
- Tactical Planning
  - Search For Wellhead
  - Move To Pre-Docking Position
  - Dock
- Local Planning
  - Take off
  - Land
  - Hover in Plane

User Interface
- Input
  - Heart-beat
  - Current Stage
  - Success
- Output
  - Target ID
  - Size of Search Area
  - Start/ Abort

Infrastructure
- Tag
- Docking Mechanism
- Wellhead

Micro Controller
- ARDrone Microcontroller
- Pixhawk

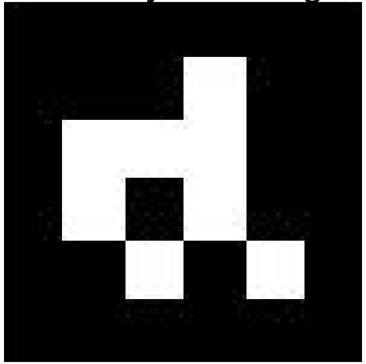
UAV
- ARDrone
- IRIS+

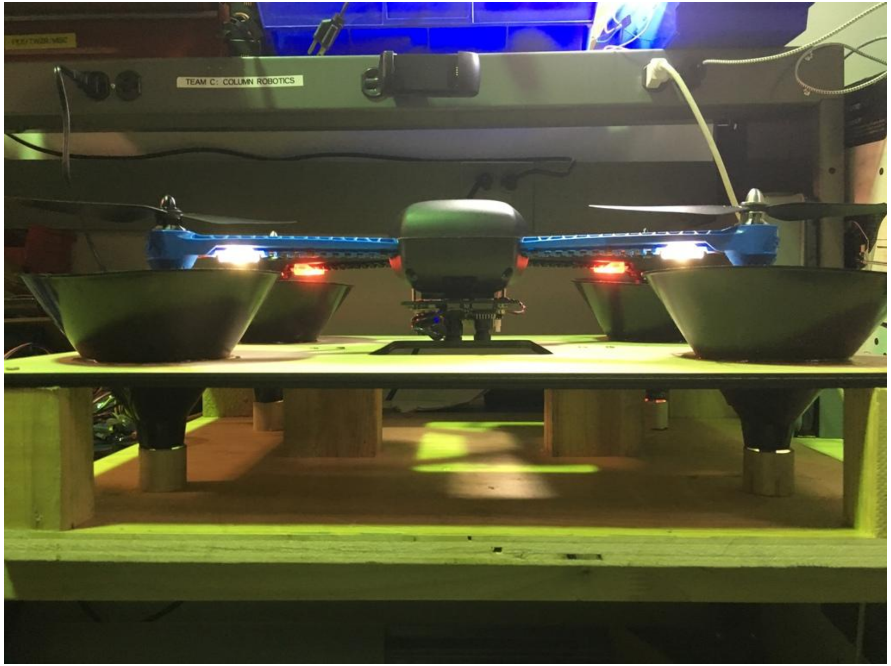
Manual RC Control

Infrastructure

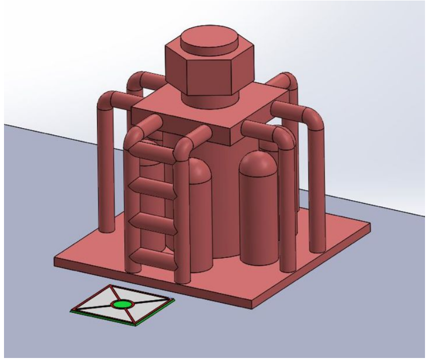
- Tag
- Docking Mechanism
- Wellhead

APRIL Tag
University of Michigan





Wellhead Model

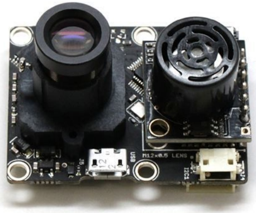


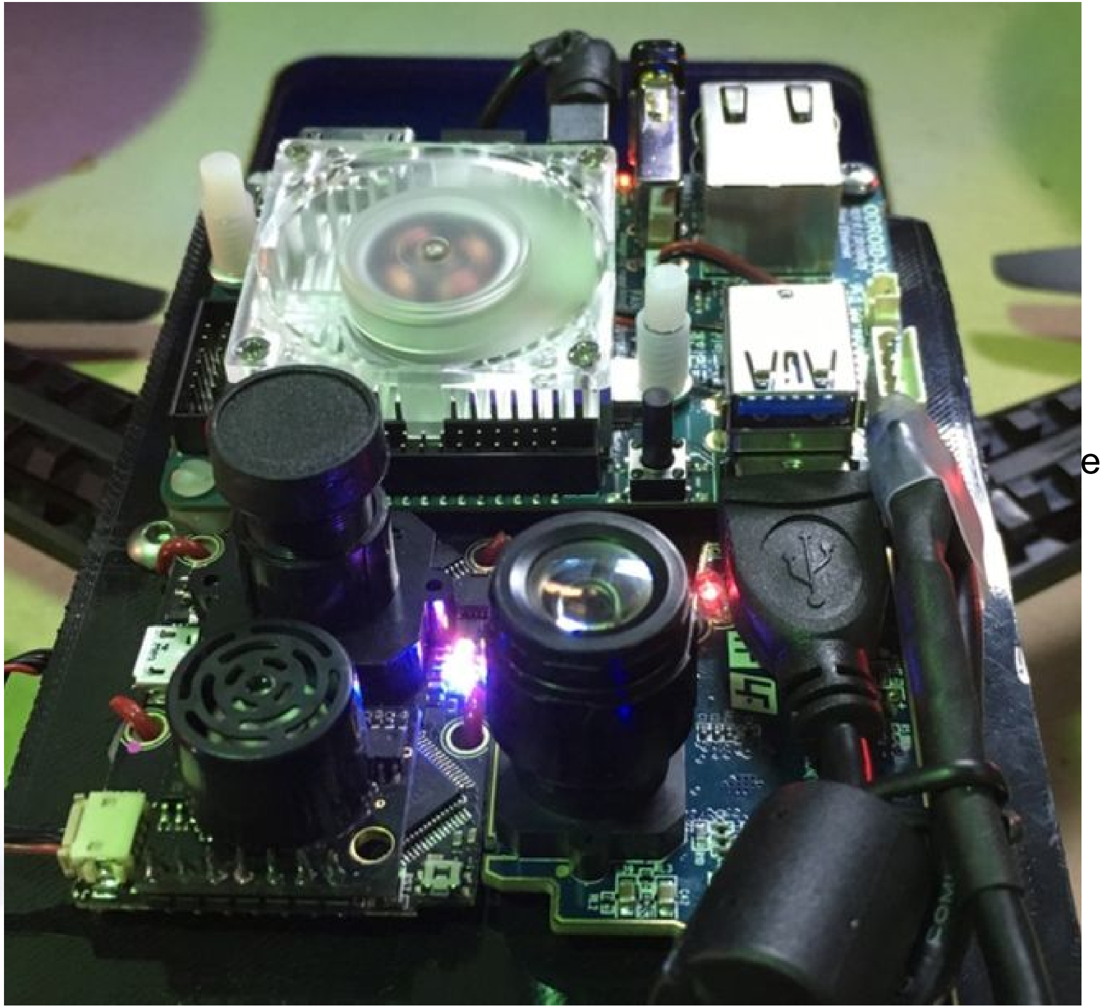
Sensors

Camera

IMU

Height Sensor





er

World Modelling

Pose Estimation

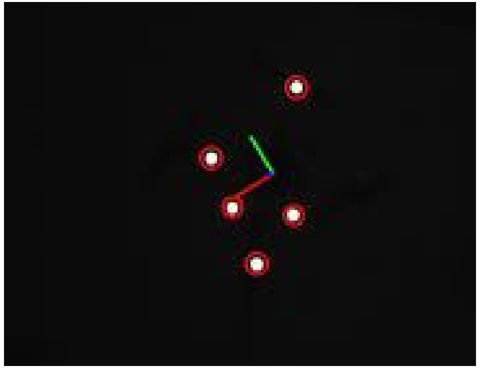
Wellhead Detection

Obstacle Detection

Lucas-Kanade Optical Flow

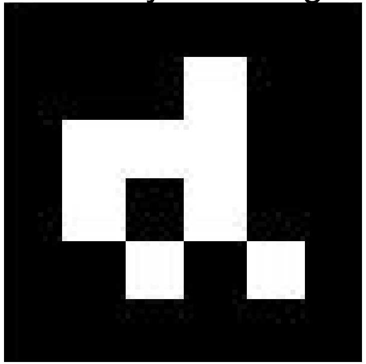
www.hizook.com

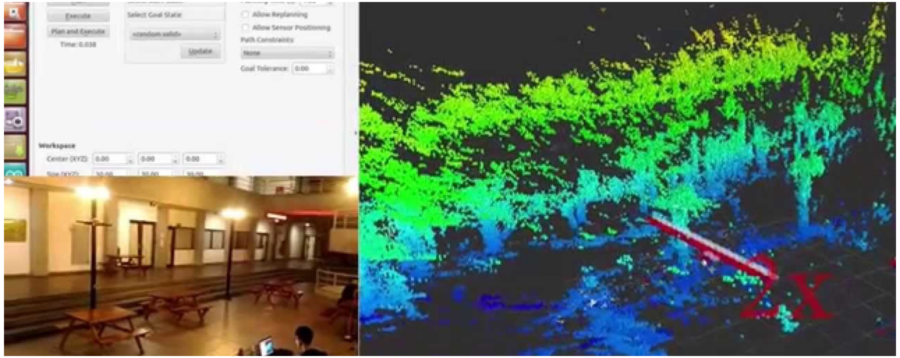
IR LED Based



www.rpg.ifi.uzh.ch

APRIL Tag
University of Michigan





LSD-SLAM: Large-Scale.
Direct Monocular SLAM.
Technical University Munich

source: https://vision.in.tum.de

Global Planning

Tactical Planning

- Search For Wellhead
- Move To Pre-Docking Position
- Attempt Docking

Local Planning

- Take off
- Land
- Hover in Plane

Micro Controller

Ardrone Microcontroller

Pixhawk

Manual RC Control

UAV

Ardrone

IRIS+

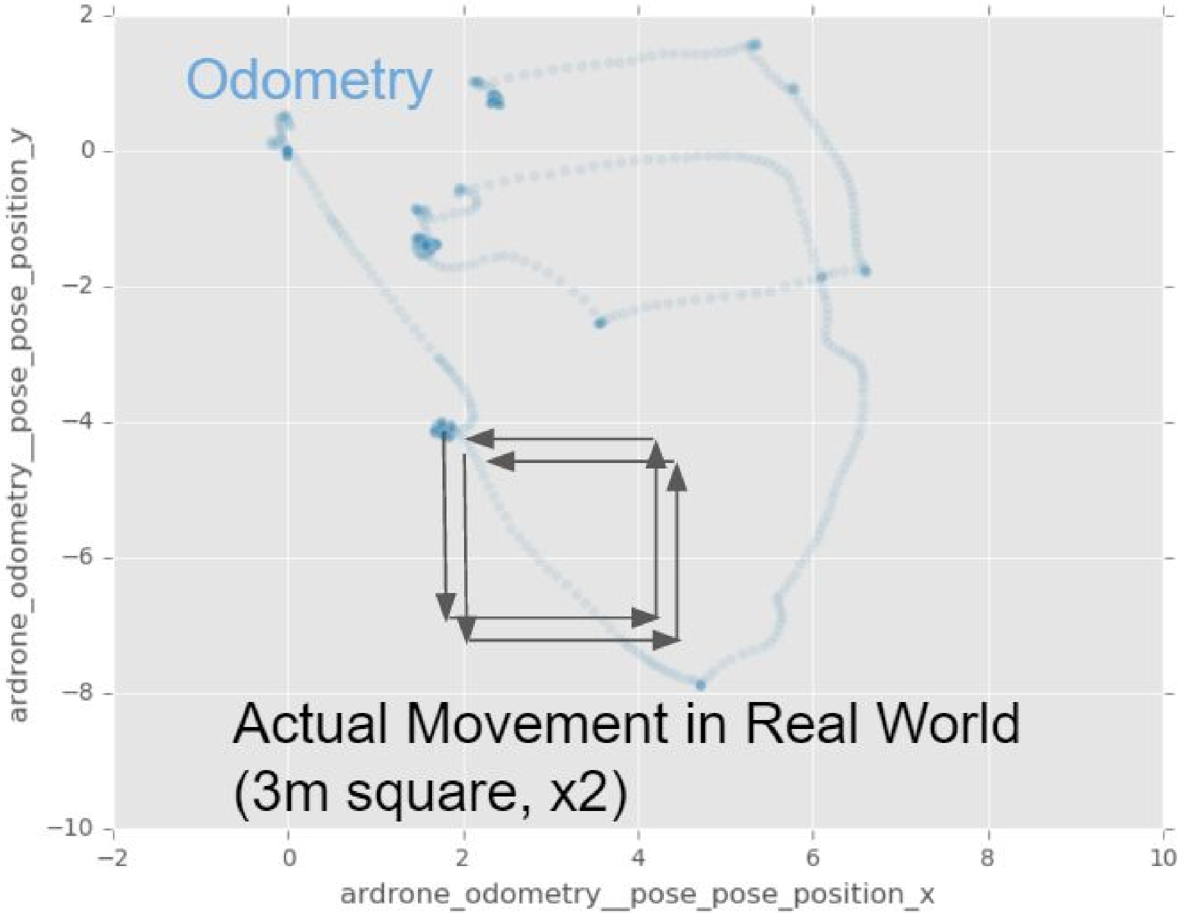
http://ardrone2.parrot.com

http://store.3drobotics.com/products/iris

TEAM C: Column Robotics

1. Project Description
2. Use Case
3. System-Level Requirements
4. Functional & Cyberphysical Architecture
5. **Current System Status**
   Requirements | System / Subsystem Descriptions | **Modeling / Analysis / Testing**
   FVE Performance Evaluation | Video | Strong / Weak Points
6. Project Management
   WBS Summary | Schedule Status | Test Plan | Budget Status | Risk Management
7. Conclusions

X vs Y Odometry Readings from Flight Test



Kalman Filtering: Computations

Notation:

A_t : *Motion Model*

B_t : *Control Input Model*

μ_t : *State Mean*

Σ_t : *State Variance*

Q_t : *Motion Model Noise*

C_t : *Observation Model*

R : *Observation Noise*

K_t : *Kalman Gain*

$z_t - C_t\bar{\mu}_t$: *Innovation*

$$\mathrm{KalmanFilter}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$$

Prediction

$$\bar{\mu}_t = A_t \mu_{t-1} + B u_t$$

$$\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^\top + Q_t$$

Gain

$$K_t = \bar{\Sigma}_t C_t^\top (C_t \bar{\Sigma}_t C_t^\top + R\ )^{-1}$$

Update

$$\mu_t = \bar{\mu}_t + K_t(z_t - C_t \bar{\mu}_t)$$

$$\Sigma_t = (I - K_t C_t)\bar{\Sigma}_t$$

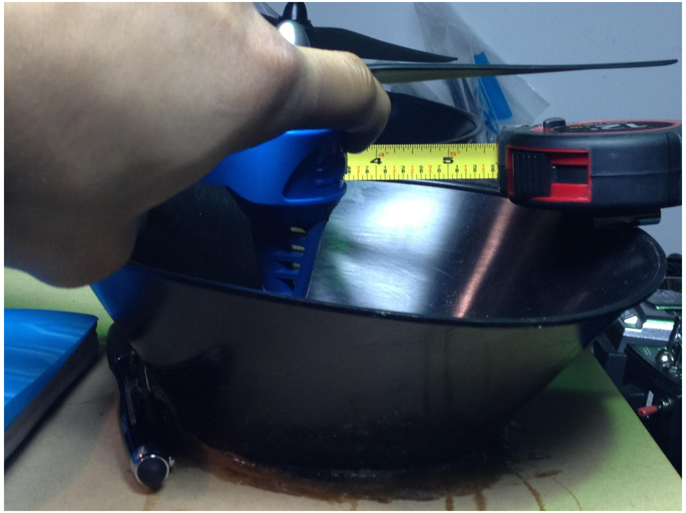
Slide courtesy Kris Kitani

No Force

No more than +/- 1 cm in displacement when fully docked, with more than 2 newtons of force applied. Average is less than 1 cm.

Push

Pull



TEAM C: Column Robotics

1. Project Description
2. Use Case
3. System-Level Requirements
4. Functional & Cyberphysical Architecture
5. **Current System Status**

   Requirements | System / Subsystem Descriptions | Modeling / Analysis / Testing

   FVE Performance Evaluation | Video | Strong / Weak Points
6. Project Management

   WBS Summary | Schedule Status | Test Plan | Budget Status | Risk Management
7. Conclusions

FVE PERFORMANCE EVALUATION

Success Conditions Met:

1. **Successful takeoff** and hover of drone under manual control
2. Drone autonomously **completes 4 search sweeps** of length > 4m each
3. Drone path during search sweeps **does not overlap with itself**
4. Drone successfully **avoided contact** with walls of hallway
5. Clear downward-facing **video feed displayed** during entire grid search process
6. Full search process **succeeded within 10 minutes** of drone takeoff

FVE PERFORMANCE EVALUATION

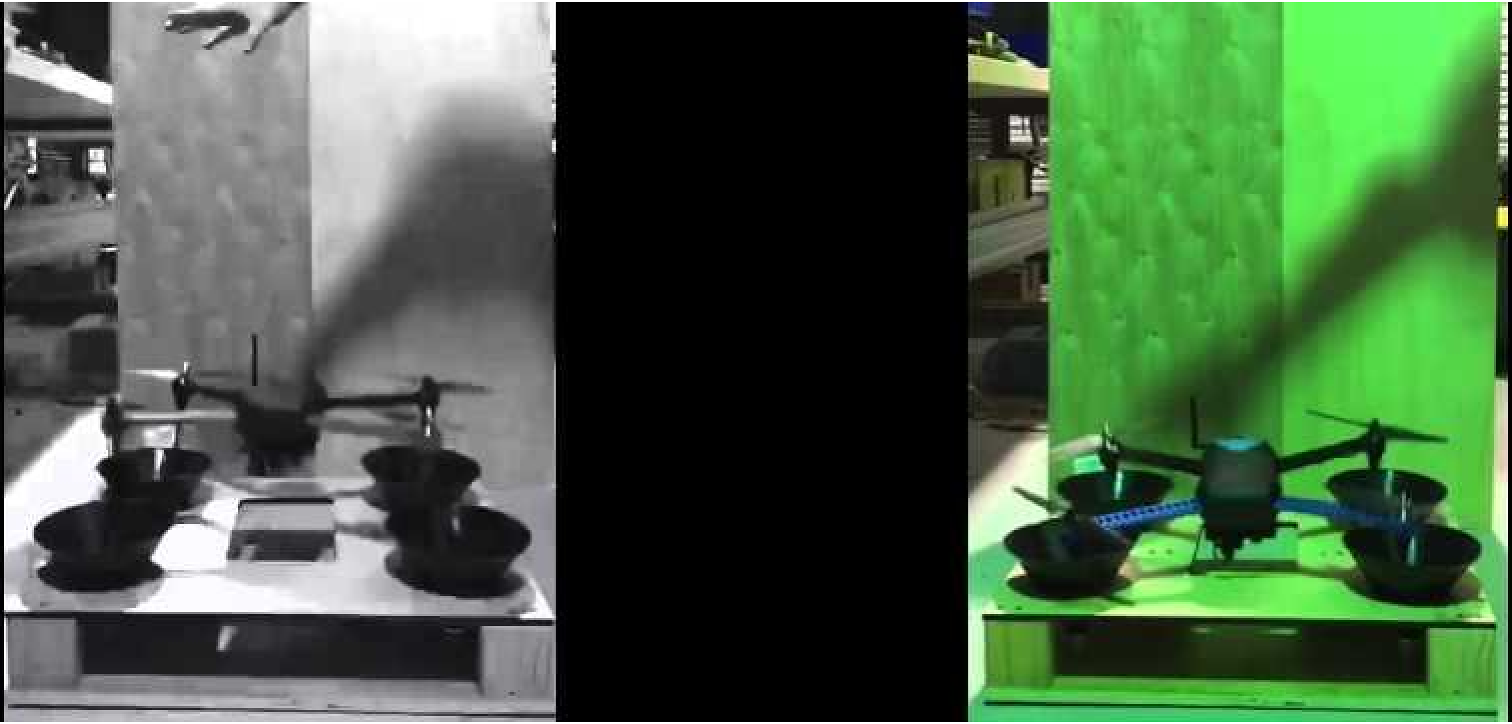
Success Conditions Met:

1. Iris+ constrained within **+/- 1cm** in all directions :
   Tighter than required +/- 2 cm in dock (5 DOF)
2. Valid orientation estimate and image (taken from the camera on the drone) is displayed on the PC
   Showed valid orientations:
   (Roll, Pitch, Yaw) = (90,0,0) and (0,90,0)
3. ‘rostopic hz’ command shows **1.09Hz** :
   Faster than required 0.1Hz on relevant topic on PC

TEAM C: Column Robotics

1. Project Description
2. Use Case
3. System-Level Requirements
4. Functional & Cyberphysical Architecture
5. **Current System Status**
   Requirements | System / Subsystem Descriptions | Modeling / Analysis / Testing
   FVE Performance Evaluation | **Video** | Strong / Weak Points
6. Project Management
   WBS Summary | Schedule Status | Test Plan | Budget Status | Risk Management
7. Conclusions

Dock Demonstration



TEAM C: Column Robotics

1. Project Description
2. Use Case
3. System-Level Requirements
4. Functional & Cyberphysical Architecture
5. **Current System Status**

   Requirements | System / Subsystem Descriptions | Modeling / Analysis / Testing

   FVE Performance Evaluation | Video | **Strong / Weak Points**
6. Project Management

   WBS Summary | Schedule Status | Test Plan | Budget Status | Risk Management
7. Conclusions

STRONG POINTS

- Robust lawnmower search with AR.Drone
- Shock absorbtion quality of the dock
- Well integrated power system
- Compact design for sensor and SBC mounting

WEAK POINTS

- Automated Iris+ control untested
- Small backwards drift of AR.Drone
- Jerky waypoint navigation
- Unsightly epoxy contaminating dock

TEAM C: Column Robotics

1. Project Description
2. Use Case
3. System-Level Requirements
4. Functional & Cyberphysical Architecture
5. Current System Status

   Requirements | System / Subsystem Descriptions | Modeling / Analysis / Testing

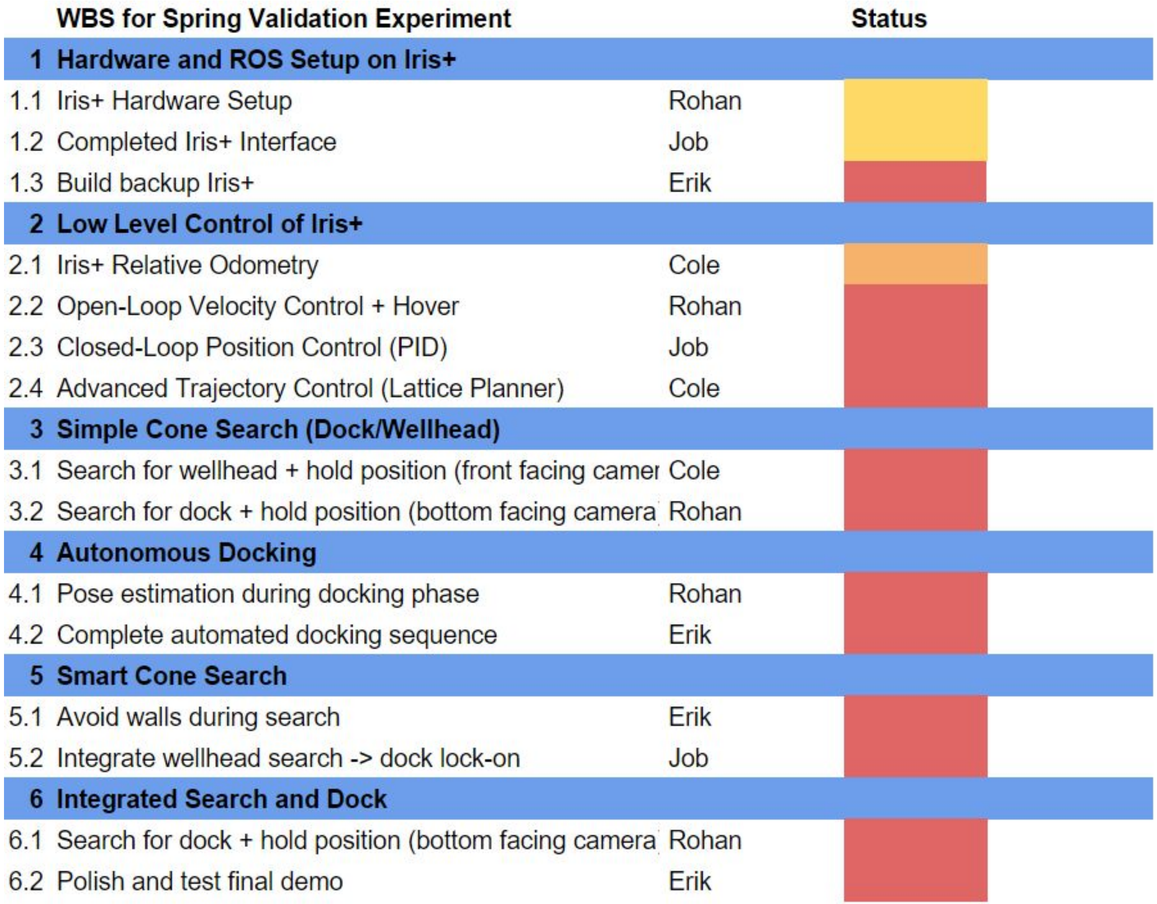
   FVE Performance Evaluation | Video | Strong / Weak Points
6. **Project Management**

   WBS Summary | Schedule Status | Test Plan | Budget Status | Risk Management
7. Conclusions

WORK BREAKDOWN STRUCTURE FALL

	WBS for Fall Validation Experiment	Status:	**PDR**	**Now**
1	**Open-loop ARDrone Control: takeoff, land, move from PC**			
1.1	Low-level open-loop control / takeoff in ROS	Erik/Job/R	🟩	🟩
1.2	Display ROS node graph	Erik	🟩	🟩
2	**Fall AR.Drone Position X,Y Movement Demo**			
2.1	AR.Drone relative odometry working in ROS	Erik (Cole)	🟨	🟩
2.2	Closed loop on absolute position control	Cole (Erik)	🟥	🟩
2.3	Integrate AR.Drone demo subsystems	Erik	🟥	🟩
3	**Hardware and ROS Setup on Iris+**			
3.1	Pixhawk -> SBC -> PC ROS setup + sensor display	Rohan (Job)	🟧	🟩
3.2	Completed Iris+ Hardware Setup	Job (Rohan)	🟥	🟨
4	**Dock Prototype**			
4.1	Formulate Dock Design Criteria	Job/Team	N/A	🟩
4.2	Dock Internal CODR	Team	N/A	🟩
4.3	Manufacturable CAD Model of Dock	Job	🟥	🟩
4.4	Tested, working physical prototype	Job	🟥	🟩
5	**Non-Demo Focus Areas:**			
5.1	Iris+ Relative Odometry	Rohan	🟥	🟧
5.2	Stable Open-Loop Control of Iris+		🟥	🟥
5.3	Integrated closed-loop position control of Iris+		🟥	🟥
5.4	Searching for tag on ground with AR.Drone	Cole	N/A	🟥

WORK BREAKDOWN STRUCTURE   SPRING



	WBS for Spring Validation Experiment		Status
1	**Hardware and ROS Setup on Iris+**		
1.1	Iris+ Hardware Setup	Rohan	🟨
1.2	Completed Iris+ Interface	Job	🟨
1.3	Build backup Iris+	Erik	🟥
2	**Low Level Control of Iris+**		
2.1	Iris+ Relative Odometry	Cole	🟧
2.2	Open-Loop Velocity Control + Hover	Rohan	🟥
2.3	Closed-Loop Position Control (PID)	Job	🟥
2.4	Advanced Trajectory Control (Lattice Planner)	Cole	🟥
3	**Simple Cone Search (Dock/Wellhead)**		
3.1	Search for wellhead + hold position (front facing camer	Cole	🟥
3.2	Search for dock + hold position (bottom facing camera	Rohan	🟥
4	**Autonomous Docking**		
4.1	Pose estimation during docking phase	Rohan	🟥
4.2	Complete automated docking sequence	Erik	🟥
5	**Smart Cone Search**		
5.1	Avoid walls during search	Erik	🟥
5.2	Integrate wellhead search -> dock lock-on	Job	🟥
6	**Integrated Search and Dock**		
6.1	Search for dock + hold position (bottom facing camera	Rohan	🟥
6.2	Polish and test final demo	Erik	🟥

TEAM C: Column Robotics

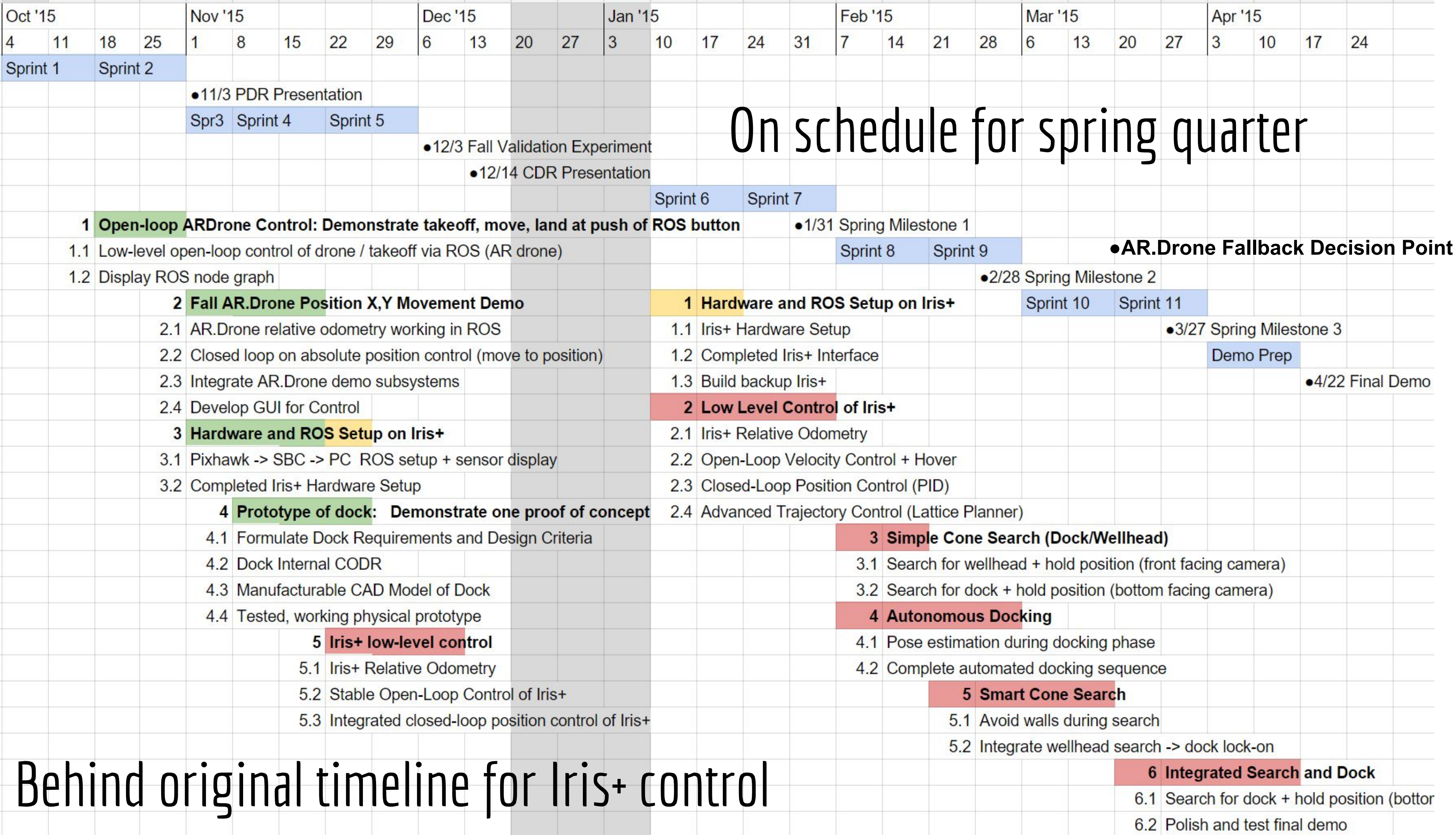
1. Project Description
2. Use Case
3. System-Level Requirements
4. Functional & Cyberphysical Architecture
5. Current System Status

   Requirements | System / Subsystem Descriptions | Modeling / Analysis / Testing

   FVE Performance Evaluation | Video | Strong / Weak Points
6. **Project Management**

   WBS Summary | **Schedule Status** | Test Plan | Budget Status | Risk Management
7. Conclusions

On schedule for spring quarter



Oct '15 | Nov '15 | Dec '15 | Jan '15 | Feb '15 | Mar '15 | Apr '15

4 11 18 25 | 1 8 15 22 29 | 6 13 20 27 | 3 10 17 24 31 | 7 14 21 28 | 6 13 20 27 | 3 10 17 24

Sprint 1 Sprint 2

●11/3 PDR Presentation

Spr3 Sprint 4 Sprint 5

●12/3 Fall Validation Experiment

●12/14 CDR Presentation

Sprint 6 Sprint 7

1 **Open-loop ARDrone Control: Demonstrate takeoff, move, land at push of ROS button**

1.1 Low-level open-loop control of drone / takeoff via ROS (AR drone)

1.2 Display ROS node graph

2 **Fall AR.Drone Position X,Y Movement Demo**

2.1 AR.Drone relative odometry working in ROS

2.2 Closed loop on absolute position control (move to position)

2.3 Integrate AR.Drone demo subsystems

2.4 Develop GUI for Control

3 **Hardware and ROS Setup on Iris+**

3.1 Pixhawk -> SBC -> PC ROS setup + sensor display

3.2 Completed Iris+ Hardware Setup

4 **Prototype of dock: Demonstrate one proof of concept**

4.1 Formulate Dock Requirements and Design Criteria

4.2 Dock Internal CODR

4.3 Manufacturable CAD Model of Dock

4.4 Tested, working physical prototype

5 **Iris+ low-level control**

5.1 Iris+ Relative Odometry

5.2 Stable Open-Loop Control of Iris+

5.3 Integrated closed-loop position control of Iris+

●1/31 Spring Milestone 1

Sprint 8 Sprint 9

●AR.Drone Fallback Decision Point

●2/28 Spring Milestone 2

Sprint 10 Sprint 11

●3/27 Spring Milestone 3

Demo Prep

●4/22 Final Demo

1 **Hardware and ROS Setup on Iris+**

1.1 Iris+ Hardware Setup

1.2 Completed Iris+ Interface

1.3 Build backup Iris+

2 **Low Level Control of Iris+**

2.1 Iris+ Relative Odometry

2.2 Open-Loop Velocity Control + Hover

2.3 Closed-Loop Position Control (PID)

2.4 Advanced Trajectory Control (Lattice Planner)

3 **Simple Cone Search (Dock/Wellhead)**

3.1 Search for wellhead + hold position (front facing camera)

3.2 Search for dock + hold position (bottom facing camera)

4 **Autonomous Docking**

4.1 Pose estimation during docking phase

4.2 Complete automated docking sequence

5 **Smart Cone Search**

5.1 Avoid walls during search

5.2 Integrate wellhead search -> dock lock-on

6 **Integrated Search and Dock**

6.1 Search for dock + hold position (bottor

6.2 Polish and test final demo

Behind original timeline for Iris+ control

TEAM C: Column Robotics

1. Project Description
2. Use Case
3. System-Level Requirements
4. Functional & Cyberphysical Architecture
5. Current System Status

   Requirements | System / Subsystem Descriptions | Modeling / Analysis / Testing

   FVE Performance Evaluation | Video | Strong / Weak Points
6. **Project Management**

   WBS Summary | Schedule Status | **Test Plan** | Budget Status | Risk Management
7. Conclusions

HIGH LEVEL TEST PLAN

Deadline	**Deliverable Functionality**	**Method to Test**
Late January Progress Review 7	Low level control of Iris+. Backup Iris+ hardware completed	Stable, teleoperated control of iris+ via ROS. Demonstrate in net.
Mid-February Progress review 8	Simple cone search with Iris+	Cone-shaped search pattern approaching wellhead; stop when wellhead tag identified.
Late February Progress Review 9	Autonomous docking of Iris+	Autonomously recognize dock from above, approach and land on dock, confirm rigidity in 5 DOF
Mid-March Progress Review 10	Smart cone search with Iris+ **(AR.Drone Fallback Decision Point)**	Iris+ searches for wellhead and locks on dock position while avoiding hitting walls.
Early April Progress Review 11	Integrated Search and Dock	Iris+ searches for wellhead, locks on dock position, and autonomously docks.
Mid April Progress Review 12	Integrated working system	Full demo: Take off, Search for wellhead, Orient to dock, land on dock, send signal.
April 22 and April 29 Spring Validation Experiment	Demonstration of integrated system	Same as above, but better!

Spring Validation Experiment

Needed Equipment: Iris+ with hardware, wellhead, dock, caution tape
Operational Area: 25m^2 in B - Level Basement
Test Process:

1. Cordon off section of hallway
2. Place wellhead at one corner of search area and dock 1m in front of the wellhead
3. Place Iris+ on ground at opposite corner of search area facing wellhead within +/- 5 degrees
4. Hit START button on PC to initiate sequence
5. Confirm Iris+ lifts off and begins searching for wellhead (marker)
6. Confirm Iris+ arrives within 3 meter radius of wellhead
7. Confirm Iris+ orients above dock in pre-docking position (within 1 meter of dock)
8. Confirm Iris+ successfully lands in dock, constrained in 5 DOF

Success Conditions:
Mandatory:

1. Iris+ autonomously takes off from ground
2. Iris+ arrives within 3 meter radius of wellhead
3. Dock with docking station, constrained in 5 DOF

Desired:

1. Dock constraints 6 DOF
2. Successfully avoid obstacles

TEAM C: Column Robotics

1. Project Description
2. Use Case
3. System-Level Requirements
4. Functional & Cyberphysical Architecture
5. Current System Status

   Requirements | System / Subsystem Descriptions | Modeling / Analysis / Testing
   FVE Performance Evaluation | Video | Strong / Weak Points
6. **Project Management**

   WBS Summary | Schedule Status | Test Plan | **Budget Status** | Risk Management
7. Conclusions

Budget

Total budget: $4000

Total spent to date: $1611.11

Big ticket items:

- 3DR Iris+ Drone: $599
- Minnowboard Max x86 SBC: $150
- Odroid XU-4 Arm SBC: $83
- NicaDrone Magnet: $90
- PX4 Flow Optical Flow Camera: $149

TEAM C: Column Robotics

1. Project Description
2. Use Case
3. System-Level Requirements
4. Functional & Cyberphysical Architecture
5. Current System Status

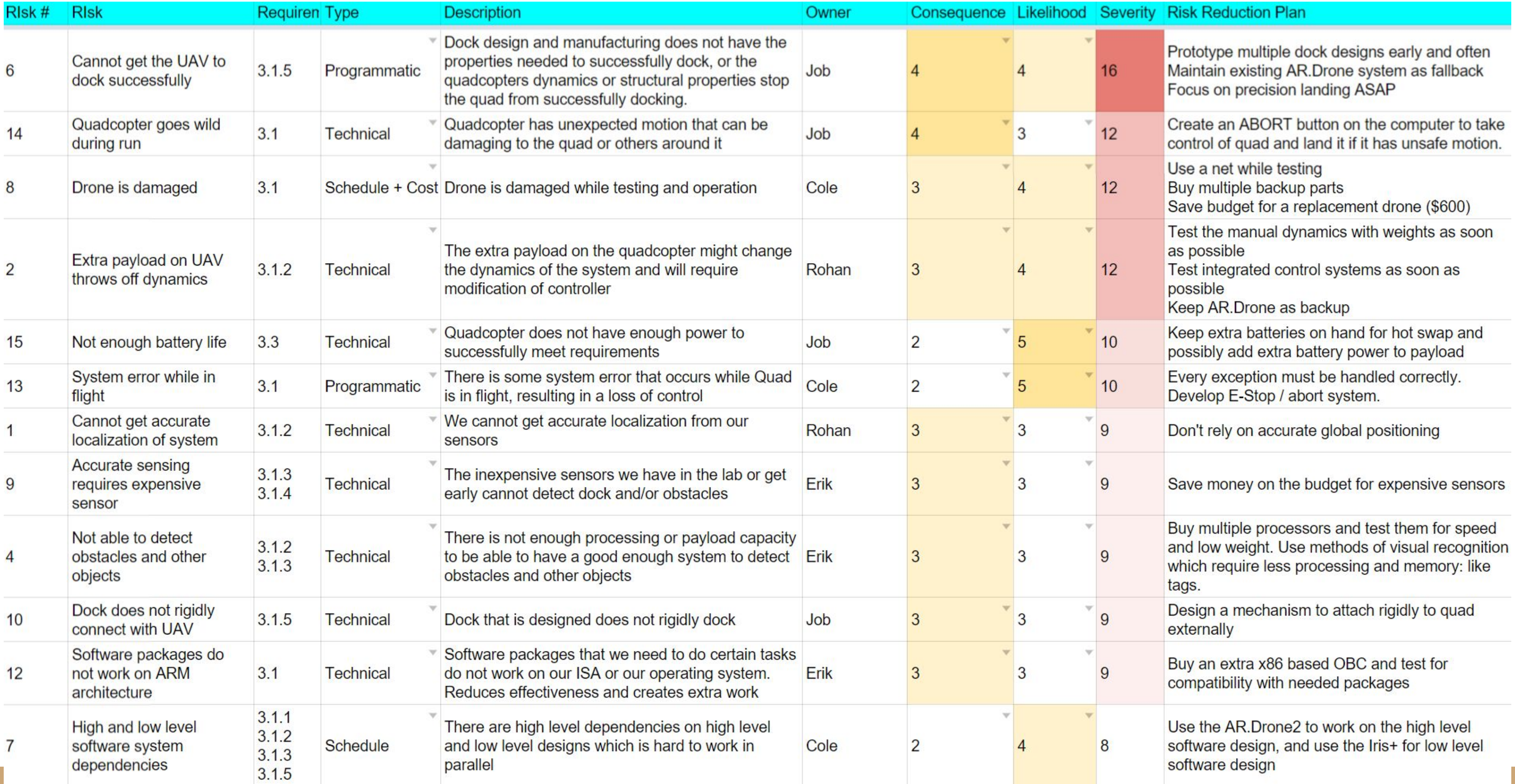
   Requirements | System / Subsystem Descriptions | Modeling / Analysis / Testing

   FVE Performance Evaluation | Video | Strong / Weak Points
6. **Project Management**

   WBS Summary | Schedule Status | Test Plan | Budget Status | **Risk Management**
7. Conclusions

Risk Management

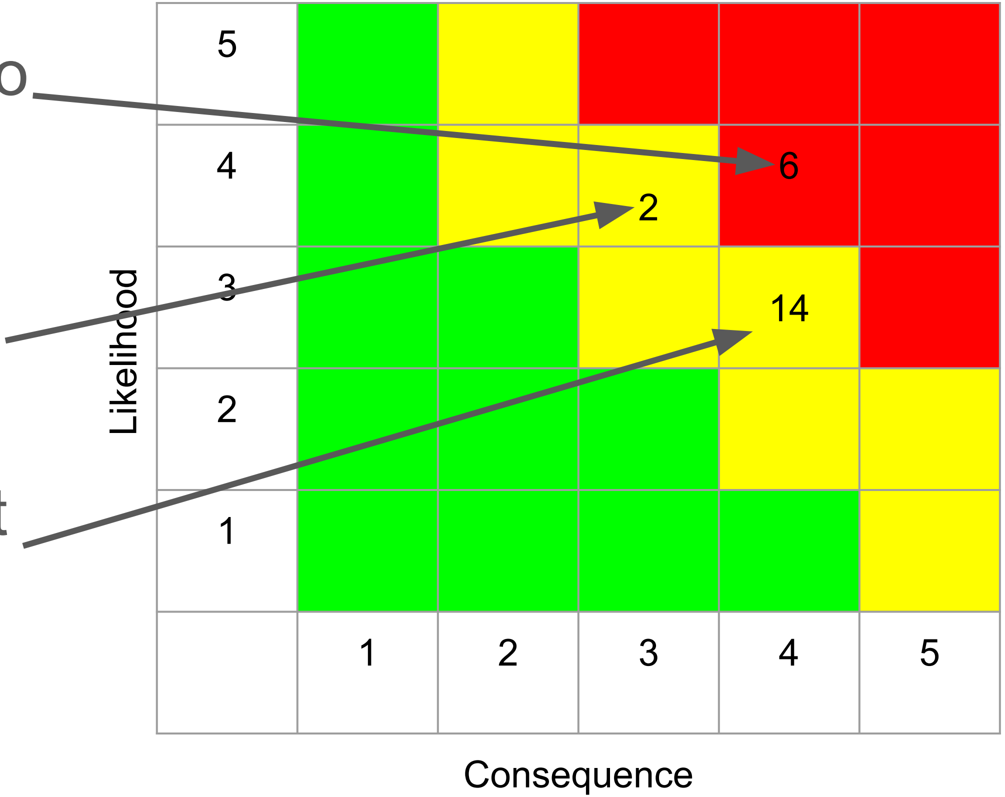
severity = consequence * likelihood



RIsk #	RIsk	Requiren	Type	Description	Owner	Consequence	Likelihood	Severity	Risk Reduction Plan
6	Cannot get the UAV to dock successfully	3.1.5	Programmatic	Dock design and manufacturing does not have the properties needed to successfully dock, or the quadcopters dynamics or structural properties stop the quad from successfully docking.	Job	4	4	16	Prototype multiple dock designs early and often Maintain existing AR.Drone system as fallback Focus on precision landing ASAP
14	Quadcopter goes wild during run	3.1	Technical	Quadcopter has unexpected motion that can be damaging to the quad or others around it	Job	4	3	12	Create an ABORT button on the computer to take control of quad and land it if it has unsafe motion.
8	Drone is damaged	3.1	Schedule + Cost	Drone is damaged while testing and operation	Cole	3	4	12	Use a net while testing Buy multiple backup parts Save budget for a replacement drone ($600)
2	Extra payload on UAV throws off dynamics	3.1.2	Technical	The extra payload on the quadcopter might change the dynamics of the system and will require modification of controller	Rohan	3	4	12	Test the manual dynamics with weights as soon as possible Test integrated control systems as soon as possible Keep AR.Drone as backup
15	Not enough battery life	3.3	Technical	Quadcopter does not have enough power to successfully meet requirements	Job	2	5	10	Keep extra batteries on hand for hot swap and possibly add extra battery power to payload
13	System error while in flight	3.1	Programmatic	There is some system error that occurs while Quad is in flight, resulting in a loss of control	Cole	2	5	10	Every exception must be handled correctly. Develop E-Stop / abort system.
1	Cannot get accurate localization of system	3.1.2	Technical	We cannot get accurate localization from our sensors	Rohan	3	3	9	Don't rely on accurate global positioning
9	Accurate sensing requires expensive sensor	3.1.3 3.1.4	Technical	The inexpensive sensors we have in the lab or get early cannot detect dock and/or obstacles	Erik	3	3	9	Save money on the budget for expensive sensors
4	Not able to detect obstacles and other objects	3.1.2 3.1.3	Technical	There is not enough processing or payload capacity to be able to have a good enough system to detect obstacles and other objects	Erik	3	3	9	Buy multiple processors and test them for speed and low weight. Use methods of visual recognition which require less processing and memory: like tags.
10	Dock does not rigidly connect with UAV	3.1.5	Technical	Dock that is designed does not rigidly dock	Job	3	3	9	Design a mechanism to attach rigidly to quad externally
12	Software packages do not work on ARM architecture	3.1	Technical	Software packages that we need to do certain tasks do not work on our ISA or our operating system. Reduces effectiveness and creates extra work	Erik	3	3	9	Buy an extra x86 based OBC and test for compatibility with needed packages
7	High and low level software system dependencies	3.1.1 3.1.2 3.1.3 3.1.5	Schedule	There are high level dependencies on high level and low level designs which is hard to work in parallel	Cole	2	4	8	Use the AR.Drone2 to work on the high level software design, and use the Iris+ for low level software design

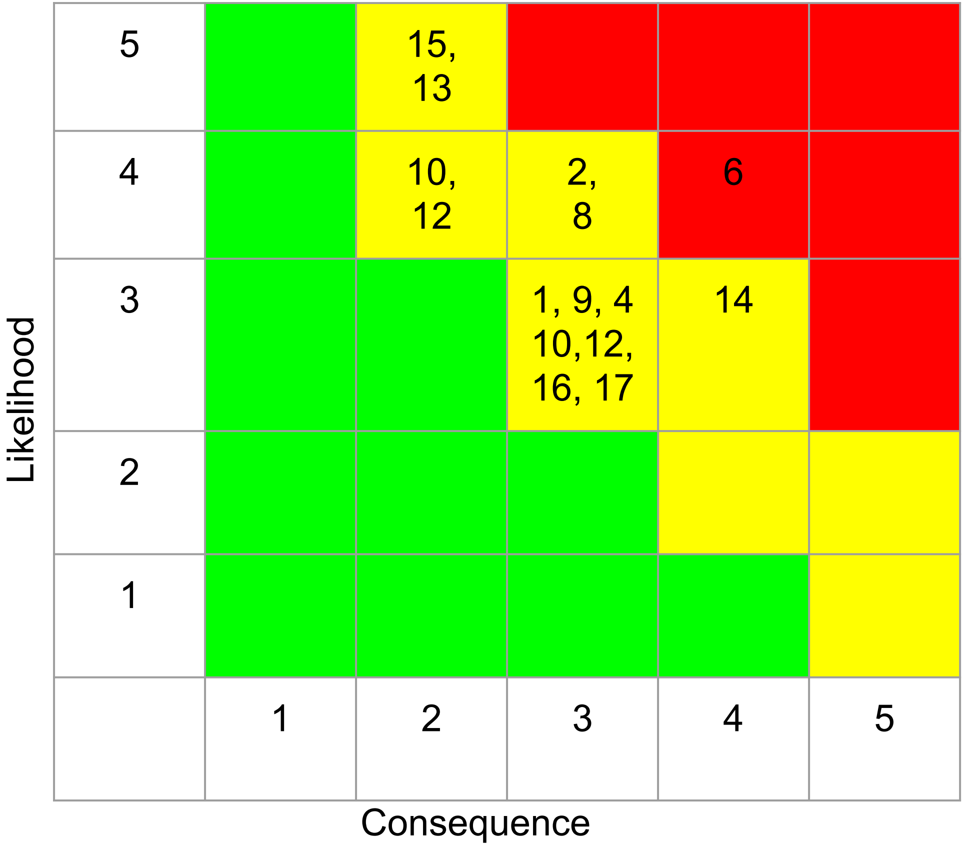
Top Risks

1. We cannot get the UAV to dock successfully
2. Extra payload on UAV throws off dynamics of system
3. UAV goes completely out of control during the run damaging itself or someone/something else



5	🟩	🟨	🟥	🟥	🟥
4	🟩	🟨	🟨 2	🟥 6	🟥
3	🟩	🟩	🟨	🟨 14	🟥
2	🟩	🟩	🟩	🟨	🟨
1	🟩	🟩	🟩	🟩	🟨
	1	2	3	4	5

Risk Matrix



Likelihood

5	🟩	🟨 15, 13	🟥	🟥	🟥
4	🟩	🟨 10, 12	🟨 2, 8	🟥 6	🟥
3	🟩	🟩	🟨 1, 9, 4 10,12, 16, 17	🟨 14	🟥
2	🟩	🟩	🟩	🟨	🟨
1	🟩	🟩	🟩	🟩	🟨
	1	2	3	4	5

Consequence

Risk Mitigated

Risk

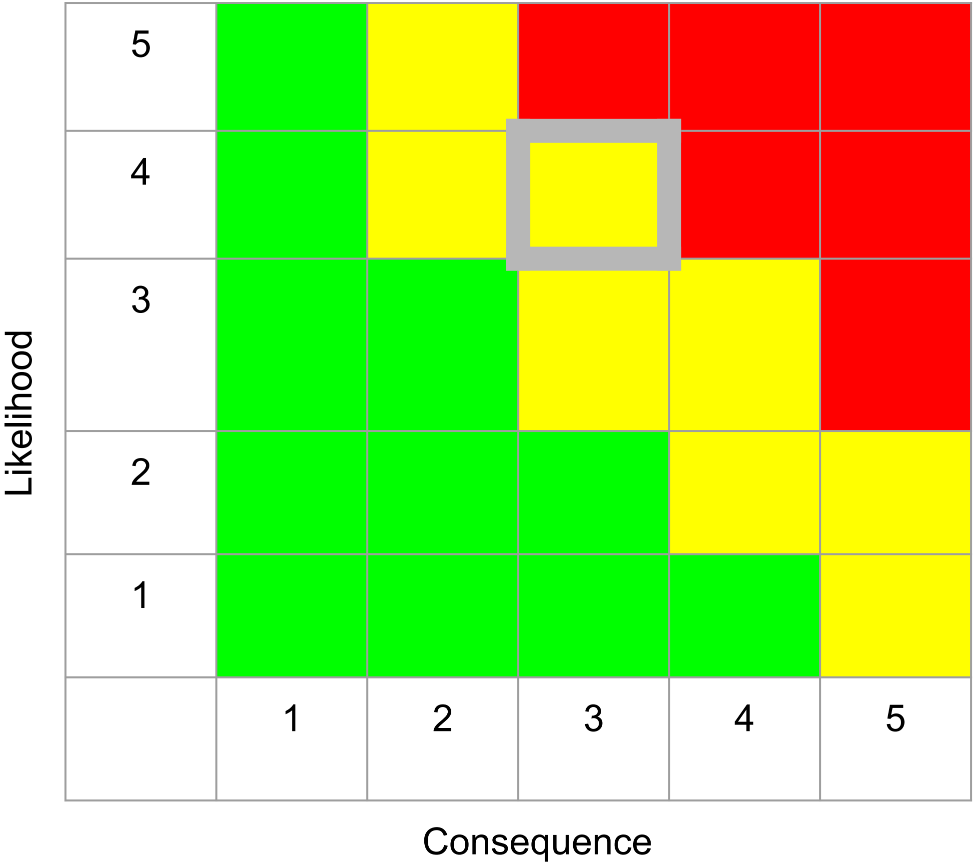
Extra payload on UAV throws off dynamics

Risk Mitigation

Position control in AR.Drone validated

Risk Mitigated

November 11, 2015



5	🟩	🟨	🟥	🟥	🟥
4	🟩	🟨	🟨	🟥	🟥
3	🟩	🟩	🟨	🟨	🟥
2	🟩	🟩	🟩	🟨	🟨
1	🟩	🟩	🟩	🟩	🟨
	1	2	3	4	5

Risk Mitigation Strategies

Risk ID:	Risk Title:	Risk Owner:	Date Submitted:	Date Updated:
6	Cannot get UAV to successfully dock	Job	10/21/2015	12/13/2015

Description:
Dock design and manufacturing does not have the properties needed to successfully dock, or the quadcopters dynamics or structural properties stop the quad from successfully docking.

Consequences:	Risk Type:	Risk Level:
The quadcopter will not be able to dock, and a major performance requirement will not be able to be accomplished	- Technical - Programmatic	16

Risk Reduction Plan	Expected Outcome:	Comments
1. Prototype multiple dock designs early and often 2. Focus resources on precision landing **3. March 9th as decision date to switch from Iris+ to AR.Drone**	Majority of work time spent on developing controls and hardware of dock	

Added Risk Mitigation Strategies

Risk ID:	Risk Title:	Risk Owner:	Date Submitted:	Date Updated:
16	AR.Drone breaks during testing	Cole	11/15/2015	11/25/2015

Description:
AR.Drone breaks or is damaged during a test run before the FVE

Consequences:	Risk Type:	Risk Level:
Team will not be able to complete the FVE challenge	- Schedule - Programmatic	YELLOW 9 / 25

Risk Reduction Plan	Expected Outcome:	Comments
1. Take out a second AR.Drone from inventory	AR.Drone is available in inventory, so this will be no problem	MITIGATED

TEAM C: Column Robotics

1. Project Description
2. Use Case
3. System-Level Requirements
4. Functional & Cyberphysical Architecture
5. Current System Status

   Requirements | System / Subsystem Descriptions | Modeling / Analysis / Testing

   FVE Performance Evaluation | Video | Strong / Weak Points

6. Project Management

   WBS Summary | Schedule Status | Test Plan | Budget Status | Risk Management

7. **Conclusions**

Lessons Learned

- It is difficult to communicate and get everyone on the same page
- One person’s plan may not meet what other’s feel
- Easy to get busy with other things and not deliver every sprint

Key Spring Activities

- It is difficult to communicate and get everyone on the same page
  - **Go back to requirements**

- One person’s plan may not meet what other’s feel
  - **Communicate and record**

- Easy to get busy with other things and not deliver every sprint
  - **Get specific demonstrable deliverables for each sprint for each person**
  - **Show demos at the sprint kick-offs**

Questions