- Project Description
    - Skipped
- Sensor Information
    - Moisture to
    - Water level is no longer used
    - Solar/Light (both are being used) to
    - Temperature to
    - Alarm from
    - Sprinkler from
    - Water Pump from
    - Sonar to
    - GPS to
    - Infrared Sensor to
    - Compass to
    - Accelerometer no longer used
    - Stop Buttons to
    - Bump sensors to
    - Indiana Jones switch to
    - Sensor servos from
    - Motors from
- Safety Concerns
    - Looked into watchdog pattern and updated UML to get separate safety system
        - Going to have watchdog pattern, class, or package that will receive heartbeat pulses from threads to either reset the thread, or overwrite what it was supposed to do and tell the robot to stop
        - Does that carry through down to MSP430s?
            - Yes. At the moment there's one class, but there will be more
            - There's one class that's just for internal software at the moment, but that will change so that it incorporates with MSP430's as well
- Updated UML Design
    - Red lines are for internal software use
    - Blue line is for output
    - Actual input class will be implementing its own watchdog. We can check to see if the data has changed. It's almost built into the register to check the heartbeat. The register/thread can be and is reset each time
    - Updated to separate out emergency outputs into a separate box and included red and blue lines for safety (watchdog).
    - Every sensor will update within 600ms. At top speed, the robot will move 8.8 inches within one second
    - Try to get movement sensors to update more regularly
    - Stop buttons are strictly hardware
    - Bump sensors should go to MSP430s (need software integration)

  - If bump sensor gets tripped, or Indiana Jones switch is activated, robot should stop
  - Black arrows aren't implying a data flow, they're location of dependency. Any time an arrow is located, it represents where things came from
  - Motor as lower level code is okay?
    - Yes.
    - Is that already implemented?
      - Yes. It's part of the controller code
    - How do you do turns?
      - 1 forward and 1 backward, or slow one down and keep the other moving, or stop one and keep the other moving.
      - Run by PWM
      - Needs to be documented as to how to get from code already written for this, to the software engineers
      - Wheels moving forward at same speed, but then getting adjusted based on encoder input. You might not be getting the same speed on each encoder. Is there a guarantee that the wheels will move forward at the same speed? And if it's not, give the SEs a value/difference that they can characterize. We either have to adjust the control algorithms, or fix it at the lower level so that they spin the same.
        - Maybe implement a PID controller.
        - At the moment, there has been identification that the encoder code needs changing.
    - Leaving team should carefully demonstrate that protocol document is true to current team
  - CalculateMove is starting to get complicated. Have you started to think about what that looks like architecturally?
    - The idea is that it's in the util class or method. So that if we wanted to do a specific angle, it's in the util class or method.
    - Or have a look-up table. Professor Hawker would recommend getting started on that soon
  - What is the relationship between the beagleboard and MSP430s. Haven't seen that system level diagram
  - Logic Classes and Sensors Input/ Output is in pre-read but was not discussed

- ➔ Updated Sequence Diagrams
  - Heartbeats aren't incorporated into these diagrams
  - Plant Care
    - There's a 10-20 minute loop because that specific one is checking soil moisture, plant life because they're not necessary to be checked all the time.
      - The exception should be the sunlight, it should be checked more often than every 10-20 minutes (maybe every 1-3 minutes), but the soil moisture is fine at about 20 minutes.

    - These are just early guesses, and there is room for modification
  - If conditions are less than optimal, plant will go into super-care mode
- Navigation
  - Starts with the switch to super-care
  - Will eventually get to the movement library and updating sensor values.
  - Where do the encoders for the motor fit in?
- Triggering the Angry Personality
  - Start off with emergency input switches and if they've been triggered or not.
  - Will then alarm activated or sprinklers activated
  - Who is responsible for the plant being down and getting out of angry mode?
    - Probably manual, maybe another trigger which would be the exact opposite of the angry personality trigger.
    - Needs to be determined if that's a manual reset, or on software, maybe have a timer on it?
      - Vote is to let it be software (if anything gets put back down, the alarm system turns off)