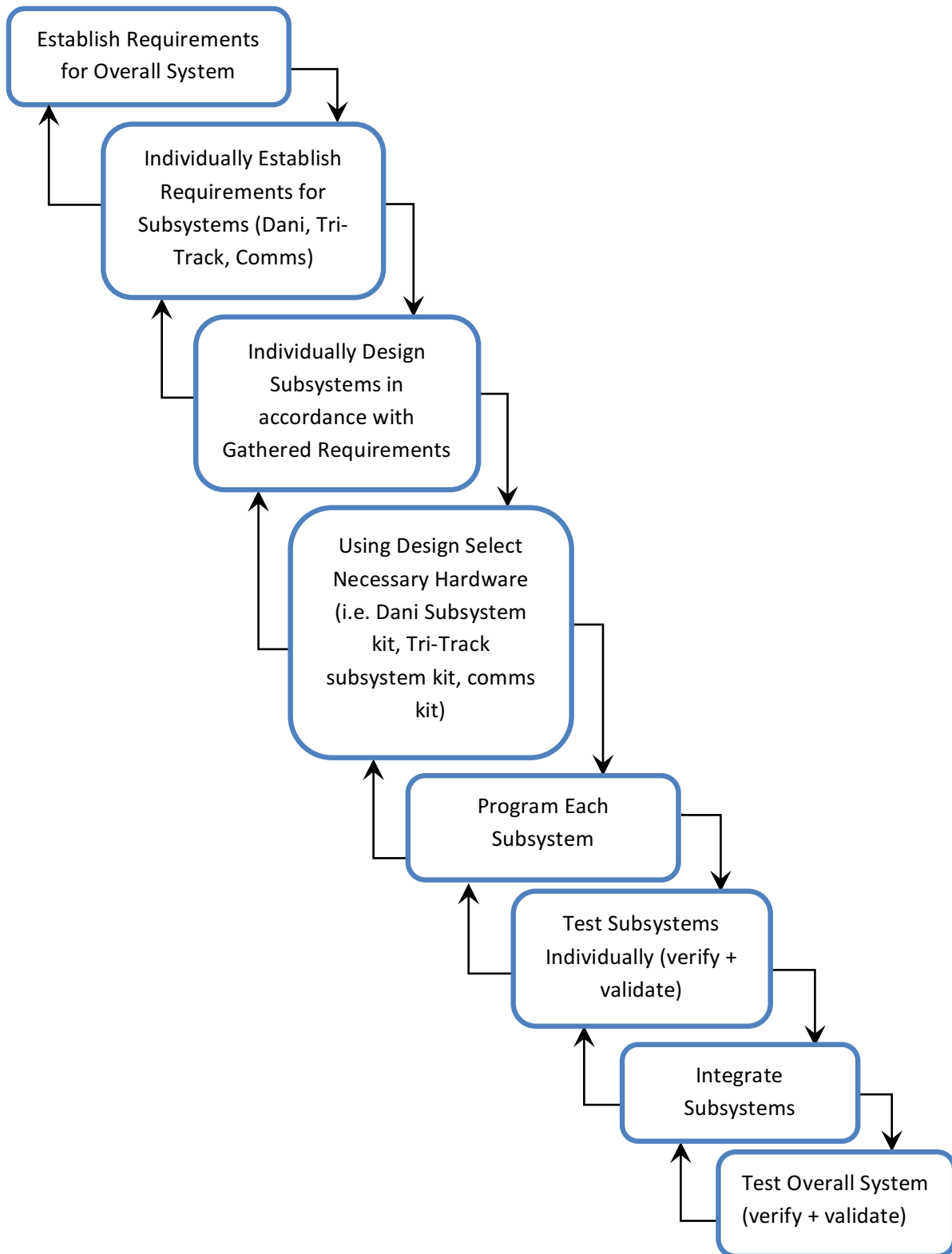


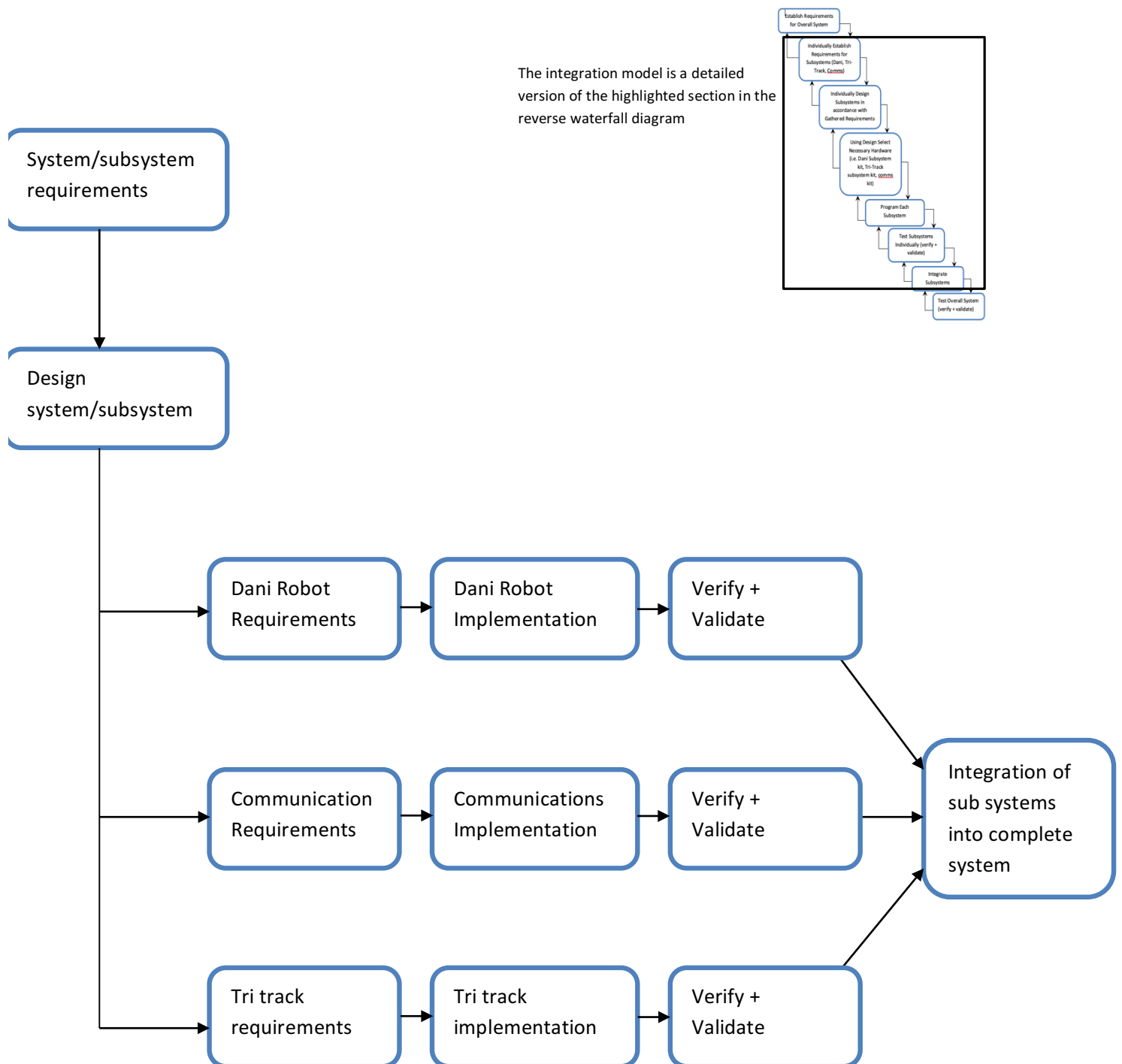
Systems Architecture

As a group we came to the conclusion of needing three system architecture models.

One for the overall system, which we chose to have a reverse waterfall; this allows the team to keep track of which stage of the project they are at.



The second architecture is specific to subsystem development and integration; this model shows the approach for the subsystems. This model is not to model the entire system, only to model the development of subsystems and integration. Each subsystem is conducted by a sub-team in Team ICARUS



Finally one specific just for the programming aspect; as there will be two different programs being created (i.e. one for Dani robot system and one for the tri track system) they have a spiral model each. As the tri track is going to use the raspberry pi, this is programmed in python; this language will work very well with the spiral model as you can program a feature at a time.

The code will be developed step by step (feature by feature); therefore one aspect will be planned, developed and tested before moving onto the next the spiral model.

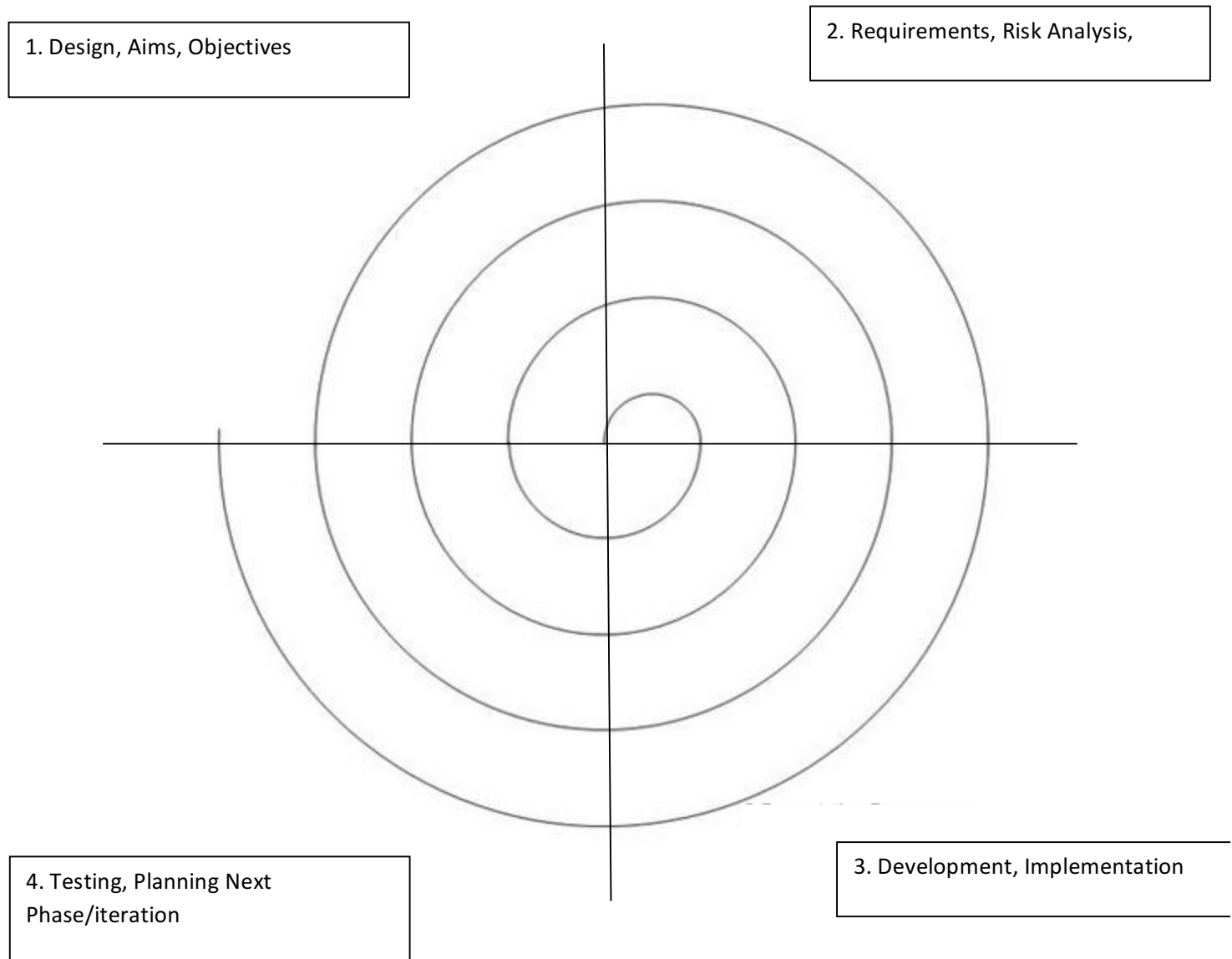


Figure 1: <http://www.presentation-process.com/spiral-model.html>

Spiral only; axis and labels added manually

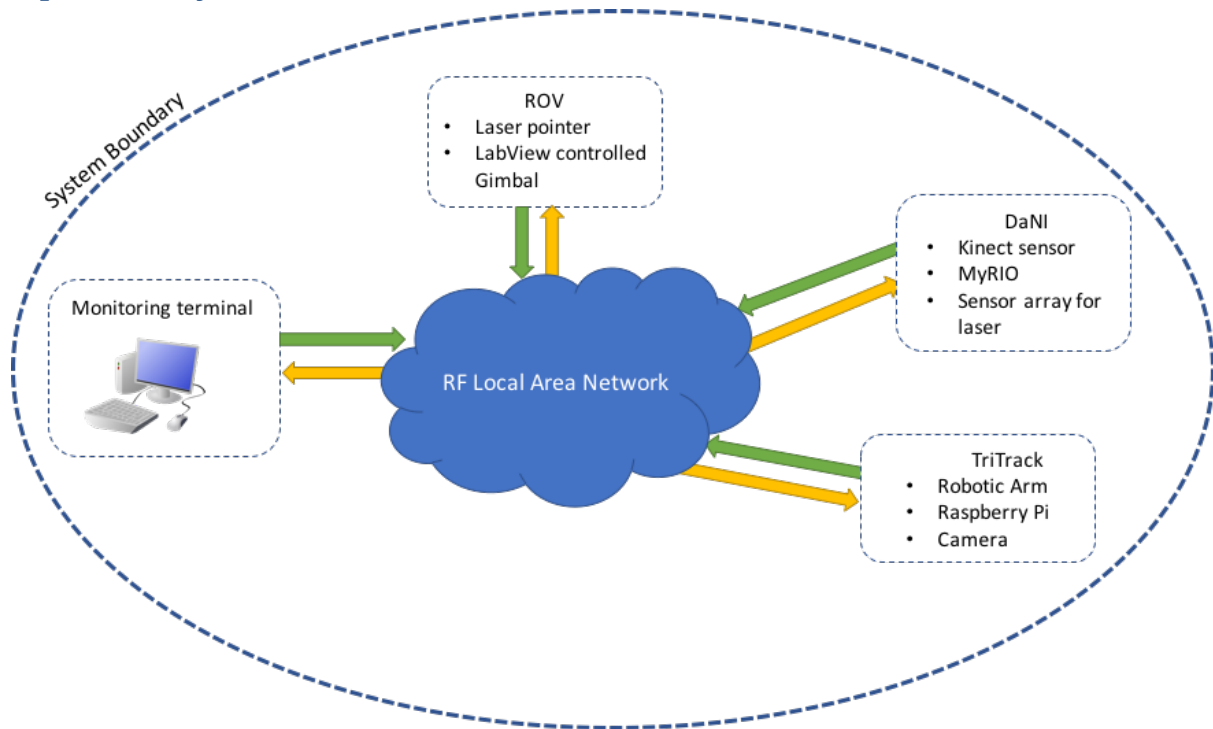
Iterations

#	Dani Subsystem
1	Obstacle detection
2	Obstacle avoidance
3	Route finding algorithm for navigation

#	Tri-Track Subsystem
1	Moving the subsystem forward, backwards and turn (left, right)
2	Basic claw manoeuvring with controller
3	Identifies the black box in a photo
4	Calculates distance in centimetres that the robot needs to move + direction

#	Additional Iterations	Subsystem
1	Voice control assistant; 'Jarvis'	Tri-track Subsystem

Updated System Architecture:



Monitoring Terminal:

Standard Windows PC running Labview that monitors the status of the two robots via interfacing with the local RF network. Performance and bandwidth constraints must be considered when retrieving data from the two robots.

ROV:

Gimbal mounted laser guiding the DaNI robot via a closed feedback loop to ensure the DaNI's location is accurately known as this has a direct impact on the effectiveness of the TriTrack robot's abilities to retrieve the black box and close the fuel valve.

DaNI:

The DaNI robot uses a MyRIO to host Labview and uses a Kinect sensor to identify hazards and map the route for the Tritrack. This is coupled with the ROV's capabilities to guide the DaNI. The two devices interface through the local RF network.

TriTrack:

The Tritrack is used to retrieve the black box and manipulate the fuel valve. It receives instructions on where to travel from the DaNI robot which maps the area and plots a path for the TriTrack to follow. The TriTrack uses a raspberry pi to control the robotic arm and communicated to the DaNI and Monitoring Terminal via the local RF network.