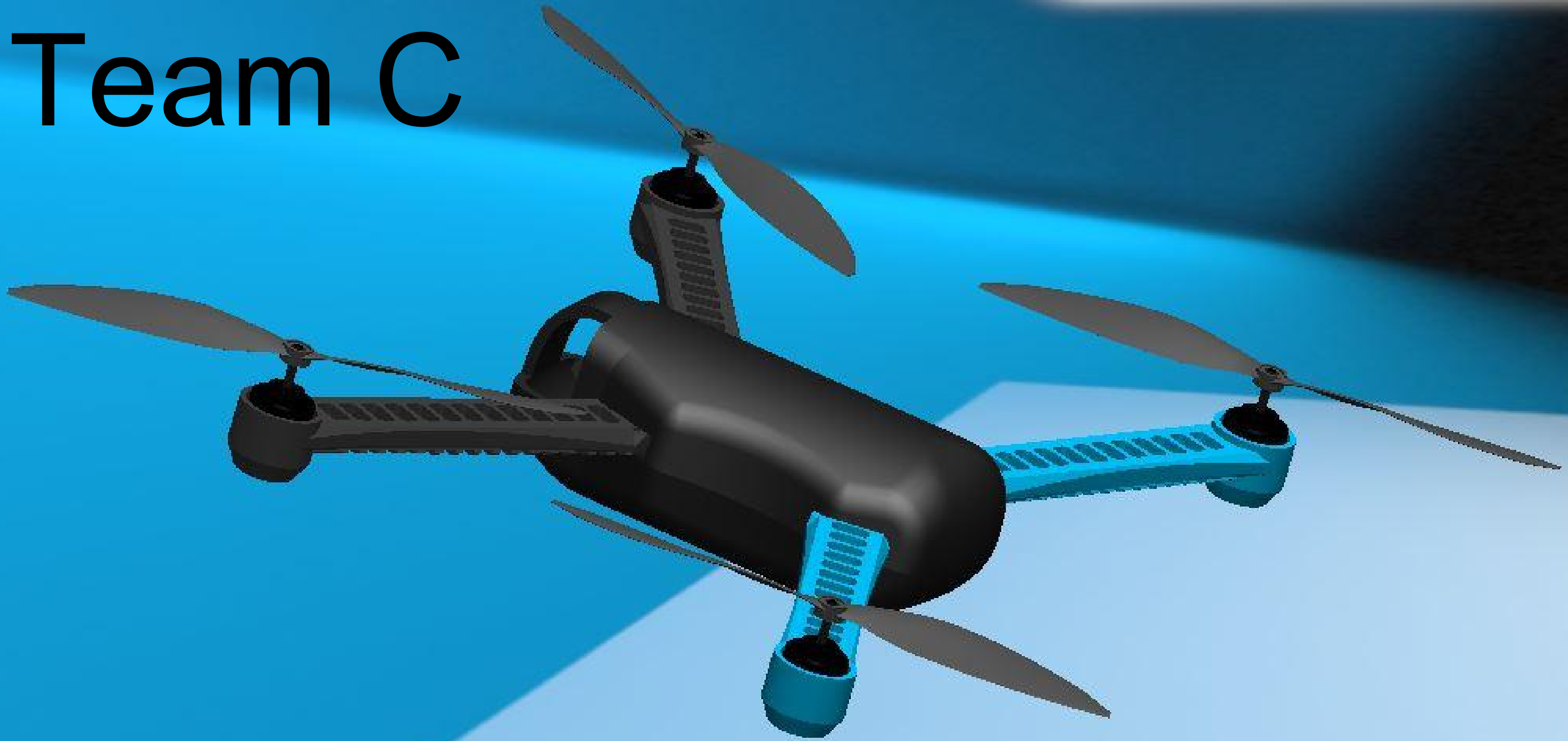
Team C



Systems Engineering Presentation 3

WBS reflecting good progress towards FVE

On track for FVE deliverables

AR.Drone2 platform performing well with EKF state estimation

Iris+ platform hardware progressing nicely

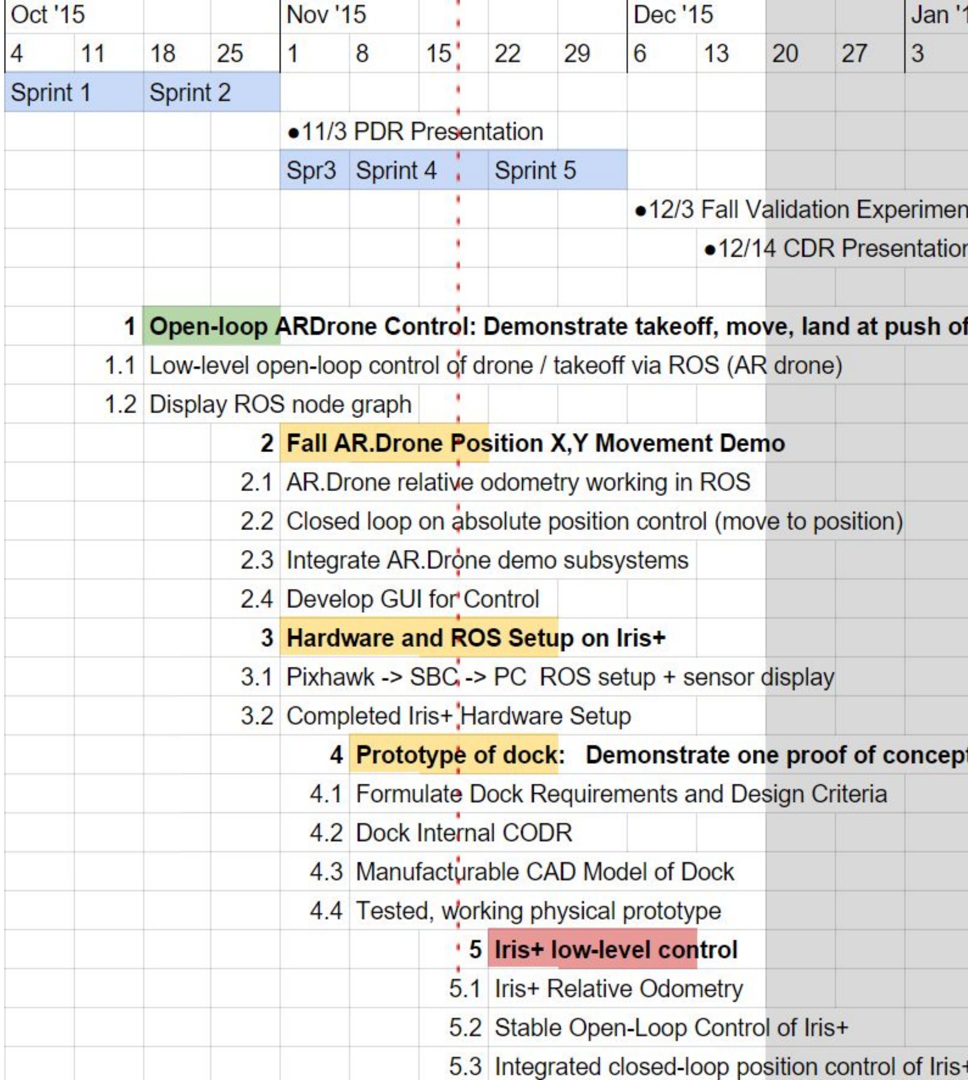
Fundamental review of dock design process fruitful

Growing amount of non-demo work...

	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	

Still Forecasting Success

- **On track for all FVE deliverables**
- Growing non-demo workload needs re-scoping
  - plan to update non-demo goals for this semester during sprint 5 kickoff



System Progress: Extended Kalman Filters Work

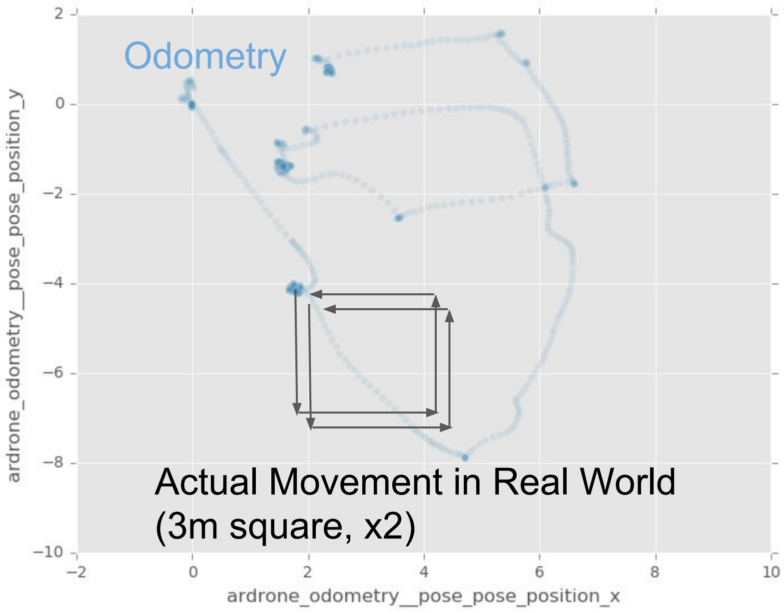
Issue: Standard AR.Drone2 on-board odometry showed massive drift

- Up to 3+ meters drift over 3 meters

Order-of-magnitude improvement leveraging Extended Kalman Filter + dynamics model

- <<1m drift over the same flight pattern shown to the right
- EKF and quadrotor movement model from tum_ardrone ROS package

X vs Y Odometry Readings from Flight Test



System Progress: Iris+ hardware coming together

- Power distribution board design complete
- Acquired hardware (IRIS+, Camera, SBC, PX4Flow)
- Set up Odroid-XU4
- Read sensor data from IRIS+ over WiFi
- Tested SVO visual odometry algorithm
- Dock design review

Updated Fall Validation Experiment

Test stage 1: Accurate AR.Drone2 Odometry
Location: NSH B-level hallway
Equipment: Laptop, AR.Drone2; Caution tape; Target marker

Test process:

1. Cordon off section of hallway
2. Place AR.Drone on ground and connect via WiFi
3. Place target area identifier (diameter 1 m) at target location (within 6m of (0,0) location)
4. Hit button for takeoff. Confirm ARDrone is stable
5. Once stable, move drone to (0,0) location decided in step 3
6. Input target coordinate in meters into interface
7. After movement is completed, mark position of drone
8. Confirm drone is partially within target area marker
9. Repeat steps 3-8 for second target location

Success Conditions:

1. AR.Drone hovering partially over target area marker (diam = 1m)
2. Success on 2 of 2 trials (within 2 minutes per trial)

Updated Fall Validation Experiment

Test stage 2: Hardware Setup of Dock and Iris+ Drone
Location: NSH B-level MRSD Lab
Equipment: Dock Hardware, Iris+ hardware

Test process:

1. Validate mounting of camera and SBC
2. Position Dock prototype hardware on benchtop
3. Physically mate Iris+ to dock and demonstrate physical fit
   a. Confirm rigidity in 5 DOF
4. Boot SBC and Pixhawk on Iris+ and:
   a. Run ROS LAUNCH
   b. Observe orientation estimation of Iris+ orientation on PC
   c. Observe camera feed on the PC

Success Conditions:

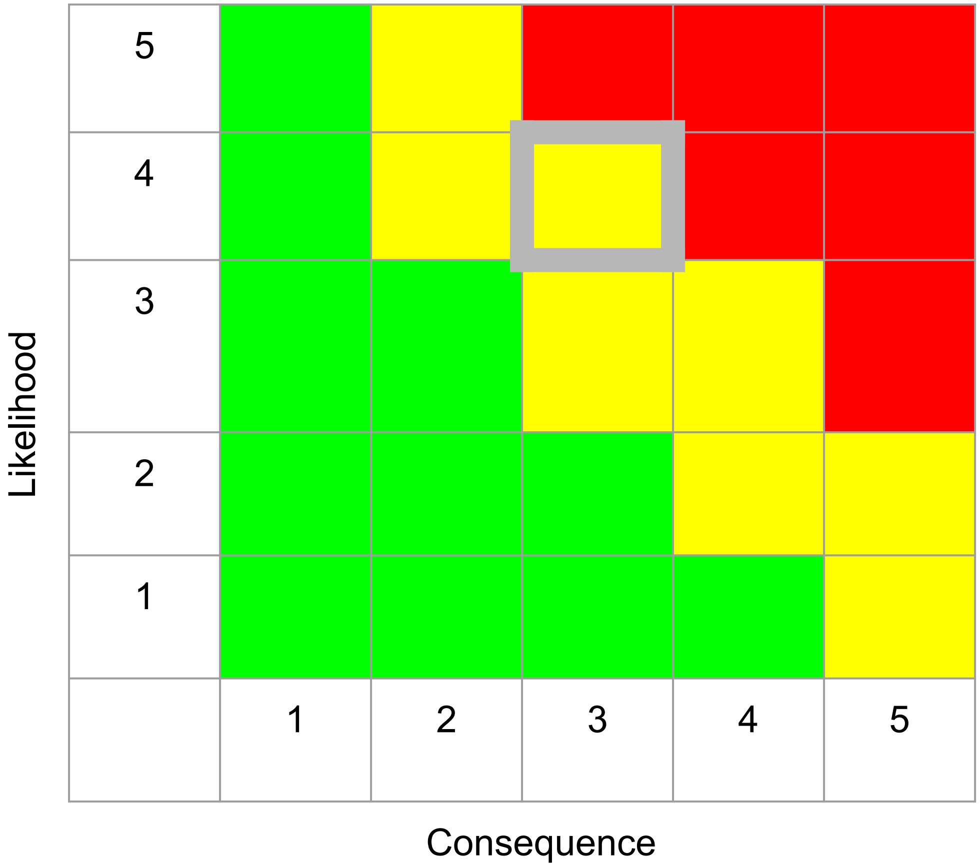
1. Iris+ constrained within +/- 2 cm in dock (5 DOF)
2. Valid odometry data displayed and PC
3. ‘rostopic hz’ command shows > 0.1Hz on relevant topic on PC

Updated Performance requirements

1. AR.Drone moves to any specified location (x,y) within 6m of (0,0) and is hovering partially over target area marker (**diam = 1m**)
2. AR.Drone reaches target within **2 minutes** per trial
3. AR.Drone successfully completes 2 out of 2 trials
4. Iris+ is constrained within **+/- 2cm in dock** (5 DOF)
5. Communication from Iris+ SBC to PC at frequency greater than **0.1Hz**

Risk Mitigated

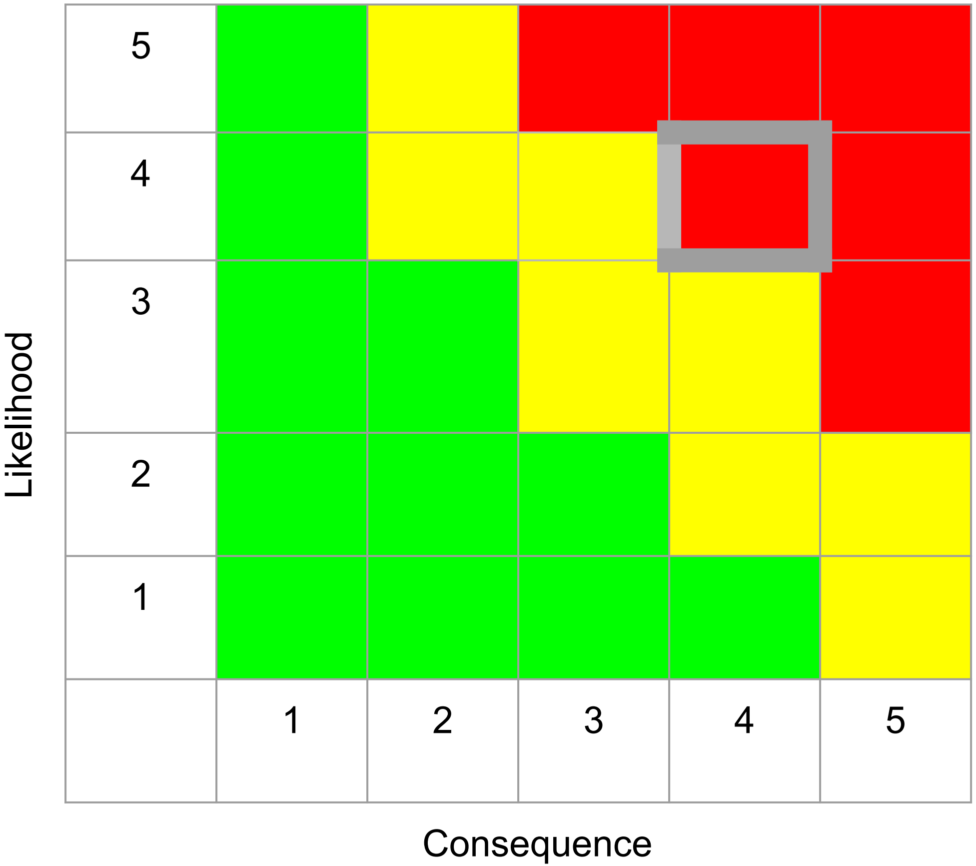
Risk
Extra payload on UAV throws off dynamics
Risk Mitigation
Position control in AR.Drone validated
Risk Mitigated
November 11, 2015



Risk Mitigated

Risk
UAV cannot successfully dock
Risk Mitigation
Run tests on cone slopes
Risk Mitigated
In Progress

walmart.com



Added Risk Mitigation Strategies

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

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

Added Risk Mitigation Strategies

Risk ID:	Risk Title:	Risk Owner:	Date Submitted:	Date Updated:
17	Dock parts do not come in on time or are ineffective	Job	11/15/2015	11/15/2015

Description:
During the manufacturing process, the designed or manufactured parts are not effective and need to be replaced, but there is not time.

Consequences:	Risk Type:	Risk Level:
The team will not be able to complete the FVE effectively	- Technical - Programmatic - Schedule	YELLOW 9 / 25

Risk Reduction Plan	Expected Outcome:	Comments
1. Order multiple dock prototype parts of different properties 2. Order parts ASAP	Dock Design will be able to be completed before the FVE	

Questions?

	Demo Functionality	Task Description	Owner	Sprint	Not Started	In Progi	Comple	Est.	Left	Done	Dependencie
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)**		**Erik/Job/R**								
		Set up ROS framework / GIT repo		1			Erik	4	0	3	e
		Write READER node to acquire AR.Drone data		2			Rohan	8	0	8	e
		Design and write MOVER node to issue commands		2			Job	8	0	8	e
		Document and share READER interfaces		2			Rohan	1	0	0	e
		Document and share MOVER interfaces		2			Job	1	0	1	e
		Integrate and test control script (implement move-while-pressed)		2			Cole/Job	8	0	3	e
		Add takeoff / land / abort features to MOVER		2			Job	4	0	4	e
1.2	**Display ROS node graph**		**Erik**								
		Setup launch script for nodes and topics		2			Erik	2	0	3	5
2	**Fall AR.Drone Position X,Y Movement Demo**										
2.1	**AR.Drone relative odometry working in ROS**		**Erik (Cole)**	3							
		Test if x,y odometry is working when AR.Drone in flight		3			Erik	1	0	1	a
		Evaluate / test accuracy and drift of built-in odometry		3			Erik	2	0	2	a
		Get it all working / documented					Erik	4	0	2	a
2.2	**Closed loop on absolute position control (move to position)**		**Cole (Erik)**	3							
		Research + document robot_localization package		1			Erik	2	0	2	
		Implement 2d x,y map display in ROS		2			Erik	4	0	4	
		Test current implementation with simple waypoint		3			Erik	2		1	a
		Get closed loop system working / documented		3			.	8	0		a
2.3	**Integrate AR.Drone demo subsystems**		**Erik**	4							
		Thoroughly test actual demo process (with adversrial)		4		Erik		4	1	3	
2.4	**Develop GUI for Control**						.	8	0		