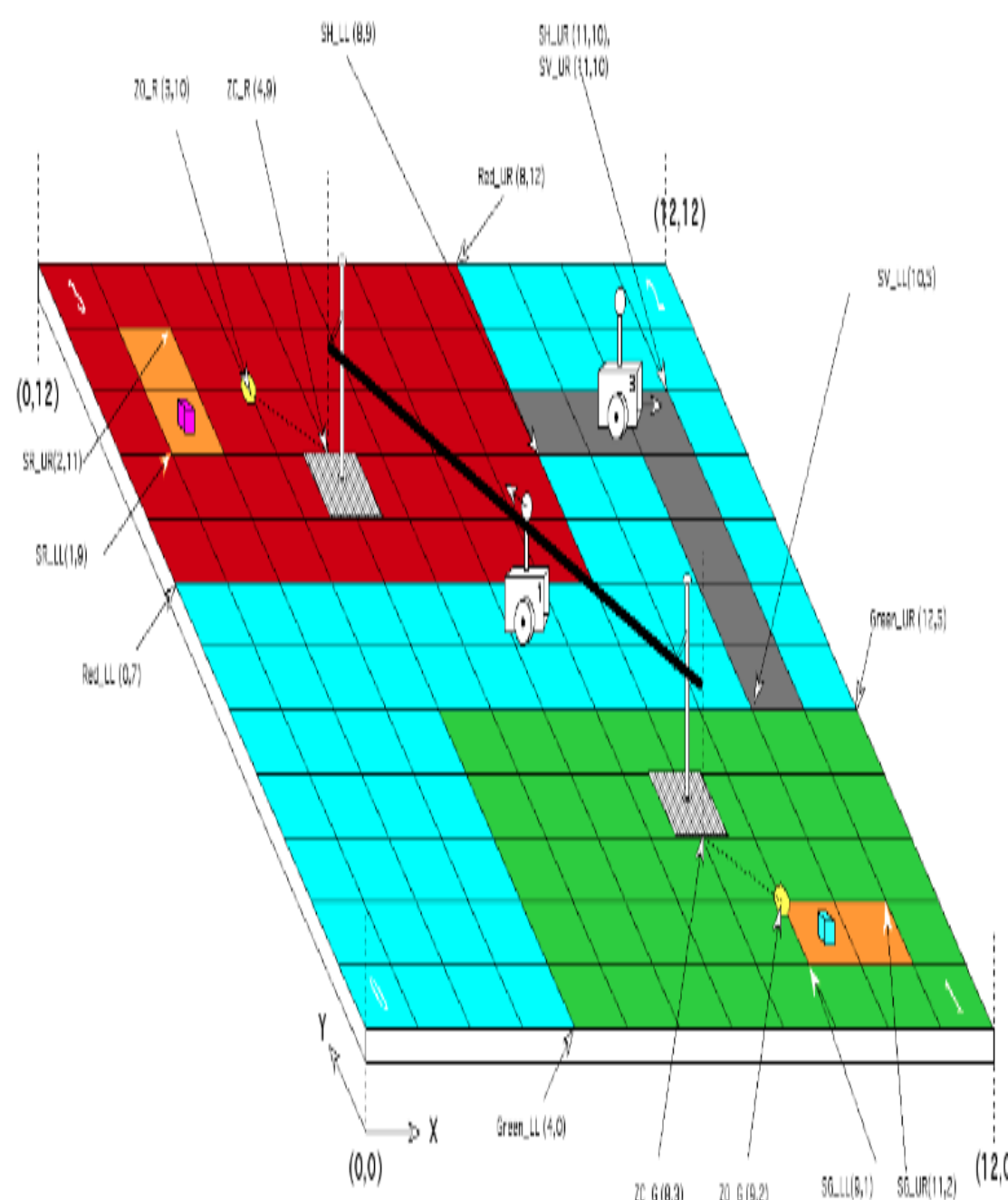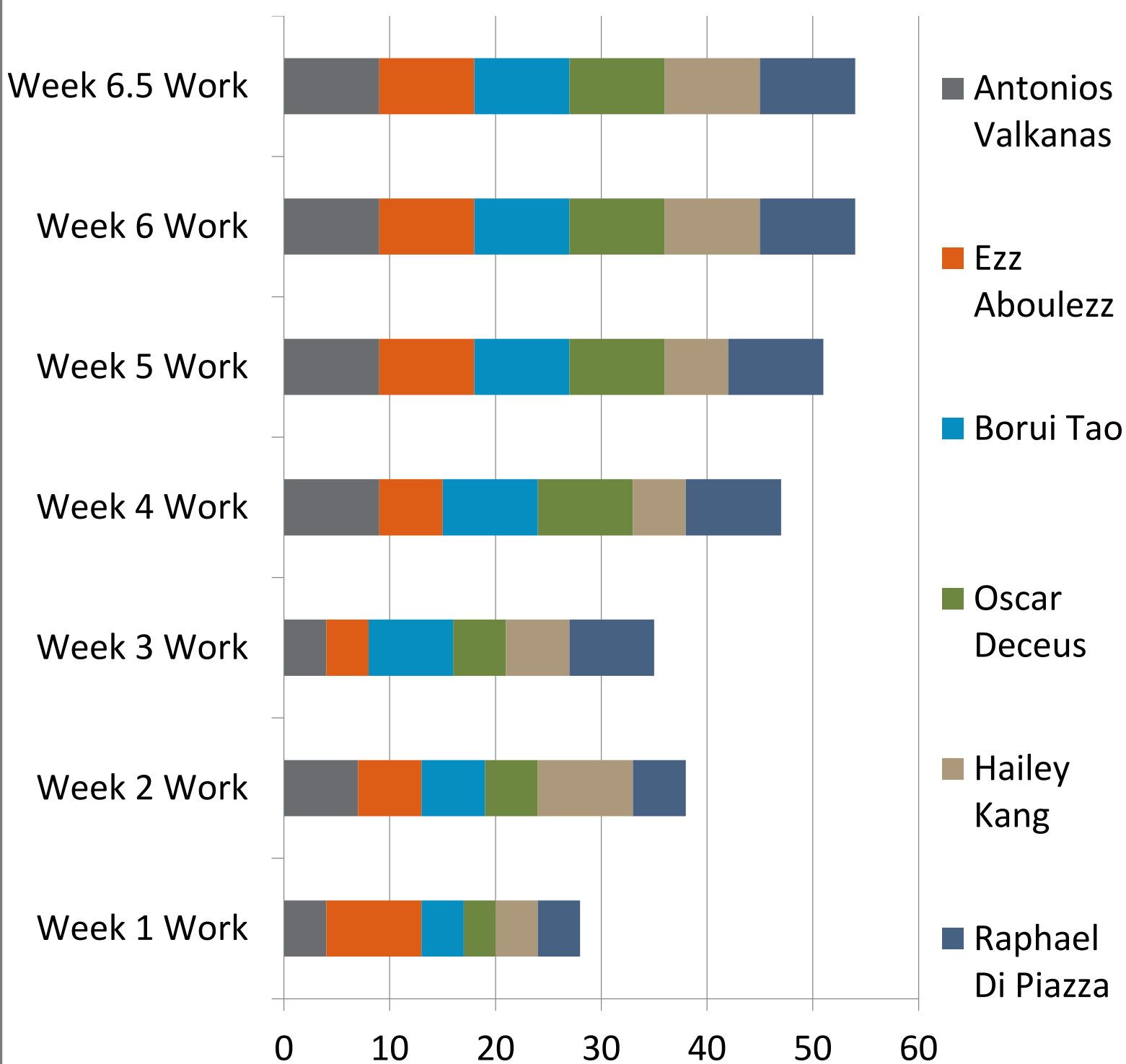# ICARUS

SPECIFICATION

The aim of the project was to design a robot for the purposes of playing a one versus one "Capture the Flag" game. The robot must be able receive data from the Wi-Fi, localize, navigate while avoiding obstacles, traverse a zipline and detect a block of a specific color that represents the flag. The time limit is 5 minutes.

TEAM MANAGEMENT & BUDGET

Hours worked per week, by team member:

Legend:
- Antonios Valkanas
- Ezz Aboulezz
- Borui Tao
- Oscar Deceus
- Hailey Kang
- Raphael Di Piazza
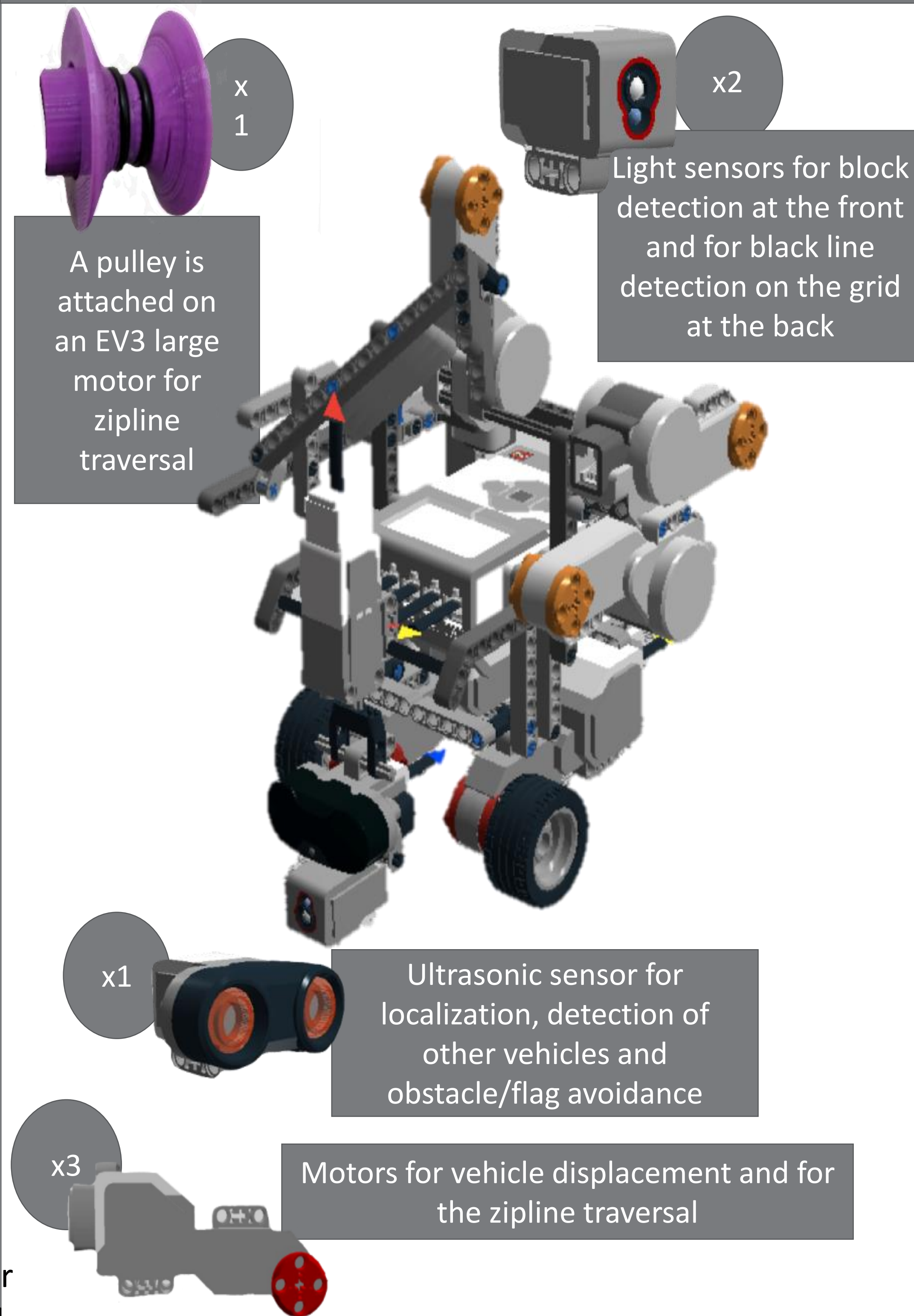
Budget: 350 hours
Total Cost: 310 hours
Slack Time: 40 hours

Team :
Antonios – Team Manager
Ezz – Documentation Lead
Borui – Software Lead
Oscar – Testing Lead
Hailey – Hardware Lead
Raphael – Software Eng.

HARDWARE

x1 — A pulley is attached on an EV3 large motor for zipline traversal

x2 — Light sensors for block detection at the front and for black line detection on the grid at the back

x1 — Ultrasonic sensor for localization, detection of other vehicles and obstacle/flag avoidance

x3 — Motors for vehicle displacement and for the zipline traversal
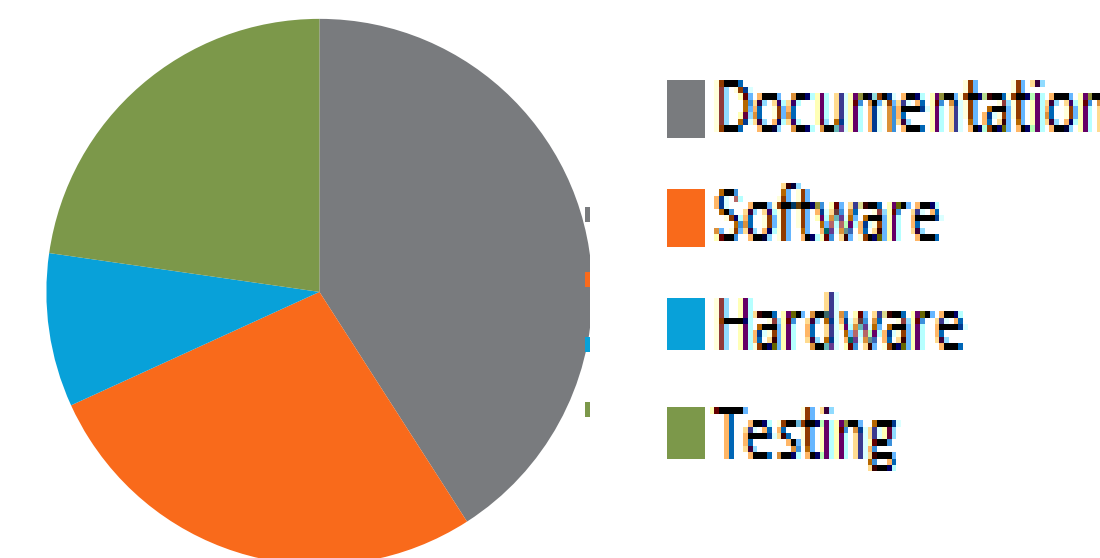
RESULTS

Very consistent when we start up to 8 tiles from the zipline, less consistent due to error accumulation further than that. Potential improvement: Odometry Correction.

DESIGN PROCESS

Gantt chart	W1	W2	W3	W4	W5	W6
Initial Ideas – R&D	■					
Mechanical Design		■	■			
Software Design		■	■			
Beta Testing			■	■	■	
Final Presentation						■
Final Documentation						■

Testing Strategy & Structure: Start with the easy parts, debug and add new parts on top of what already works.
- Characterization of the components
- Odometer tests
- Localization tests
- Navigation tests
- Zip-line Traversal tests
- Integration tests

- Documentation
- Software
- Hardware
- Testing

SOFTWARE

OdometerDisplay
run()

LightSensor
run()

CaptureFlag

UltrasonicPoller
run()

LightLocalizer
doLightLocalization()
adjustPosition()
midpointLocalization()
setCurrentTheta()

UltrasonicLocalizer
doUltrasonicLocalization()
fallingEdge()
risingEdge()
correctAngleAndLocation()
doBangBang(int bandCenter, int bandWidth, int motorLow, int motorHigh)

Navigation
travelTo(double x, double y)
turnTo(double theta)
boolean isNavigating()
motorStop()
getPosition(Odometer od)
doZipline(double x, double y, double zipDis, double travelDis)
setCurrentCoordinates
int convertDistance(double rad, double dis)
int convertAngle(double rad, double wid, double agl)

Odometer
run()
double getX()
double getY()
double getTheta()
setX(double x)
setY(double y)
setTheta(double theta)