- Project Description
    - Idea of robot is that it will support a plant. Plant is on the robot and there will be sensors attached on the robot of everything the plant needs to stay alive.
    - To give robot a personality for interacting with people.
    - Make software for the robot to do all those things is the primary goal.
- Risks/Customer Needs/Engineering Specs
    - Risks
        - I2C communication we've had a few problems with.
        - We've had problems with internet access
        - Underestimating time needed for development (mitigated with tight schedule)
        - Software needs to be in tact on the robot
    - Customer Needs
        - Needs to stay within a certain area (GPS coordinates)
        - Storing sensor input
        - Watering the plant
        - Make sure it doesn't run into people
        - Personalities
        - QNX is on the board
            - No display, but don't need it
            - Want qconn so that remote debugging can be done through qconn
            - Holdup is getting network connectivity
                - Doesn't recognize USB dongle to register it on the board
            - Is there a fallback?
                - Haven't really thought about it
                - Major fallback is to do it through Angstrom instead of QNX
                    - IF everything happens to break and fall apart, we have Angstrom
                - Also can use GCC on QNX so there's at least the image and run it
            - Did you try the wireless adaptor?
                - Wired should come before wireless, trying to establish IP address
    - Engineering specs
        - List of engineering specs that correspond with customer needs (most are binary)
        - Bumps into things (not people)
        - Number of personality modes
        - QNX loaded
        - Number of personalities
        - On the bumping, are there other sensors
            - Sonars that are designed to see things far away
            - Bump sensors so that it will stop

- If it’s trying to spin the wheels, and they’re not spinning, the robot will stop because it means it ran into something
- Sonar range
  - 2’ diameter, 21’ range
- Proposed Design
  - Main will initialize the program
  - Two threads spawn
    - Basic care
      - Update every 5-10 minutes
      - Waters plant
      - Determines some of the personalities (using the plant interpreter)
        - Lack of sunlight =super care mode
        - Plant gets stolen=angry mode
    - Calculate move
      - Abstract (basically the brain of the program)
      - Utility class is vague
        - Meant to assist in what it’s supposed to do (support classes)
      - Once move has been calculated, movement gets drawn from movement library, and robot will move
  - “Move” sensors?
    - Mechanical parts are what’s being included
  - Where’s the interface to the motors?
    - Under moveinterpreter
  - Boxes are meant to be the actual hardware, and what’s above them is what pulls from the registers
  - Calculate move will decide what it wants to do, then calls actual movement command, then the box represents the actual hardware and the motors are told what to do
  - What’s the different between things that are primarily input of data, and things that are output of data?
    - Alarm and sprinkler are output
    - Move, motor control input and output
    - Might want to better distinguish
  - Move class is the same for all personalities, but the amount depends on the personality
    - Might want to put that move abstract (that’s distinguished by personality) into the “personality” box
    - “Move” is the actual movement library; it’s the same for all personalities
      - Is it on/off, or are there speed personalities
        - Right now, the mode is continuously variable (on the controller)
        - Will make it stopped and moving
        - Might include speed
    - “Personality” just tells “move” what it wants to do

  - Include more descriptions as to what each of the classes is
    - Where are the following personality/movement aspects?
      - Navigation
        - In movement
      - Sensor fusion
        - IR
        - Sonar
        - GPS
        - Encoders
      - “World Model”
      - Where/how to get water
      - How will it get water
      - Where is power management?
        - Currently battery is large enough;
        - Long range want it to use sonars
      - Geo-spacial
        - Sun
        - Danger (i.e., steep slopes)
        - Might not have enough time
        - Reach/understand world
      - Robot safety and health; people safety and health
        - There will always be somebody near it
        - Has bump sensors, emergency shut off switches
        - Mechanical casing
      - Safety and health watchdog
        - Sensors aren’t going off for example
        - Add package for safety and health
          - Backup of the software (basically emergency switch)
  - Include descriptions in UML chart

- ➔ Use Cases
  - Water the Plant
    - Will know when to water plant
  - Plant gets stolen
    - Alarm will go off until somebody deactivates it
    - Sprinkler will only go off once
  - Switch personalities autonomously
    - Depends on position and situations
  - Switch personalities manually
    - Will be able to be switched from the touch screen
  - Apathetic wander
    - System will randomly move about within boundaries and avoid obstacles

- Sequence Diagrams
    - Plant care
        - Explained the sequence of how plant care will work
            - Basic care, plant interpreter, plant sensors, personality controller, and navigation
    - Navigation
        - Explained how navigation will work (more in depth from previous sequence diagram)
            - Personality controller, super care move, navigation interpreter, navigation sensors, move, move interpreter, motors
        - Towards the left more event driven, towards the right, cycle. Is there a response time?
            - Expecting sensors will update as often as possible
            - Do you know the rates or estimate?
                - Depends on the MSP430 and how it's coded
                    - Plant is every 20 seconds
                    - Navigation is every 800ms
                    - Motor controller hasn't been looked at lately because it's moving
            - It's top speed is 0.5mph
            - Connection between beagleboard and MSP430 for the wheels it's serial based, so it'll be faster. It's PWM with encoders.
                - If the pulse width doesn't change, it will continue to go.
            - If the right side of the sequence diagram goes, will it stop? Or will it continue going (safety)?
                - Needs to be getting something from the upper level every second, and if it doesn't, it should stop
                - Implement into MSP430s
                - Not software controlled at the moment, so we don't know
        - Define how data flows (protocol) so that we know safety
            - Need flow of timing
        - Multiple processers? Is it fast enough?
        - Make performance protocols
            - Meet after meeting to decide
- SE Tentative Schedule
    - Going to redesign and finalize schedule and start the development phase this week
        - Starting with basic structure, then moving into actual movements and movement patterns.
        - Goal is to mostly get general framework done before the end of this quarter, that way all of next quarter can be worked on personalities
        - Motors turning on is the "Interpreter/Move library" segment of schedule.

        - If stuff gets completed soon, schedule will move up.
        - This schedule is truncated and exam weeks/break week is not completed.
        - To test, beagleboard is needed, otherwise it is not.
    - Slack would like to get board on robot, and set up with the joystick so that ME's can get working on that.
        - Doesn't impact SE at all
    - Testing starts with initial development. As it's being developed, it's being tested.
        - On robot it will probably start next quarter. It will be local testing for now.
    - Simulation environment will probably be built so that robot is not always needed
        - How much can you simulate?
            - A chunk. In SE viewpoint, all they're working with is data, so if they have data, they can simulate it.
        - All done on the PC
- ➔ Personality Framework
    - There were no new personalities, we did some within-team generation, but mostly took ideas from design review last week and tried to implement it.
    - Going to add colors somehow to each mood
    - First mode is diagnostic mode, then apathetic, and finally angry and all of the other ones.
    - 4 places/coordinates were picked before by team P10218
        - Those might need to be redefined now so that it can be confined to a location to observe it better
    - GPS has a 2m accuracy
        - Maybe let robot remember where it's been and where it should travel and remember where there was sunlight.
            - Was decided that there's not really a point to have this, except for remembering where home is and maybe some waypoints on the way there.
    - GPS coordinates
        - Will they be selectable?
            - On LCD?
            - We want to avoid people randomly coming up and inputting GPS coordinates
                - Password protect
    - Original thinking would be for simplicity keep it to one set of GPS coordinates. However, it would be very nice to see the robot move with software; that's the primary task, more complicated things can come later.
- ➔ Sonar Placement
    - Discussed current design and field of view
        - Sonars have a 2' diameter cone. Takes 2' to get to it's full width, and then kind of tapers.

        - Current placement doesn't take advantage of that full beam
    - Discussed idea
        - Would allow it to see above and below, and to the sides
        - Sonar 1 scans in just the front on the bottom
        - Sonars 2 and 3 are on corners also scanning up towards the top
        - Sonar 4 sweeping in the back (was on a servo always)
        - Currently there's going to be a slight gap between the two, but if we can get a 3' diameter sonar it would be better.
            - The current sonars are waterproofed, which is why a 2' diameter one is used
        - Would need two more servos to implement this, looking for approval
            - They work currently (the servos)
        - Software
            - As long as it's giving information, it would be simple
            - Wants protocol document (who's commanding what, who's receiving)←Ken
                - What it will probably be is using I2C, and giving it number of degrees to move to, and have the MSP430 move it. Ken, define what that is, what data are you sending from a software point. (what's the MSP430 receiving)
        - Mechanically Anna and Nick are doing this if it's going to be implemented.
- EE/CE status (not part of design review meeting, but still talked about)
    - Explained what was working and what wasn't.
    - Servos and Sonars work with some coding still needed
        - Relationship for servos was found by P11215
    - IR sensors aren't done yet
    - Do we have enough IR sensors?
        - There are 3
            - Spare in the bag
            - 1 where the pin broke off(is it reparable)
            - Order some more
            - Dr. Sahin mentioned getting a moving average loop for P11215 so that it can avoid falling off of cliffs
    - Accelerometer is low priority (don't know if it's needed b/c of other stuff)
    - Compass is working
    - Bump sensors are new, need coding
        - Electrical circuit is all set.
        - Needs a place to connect to MSP430s
            - Figure out which pin will be used, and let John know.
    - Battery sensor needs to be double checked
    - Water sensor is the water level in the tank;

    - Was removed.
      - Last time it sat in water tank and could register water level but didn't work too well;
      - Decided it wasn't needed because you can see where the level is
  - Moisture sensor
  - Temperature sensor
    - Has code, ADC values are all wrong
    - Tries to convert to actual temperature, but actual values are wrong;
    - ADC is working
    - Where is it mounted?
      - Not sure yet, find a spot and work up harness
  - Light sensor
    - Doesn't work too well
    - Same state as temperature
    - The conversion values of the ADC values are wrong
    - Needs a place to be mounted
  - Alarm is Indiana Jones switch
- Other (pre-review)
  - Concerned about electronics and that Rui isn't as involved as she can be
    - Need a test plan for EE/Software side
  - Going back and using Angstrom?
    - Problem with I2C is still a hardware issue
    - Work on QNX, but if not, go back to Angstrom
    - There's a UR to USB converter on the board
      - If I2C doesn't work, there's another backdoor (maybe a frontdoor)
      - UR to USB to hub to Beagleboard
        - How SE's get information, it doesn't matter, as long as information is there.
      - If things don't happen today, need to start looking at alternatives
    - Been in contact with QNX representative. We have functionality, but nothing going to display
      - Trying to set up internet(wireless and wired);according to representative, it should just plug in and just work
  - Motors on Wandering Ambassador